Enhancements of T-REFWA to Mitigate Link Error-related Degradations in Hybrid Wired/Wireless Networks

Hiroki Nishiyama, Tarik Taleb, Abbas Jamalipour, Yoshiaki Nemoto, and Nei Kato

***Abstract*— With the on-going wireless access technologies, the Internet has become accessible anytime anywhere. In wireless networks, link errors significantly degrade the performance of the Transmission Control Protocol (TCP). To cope with this issue, this paper improves the recently-proposed Terrestrial REFWA (T-REFWA) scheme by adding a new error recovery mechanism to its original design. In the T-REFWA scheme, senders are acknowledged with optimal sending rates at which an efficient and fair utilization of network resources can be achieved. As the feedback values are computed independently of link errors, senders can keep transmitting data at high rates even in case of link error occurrences. Using this feature, the proposed error recovery mechanism can achieve high throughput in environments with high bit error rates. The throughput is further improved by modifying the exponential backoff algorithm of TCP so that long idle times are avoided in case of link errors.**

We show through simulations that the proposed method improves TCP performance in high bit error rates. Compared with several TCP variants, the proposed error recovery scheme exhibits higher link utilization and guarantees system fairness for different bit error rates.

***Index Terms*— Link error, wireless network, transmission control protocol (TCP), congestion control, T-REFWA**

I. INTRODUCTION

ALONG with the rapid globalization of the mobile telecommunications industry, Internet, at present, has become available anywhere anytime. Wireless LAN (WLAN) systems relying on Wireless Fidelity (WiFi), such as 802.11a/b/g, enable users to access the Internet via broadband links. Mobility is enabled also by different technologies, such as Universal Mobile Telecommunications System (UMTS). New wireless access technologies (e.g. Worldwide Interoperability for Microwave Access (WiMAX) and 4th Generation cellular systems (4G)) are expected to provide more broadband wireless links.

In such environments, a large amount of data containing high-quality images, movies, or music is more likely to be exchanged between servers and many mobile users over the Internet. However, in case that a wireless link is included in the path, the performance of the Transmission Control Protocol (TCP), the most widely used protocol for data transmission, is significantly degraded due to link errors in the wireless part.

H. Nishiyama, T. Taleb, Y. Nemoto, and N. Kato are with the Graduate School of Information Sciences, Tohoku University, Sendai, Japan. (e-mail: bigtree@it.ecei.tohoku.ac.jp, talebtarik@ieee.org, nemoto@nemoto.ecei.tohoku.ac.jp, kato@it.ecei.tohoku.ac.jp)

A. Jamalipour is with the School of Electrical and Information Engineering, University of Sydney, Australia (e-mail: a.jamalipour@ieee.org)

A TCP sender operates on the conservative assumption that any segment losses are due to network congestion. Accordingly, it unnecessarily cuts down its sending rate upon a loss event. This phenomenon leads to a waste of bandwidth and ultimately lower link utilization. On the other hand, in case of communications between a server and a large number of users, TCP results in drastically unfair bandwidth allocation among users. This issue becomes more significant when users have high variance in their Round-Trip Times (RTTs) distribution. As an attempt to solve the unfairness issue of TCP, the authors have proposed the T-REFWA [1] scheme. In this paper, we further enhance the working of T-REFWA by extending its design to environments with link errors. With the proposed enhancement, the proposed method achieves efficient link utilization and fair bandwidth allocation in wireless networks with link errors.

The remainder of this paper is structured as follows. Section II highlights some research works in the context of improving the performance of TCP in IP networks. The proposed method is described in Section III. Section IV portrays the simulation environment and discusses the simulation results. Finally, the paper is concluded in Section V.

II. RELATED WORK

TCP is usually not capable of discerning the cause of losses but only its occurrence. A TCP sender considers reception of duplicate acknowledgments (ACKs) and timeout as indication of network congestion. Once the sender receives three duplicate ACKs, the congestion window is reduced by half to avoid additional packet losses. When the sender experiences a timeout indicating multiple drops due to heavy congestion, the congestion window is set to one and the TCP enters the slow start phase. Moreover, if timeouts occur continuously, the sender doubles the Retransmission TimeOut (RTO) value according to the backoff algorithm. In case of packet losses due to link error, the sender unnecessarily cuts down its transmission rate. This results in a waste of the network resources. To solve this issue, a new rate control mechanism, aware of the loss type, is required. Some improvements to the standard TCP have been devised in recent literature (e.g. TCP Santa Cruz [2], Wide area wireless networks TCP (WTCP) [3], TCP Vegas [4], and TCP Westwood [5]). In most of these methods, the data sending rate is adjusted based on measurements of RTT or the observed transmission rate at end terminals.

In TCP Santa Cruz, the sender can determine whether congestion is increasing or decreasing in both the forward and reverse paths of the connection based on the inter-arrival time of data packets at both the sender and receiver sides. This monitoring permits the detection of the incipient stages of congestion. This operation enables a prompt adjustment of the congestion window. Moreover, TCP Santa Cruz improves the RTT estimation mechanism by introducing ACK Window, which is similar to the bit vectors used in TCP SACK [6], to notify multiple losses via ACK packets. By these improvements, TCP Santa Cruz can promptly retransmit and recover lost packets, without waiting for the fast retransmit phase, even under heavy loads.

WTCP uses the ratio of the average packet inter-arrival time at the receiver to that at the sender as a primary metric for rate control. The desired sending rate is computed at the receiver side and notified to the sender via ACK packets. The sender monitors ACK packets and accordingly adjusts its rate. As a result, WTCP reduces the effect of non-congestion related packet losses on the transmission rate control. Meanwhile, WTCP does not use RTO so as not to be affected by erroneous RTO estimation. In WTCP, the receiver has to periodically send ACKs at a frequency tuned by the sender in order to signal the new transmission rate. This means that the sender receives at least one ACK during a given period of time. This procedure is hence used for detecting a deadlock instead of RTO.

TCP Vegas compares two values; the expected throughput computed based on the observed RTT and the actual throughput observed over a RTT period. It then adjusts the congestion window accordingly. When the actual throughput deviates largely from the expected throughput, the sender decreases the congestion window linearly during the next RTT to avoid possible congestion in the network. On the other hand, when the actual throughput is relatively close to the expected throughput, the sender increases the congestion window linearly during the next RTT as this indicates more available bandwidth at the network. By so doing, the congestion window size can be stabilized. With a new retransmission mechanism using the timestamp, TCP Vegas is able to retransmit a lost packet without waiting for three duplicate ACKs. Therefore, the TCP Vegas does not experience coarse timeout even if losses are significant or the bandwidth is not large enough to transmit three duplicate ACKs.

TCP Westwood is the most notable example among protocols which improve TCP performance, particularly in wireless networks with high bit error rate. The key concept behind TCP Westwood is to estimate the available bandwidth by measuring and averaging the rate of returning ACKs at the sender's side. After a timeout occurrence or reception of three duplicate ACKs, the sender estimates bandwidth availability and accordingly adjusts the congestion window size and the slow start threshold. In this way, TCP Westwood ensures faster recovery from packet losses and guarantees an efficient utilization of network resources. Although TCP Westwood has been shown efficient in wireless networks, its performance largely depends on the accuracy level of the network bandwidth estimation.

While a number of improvements to TCP have been implemented at the end terminals, other approaches have considered the addition of new mechanisms only to the network elements, such as routers and gateways. eXplicit Control Protocol (XCP) [7], Explicit Window Adaptation (EWA) [8], and WINdow TRAcking and Computation (WINTRAC) [9] are few notable examples. In these methods, network elements along the path of a TCP source to a TCP destination signal the optimal transmission rates (window sizes) to the source. These schemes are efficient in making full use of network resources. They, however, can not resolve the unfairness issue among connections with high variance in their RTT distribution.

To cope with both the network utilization and fairness issues of TCP, the authors have recently proposed the Terrestrial-REFWA (T-REFWA) scheme. The basic operation of T-REFWA relies on the Recursive, Explicit, and Fair Window Adjustment (REFWA) [10] [11] [12] scheme proposed for satellite networks. The REFWA scheme achieves high efficiency by matching the sum of window sizes of all active TCP connections sharing a bottleneck link to the effective bandwidth-delay product of the network. Moreover, the system fairness is improved by assigning for each flow a feedback proportional to its RTT. In satellite networks, REFWA can estimate the RTT of each flow by monitoring hop counts in the backward and forward traffic of each flow. In terrestrial networks, however, prior knowledge of RTT is not available at network elements. In the T-REFWA scheme, a source notifies the information of RTT to network elements via data packets. Based on this information, senders are acknowledged optimal sending rates so as not to overload/underutilize the network. The current format of T-REFWA is not efficient in high bit error rate environments. In this paper, we extend T-REFWA to environments with link errors. We add a new error recovery mechanism to T-REFWA to enable it achieve high efficiency and fairness in hybrid wired and wireless networks. The proposed error recovery mechanism is dubbed T-REFWA Plus.

III. Operations of the T-REFWA Plus scheme

A. Overview of the T-REFWA scheme

This section describes in detail the major operations of the T-REFWA Plus scheme. We firstly give a brief overview on the original T-REFWA scheme. In current TCP implementations, RTT is computed to make an estimate of RTO. The average value of RTT is denoted as Smoothed RTT (SRTT). In T-REFWA, a sender writes down the value of SRTT in the Type Of Service (TOS) field of IP headers and sends it to specific network elements via the communication path. Given the limited size of the packet header field, RTT values are transformed to an integer value within the range $[0, 63]$. For a detailed description of this transformation procedure, the interested reader is referred to [1]. At routers, flows are grouped according to their RTT indicator α. Each group is defined as the set of flows having the same RTT value. Flows are identified by a flow ID and are defined as streams of packets sharing the quintuple: source and destination addresses, source and destination port numbers, and protocol field. A flow is considered to be in progress if the elapsed time since its last packet transmission time is less than the most recent estimate of the average RTT_α of all active flows traversing the router, RTT_{avg}.

TABLE I

CONGESTION AVOIDANCE ALGORITHMS IN STANDARD TCP, TCP WESTWOOD, AND T-REFWA PLUS

	Standard TCP	TCP Westwood	T-REFWA Plus
Duplicate ACKs	$ssthresh = old\_cwnd/2$ $cwnd = old\_cwnd/2$	$ssthresh = f(BWE)$ $cwnd = \min(cwnd, f(BWE))$	$ssthresh = \max(ssthresh, W)$ $cwnd = \max(cwnd, W)$
Timeouts	$ssthresh = old\_cwnd/2$ $cwnd = 1$ $RTO = \min(2 \times RTO, 64)$	$ssthresh = f(BWE)$ $cwnd = 1$ $RTO = \min(2 \times RTO, 64)$	$ssthresh = \max(ssthresh, W)$ $cwnd = \max(cwnd, W)$ $RTO = RTO$

Similarly to the original REFWA scheme, the feedback computation is performed periodically every RTT_{avg} time interval. The feedback computation load is thus not so heavy. At time $(t = n \cdot RTT_{avg})$, the feedback value of flows belonging to Group α, $W_\alpha(n)$, is computed as in Equation (1) (shown at the bottom of the page). In Equation (1), B and Bw are the router's buffer size and the link bandwidth, respectively. n_j denotes the size of Group j and RTT_j denotes the RTT value of its flows. $\Upsilon(n)$ and $Q(n)$ denote the aggregate TCP window size and the router's queue occupancy at time $(t = n \cdot RTT_{avg})$. ϕ and ψ are constant parameters. It should be recognized that ϕ and ψ play a significant role in exploiting well the unused bandwidth and free buffer size, respectively. Details on the setting of ϕ and ψ, and on the working of the T-REFWA scheme can be found at [1] [13].

B. Major procedures of T-REFWA Plus

When TCP data packets are dropped at any link along the communication path, TCP senders receive duplicate ACKs or set up a timeout. Senders can not infer the reason behind packet drops. They simply consider them as an indication of network congestion. Therefore, over wireless networks with link errors, standard TCP unnecessarily cuts down its congestion window and decreases its transmission rate even if no congestion has actually occurred.

In T-REFWA scheme, the sender gets appropriate feedback on congestion window written in the RWND field of ACK packets and computed at bottleneck links. A T-REFWA sender avoids network congestion by setting its congestion window size to the feedback value. Figure 1 shows the difference between T-REFWA and T-REFWA Plus in congestion avoidance. The congestion window control mechanism in T-REFWA scheme is the same as that of standard TCP and only limits the maximum window size to the computed feedback written in the RWND field. The T-REFWA needs thus long time till the congestion window size reaches the feedback value, while T-REFWA Plus sets its congestion window size to the feedback value right after a packet loss event. On the other hand, the feedback value depends on only the RTT distribution of flows sharing the same bottleneck link and the buffer occupancy at the bottleneck router. In other words, the feedback value is independent of the link error. The remainder of this section explains how this feature is used to combat error-due packet drops in T-REFWA Plus scheme.

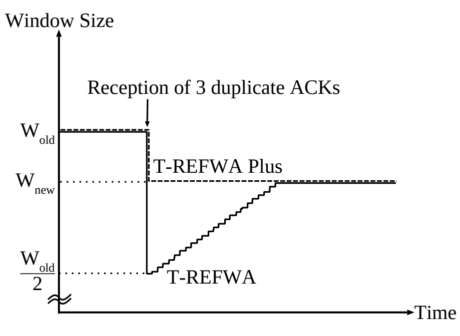


Fig. 1. Window size variations in T-REFWA and T-REFWA Plus

Upon reception of three duplicate ACKs, a TCP sender performs fast retransmit and fast recovery where its congestion window size ($cwnd$) is halved and the slow start threshold ($ssthresh$) is set equal to the congestion window size. In T-REFWA Plus, the congestion window size and the slow start threshold are updated based on the feedback value W received prior to the fast recovery phase. The congestion window is set to the feedback value if the feedback value is larger than the current congestion window size. The slow start threshold is set in the same manner. The sender then enters the congestion avoidance phase similarly to standard TCP. This control mechanism is similar to that of TCP Westwood. TCP Westwood, however, uses BandWidth Estimate (BWE) instead of the feedback value W for window controlling. The BWE is defined as the share of bottleneck bandwidth used by the connection and estimated based on the ACKs arrival rate at the sender. The sender resets the congestion window and the slow start threshold using the function of BWE (e.g. $BWE \times RTT$). TCP Westwood sets the congestion to a small value as standard TCP after a timeout expiration while the proposed method can update the congestion window and the slow start threshold as in the case of reception of duplicate ACKs. It should be noted here that the efficiency of TCP Westwood can be largely

$$\begin{aligned} W_\alpha(n) &= \frac{RTT_\alpha}{\sum_{j=0}^{63} n_j \cdot RTT_j} \cdot \Upsilon(n) \\ \Upsilon(n) &= \Upsilon(n-1) + \phi\Big(Bw \cdot RTT_{avg} - \Upsilon(n-1)\Big) + \psi\Big(B - Q(n-1)\Big) \end{aligned} \tag{1}$$

affected by errors in the estimation of BWE, whereas the working of T-REFWA Plus is based on an explicit notification of the optimal sending rate, a value most likely to be accurate. Moreover, in order to improve TCP performance in heavy loss environments, T-REFWA Plus freezes the RTO. The retransmit timeout backoff algorithm used in most TCP variants doubles the RTO when coarse timeouts occur in succession. In case of high bit error rate environments, this mechanism leads to a significant waste of both bandwidth and time. To overcome long idle waiting times due to large RTO, the proposed method does not double the timeout value after a timeout expiration. This concept is similar to the idea presented in [14]. Table I summarizes the congestion control mechanism of standard TCP, TCP Westwood, and T-REFWA Plus and compares them. Finally, it should be recalled that the proposed T-REFWA Plus scheme can be implemented without significant changes and requires a simple modification at only the TCP sender.

IV. Performance Evaluation

A. Simulation Scenario

To evaluate the performance of the proposed T-REFWA Plus scheme, we perform computer simulations using Network Simulator version 2 (NS2) [15]. We consider a simple network topology with a single bottleneck link as shown in Fig. 2. In such a network, a server provides a number of users with a particular application (File Transfer Protocol (FTP) in this simulation). The last one hop to mobile users is wireless link with a pre-defined link error rate. The bottleneck link capacity bw is set within the range of 10Mbps to 200Mbps. Wireless link delays are almost zero and each group has different RTT. $60\text{ms}, 120\text{ms},$ and 180ms are set for Group 1, 2, and 3 respectively. Queuing delays are ignored. All groups consist of equal number of mobile users and the size of each group N is varied from 1 to 100. To avoid bursty drops at the simulation launch time, all mobile users are randomly

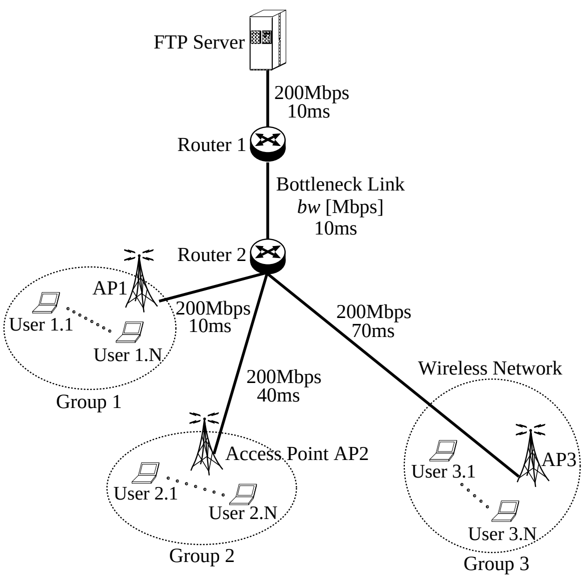


Fig. 2. Simulation topology

TABLE II
SIMULATION PARAMETERS

Parameters	Range of Values
Bottleneck link capacity bw	$10\text{Mbps} - 200\text{Mbps}$
RTTs of each group	$60\text{ms}, 120\text{ms}, 180\text{ms}$
Packet size	1024bytes
RWND	128packets
Size of each group N	$1 - 100$
Packet error rate (PER)	$10^{-5} - 0.5$
Simulation run time	100s

activated over a time interval of 1s. The TCP data packet size is fixed to 1024 bytes and RWND is set to 128 packets so that the maximum traffic rate generated by one user is about 18Mbps in Group 1 (with the smallest RTT=60ms). These parameters are chosen with no specific purposes in mind and do not change any of our fundamental observations about the simulation results. In order to remove limitations due to small buffer size on network congestion, we use buffers equal to the bandwidth-delay product of the bottleneck link. All buffers employ Drop-Tail as packet-discarding policy. To remove the influence of TCP synchronization which results from having multiple connections increasing their windows at the same time, we use TCP Reno and TCP Newreno with Random Early Detection (RED) [16] as packet discarding policy. Simulations are all run for 100s, a duration long enough to ensure that the system has reached a consistent behavior. The packet loss probability for link errors in the wireless part is varied within the range $[10^{-5}, 0.5]$. Table II shows a complete list of the simulation parameters and the range of its values.

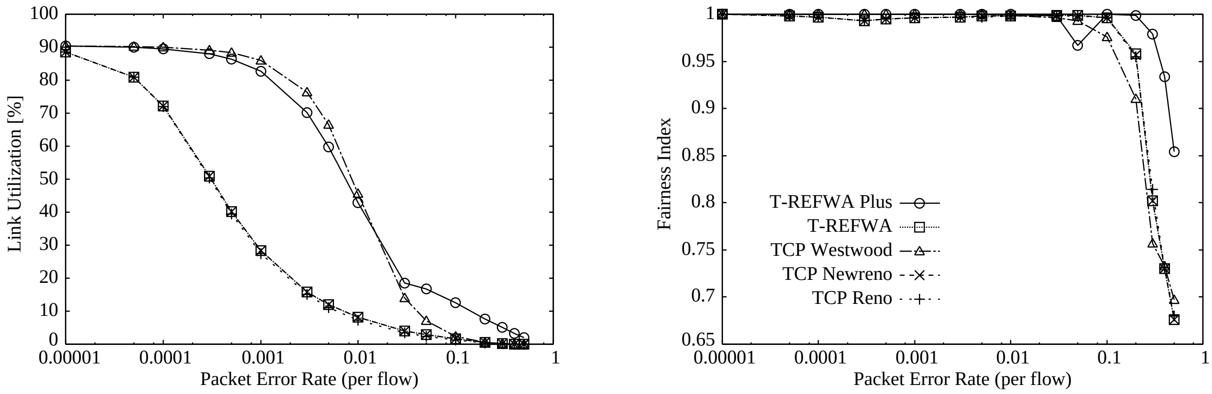
Four TCP variants are used as comparison terms: TCP Reno, TCP Newreno, TCP Westwood, and T-REFWA. As TCP Newreno achieves faster recovery from multiple losses within the same window and has the potential of significantly improving TCP performance over bursty losses, we consider the TCP Westwood based on TCP Newreno. While T-REFWA and T-REFWA Plus can be implemented on any TCP variant, we consider the implementation on TCP Newreno for the same reason as mentioned above.

In all simulations, all sources use the same protocol. All presented results are an average of several simulation runs. The following two indicators are used to evaluate the efficiency and the fairness of the schemes.

- Bottleneck link utilization
  The bottleneck link utilization is the ratio of the aggregate goodputs of all connections to the bottleneck link capacity. Here, the retransmitted packets are not counted. The goodput is thus defined by the highest sequence number of data packet received at the destination times data packet size divided by the simulation time.
- Fairness index [17]
  To investigate the fairness of the schemes, we use the fairness index shown in the following equation.

$$F(x) = \frac{(\sum_{i=1}^{M} x_i)^2}{M \cdot \sum_{i=1}^{M} (x_i)^2} \tag{2}$$

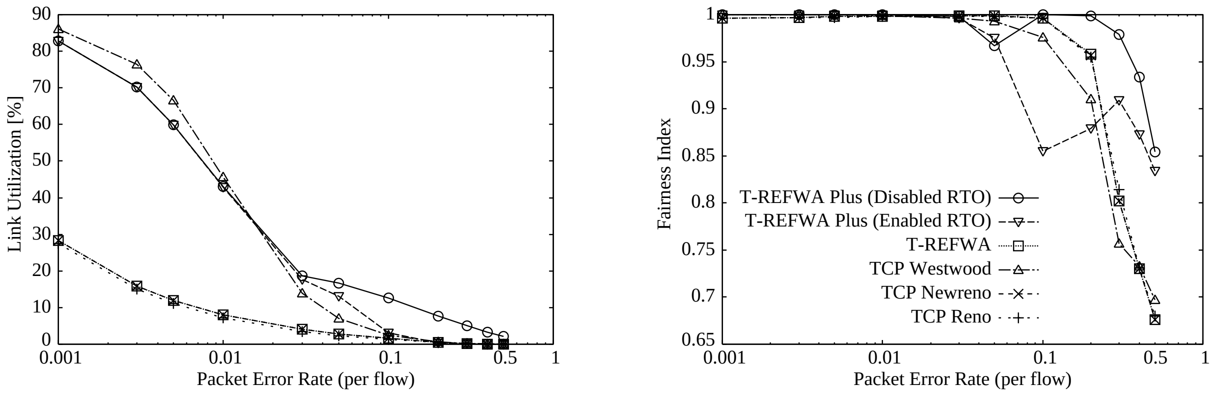
where M and x_i denote the total number of flows and



(a) Bottleneck link utilization

(b) Fairness

Fig. 3. System performance with no packet drops due to congestion (Flows with equal RTTs, $bw = 200\mathrm{Mbps}$, $N = 10$)



(a) Bottleneck link utilization

(b) Fairness

Fig. 4. Effect of the RTO backoff algorithm (Flows with equal RTTs, $bw = 200\mathrm{Mbps}$, $N = 10$)

the actual goodput of the i^{th} flow, respectively. The fairness index ranges from zero to one. Lower values of the fairness index represent poor fairness among the competing flows.

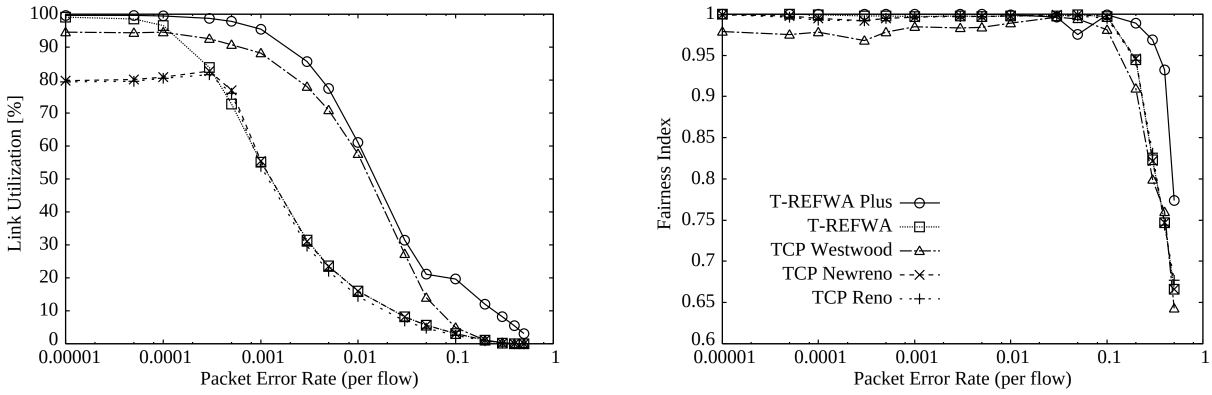
B. Robustness to link errors (No network congestion)

To describe how the proposed scheme is efficient in environment with high bit error rates, we run a simple simulation using the topology as shown in Fig. 2. Only ten users in Group 1 are active and download data from the server. The bottleneck bandwidth bw is set to $200\mathrm{Mbps}$. Since the maximum transmission rate of each flow is set to $18\mathrm{Mbps}$ as mentioned above, the network does not experience congestion and all packet losses are due to link errors in the wireless part. Fig. 3(a) shows the bottleneck link utilization in case of using the five schemes for different Packet Error Rates (PERs). In this figure, the link utilization obtained using T-REFWA Plus and TCP Westwood are always higher than those of T-REFWA, TCP Reno, and TCP Newreno. This is because the former two methods do not reduce their congestion window sizes when senders receive duplicate ACKs generated by packet drops due to link error. But the latter three protocols misinterpret duplicate ACKs as a notification of network congestion and reduce their sending rates by mistake. In high PERs, T-REFWA Plus keeps higher link utilization than the other schemes including TCP Westwood. T-REFWA Plus improves the RTO backoff mechanism so that the RTO value is fixed when timeouts occur frequently.This point will be discussed further in the next section.

Fig. 3(b) graphs the fairness index values for different PERs. Since all mobile users have the same RTT, the fairness index is expected to take high values for all PERs. However, the figure demonstrates that only the T-REFWA Plus keeps high fairness in high PERs (more than 0.1) while the other schemes achieve almost perfect fairness only in case of low PERs. This discrepancy is due to the timeout processes incurred in each protocol as will be explained later.

C. Effect of the RTO backoff algorithm

As discussed earlier, T-REFWA Plus scheme achieves the highest link utilization and the best fairness in high PERs. These improvements are attributable to the modified RTO backoff mechanism. To find out the impact of fixed backoff mechanism, we run the same simulation as above using T-REFWA Plus without freezing the backoff mechanism. Simulation results are plotted in Figs. 4(a) and 4(b). In high PER environments, a TCP sender experiences timeouts frequently. TCP variant without T-REFWA Plus using fixed backoff mechanism have to wait for a long time until the value of



(a) Bottleneck link utilization

(b) Fairness

Fig. 5. Overall performance in terms of link utilization and fairness index (Flows with equal RTTs, $N = 10$, and possible congestion at bottleneck link $bw = 100\mathrm{Mbps}$)

RTO reaches its maximum value $64\mathrm{s}$. Therefore, by fixing the RTO value, the bottleneck link utilization is improved as shown in Fig. 4(a). On the other hand, connections unequally experience timeouts. With standard backoff algorithm, a flow experiencing frequent timeouts sends less packets than a flow with less timeouts. As a result, a significant disproportion in the throughputs of flows manifests and the fairness index decreases as the bit error rate increases. However, since the proposed T-REFWA Plus scheme has fixed backoff mechanism, all connections experience equivalent idle times for RTO. Consequently, only T-REFWA Plus with fixed backoff algorithm could keep high fairness in high PERs as Fig. 4(b) indicates. From these results, it can be said that the fixed backoff algorithm plays an important role in the improvement of TCP performance in high bit error rate environments.

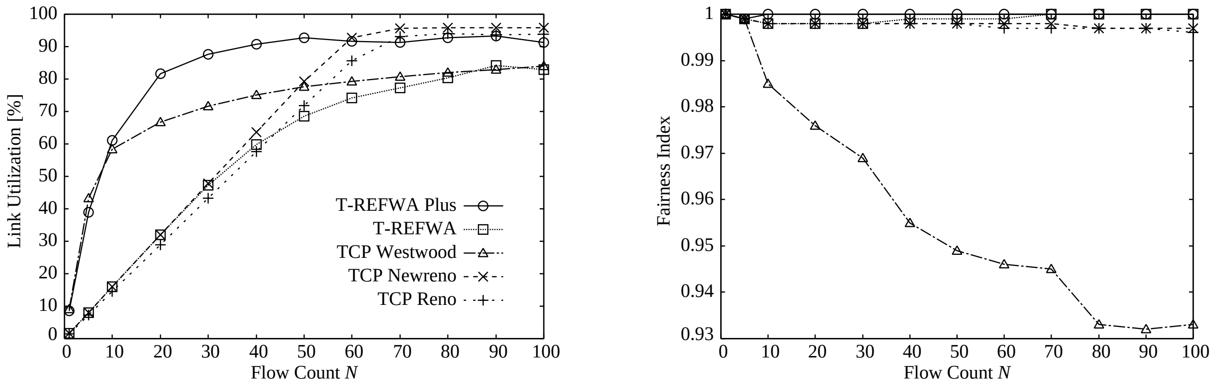
D. Performance when drops are due to both link errors and congestion

Here, we investigate the behavior of the five protocols in general conditions where packets may be dropped due to congestion or link errors. Similarly to the previous simulations, ten flows are set active. The bottleneck link bandwidth bw is set to $100\mathrm{Mbps}$. As the maximum aggregate data traffic generated by all connections is around $180\mathrm{Mbps}$, some packet drops may occur at the bottleneck link. On the other hand, packet losses due to link errors are caused at the last hop to mobile users from the access points.

Figs. 5(a) and 5(b) plot the bottleneck link utilization and the fairness index, respectively. The T-REFWA and T-REFWA Plus schemes outperform the other three methods, TCP Reno, TCP Newreno, and TCP Westwood in low PERs (less than 0.0001). TCP Reno and TCP Newreno reduce their transmission rates when duplicate ACKs are received. Although TCP Westwood controls congestion window size using an estimate of the available bandwidth, it does not make an accurate estimation. Therefore, these three protocols are not able to fully utilize the bottleneck link bandwidth in low bit error rate environments. On the other hand, in T-REFWA scheme, the feedback value is computed in a way that the aggregate throughput of flows matches the available bandwidth-delay product of the network. Consequently, the number of packets dropped at the bottleneck link due to network congestion is almost zero. As a result, the bottleneck link utilization is maintained near 100% in T-REFWA and T-REFWA Plus. However, T-REFWA scheme is not efficient in high PERs and exhibits the same performance as TCP Newreno. This is due to the fact that T-REFWA only limits the maximum sending rate while T-REFWA Plus keeps on transmitting data at feedback values when packet drops occur. The performance of TCP Westwood is better than TCP Reno and TCP Newreno. However, it does not perform as good as T-REFWA Plus in high PERs (more than 0.1). Concerning the system fairness, Fig. 5(b) indicates that T-REFWA Plus scheme achieves the highest fairness in high PERs. This performance is attributed to the fixed RTO backoff algorithm as described in Section IV-C.

E. Influence of the number of users

In this simulation, we focus on the influence of flow counts on the overall performance of each protocol. Simulations are run with PER$= 0.01$, $bw = 100\mathrm{Mbps}$, and a number of active flows with equal RTTs (e.g. Group 1). Fig. 6(a) shows that TCP Reno, TCP Newreno, and T-REFWA do not efficiently utilize the network resources when the number of users is less than 60. In these protocols, the employed window control algorithms reduce the sending rate frequently in case of high values of PER. As a result, the aggregate traffic rate is not enough to fill the network unless the size of packets is increased. On the other hand, T-REFWA Plus and TCP Westwood achieve high link utilizations even in case of few flows. The good performance of TCP Westwood comes, however, at the price of poor fairness. Indeed, Fig. 6(b) indicates that TCP Westwood results in an unfair service when the number of flows increase. This is due to the fact that the bandwidth estimate mechanism of TCP Westwood does not function efficiently when large number of flows compete for the bandwidth of the same bottleneck link. Indeed, the fairness index of TCP Westwood is inferior to that of TCP Reno, TCP Newreno, and T-REFWA Plus when the number of competing flows increase. This is due to the large disproportion that



(a) Bottleneck link utilization (b) Fairness

Fig. 6. System performance for different flow count (PER=0.01, $bw = 100\mathrm{Mbps}$, flows with equal RTTs)

occurs among throughputs achieved by flows despite the fact that they have equal RTTs. To conclude, T-REFWA Plus scheme, equipped with the explicit feedback mechanism and the error recovery function, is able to make efficient utilization of the link bandwidth while maintaining a fair service for all users.

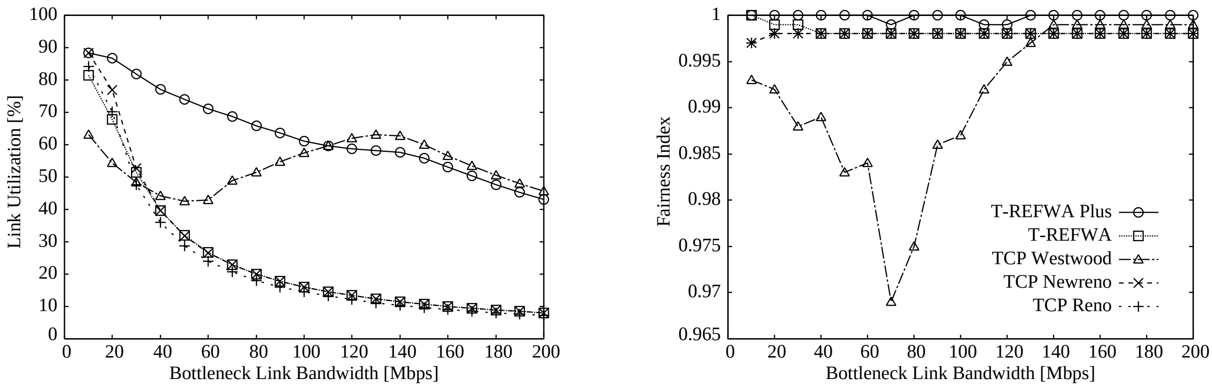
F. Influence of the bottleneck link capacity

In this simulation, the number of sources is set to ten and the PER is equal to 0.01. Due to the high bit error rate, TCP Reno, TCP Newreno, and T-REFWA could not make efficient use of the network resources even in broadband environments as shown in Fig. 7(a). On the other hand, TCP Westwood and T-REFWA Plus exhibit better link utilization. Although TCP Westwood slightly exceeds the T-REFWA Plus in terms of the link utilization when no congestion occurs in the network due to wide bandwidth, its performance is limited in terms of both link utilization and fairness (Fig. 7(b)) when the network gets congested due to narrow bandwidth. TCP Westwood performs poorly when bw is set to less than $40\mathrm{Mbps}$ due to errors in bandwidth estimation. In contrast to the fluctuations of TCP Westwood, the performance of T-REFWA Plus is stable when the bottleneck link capacity is varied. The proposed scheme adjusts its window size based on feedback rates accurately computed at the bottleneck router. Therefore, the T-REFWA Plus scheme achieves high link utilization and near-perfect fairness for all the simulated capacities.

G. Performance in case of flows with different RTTs

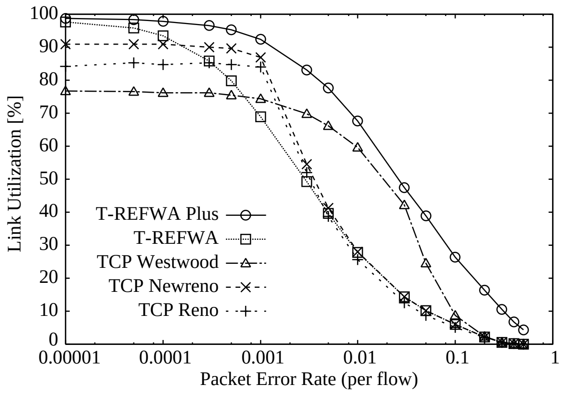
In the remainder of this section, we discuss how each protocol works when flows have high variance in their RTT distribution. The network configuration is shown in Fig. 2. The bottleneck link bandwidth is set to $100\mathrm{Mbps}$. The groups are simulated and each group consists of ten mobile users. The bottleneck link utilization and fairness index of each protocol are plotted for different values of PER in Figs. 8(a) and 8(b), respectively. Comparing Figs. 8(a) and 5(a), it can be observed that the five protocols exhibit the same performance. This demonstrates that the RTT distribution does not largely affect the link utilization. T-REFWA Plus significantly improves TCP performance in throughput while TCP Westwood degrades its performance due to the bandwidth estimation errors, particularly in low PERs.

As for the system fairness, as flows have different RTTs, flows with smaller RTTs gain more bandwidth compared to flows with larger RTTs. For this reason, in low PER, the fairness of TCP Reno, TCP Newreno, and TCP Westwood is below that of T-REFWA and T-REFWA Plus schemes that assign the bottleneck bandwidth to competing flows at a rate

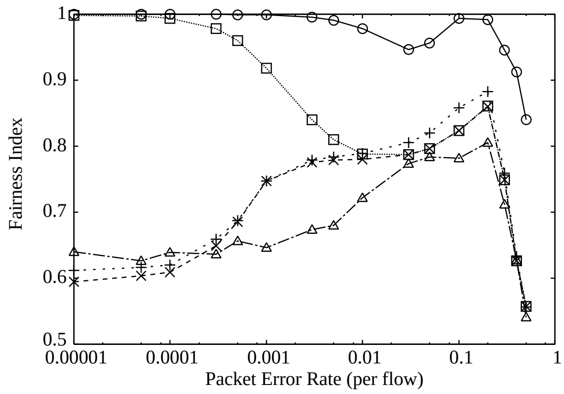


(a) Bottleneck link utilization (b) Fairness

Fig. 7. System performance for different bottleneck link bandwidths (PER=0.01, $N = 10$, flows with equal RTTs)



(a) Bottleneck link utilization



(b) Fairness

Fig. 8. Overall performance in terms of link utilization and fairness index (Flows with different RTTs, $bw = 100\text{Mbps}$, $N = 10$)

proportional to their RTT value. In high PER, the fairness of T-REFWA however degrades. In deed, since the congestion window increment algorithm of T-REFWA is similar to that of TCP Newreno, a long RTT sender can not immediately increase its transmission rate to the assigned bandwidth when a packet loss occurs due to link errors. As a result, the fairness of T-REFWA degrades significantly while the fairness of T-REFWA Plus using the improved window adjustment mechanism, remains relatively acceptable. In addition, the graphs indicate significant fluctuations in the fairness index of all protocols when PER is in the vicinity of 0.2. In high bit error rate, all flows experience heavy packet losses regardless of whether the flow has long or short RTT. The congestion window size does not increase even if the flow has short RTT. Then the impact of unfairness issue resulting from RTT distribution is reduced. But with more larger PERs, the fairness rapidly degrades for all the five protocols because the RTO value depends on the RTT value and flows have to wait for long idle times proportional to their RTT values. From the above results, it can be concluded that the T-REFWA Plus scheme achieves the highest fairness as well as the most efficient link utilization compared to the other schemes even in environments with significantly high bit error rates.

V. Conclusion

In this paper, we proposed a scheme for mitigating the impact of link errors in hybrid wired/wireless networks. The proposed scheme is an enhancement of the recently-proposed T-REFWA. Although T-REFWA achieves efficient link utilization and high fairness among competing flows in wired networks, its performance remains limited in wireless networks with link errors. To cope with this issue, the proposed T-REFWA Plus scheme controls the transmission rate of sources with more aggressiveness based on the optimal rate signaled by the T-REFWA mechanism. By so doing, the proposed scheme improves the bottleneck link utilization and the system fairness as verified by extensive simulations. The obtained results are encouraging and promising for the provision of different Internet-based applications over hybrid wired/wireless networks with some reasonable bit error rates.

References

[1] T. Taleb, H. Nishiyama, A. Jamalipour, N. Kato, and Y. Nemoto, "A Fair TCP-Based Congestion Avoidance Approach for One-to-Many Private Networks," in *Proc. of IEEE ICC*, Istanbul, Turkey, Jun. 2006.

[2] C. Parsa and J. J. Garcia-Luna-Aceves, "Improving TCP Congestion Control over Internets with Heterogeneous Transmission Media," in *Proc. of IEEE ICNP*, Toronto, Canada, Oct./Nov. 1999, pp. 213–221.

[3] P. Sinha, T. Nandagopal, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," *Wireless Networks*, vol. 8, no. 2–3, pp. 301–316, Mar. 2002.

[4] 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, Oct. 1995.

[5] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," in *Proc. of ACM MOBICOM*, Rome, Italy, Jul. 2001, pp. 287–297.

[6] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options," IETF, RFC 2018, Oct. 1996.

[7] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," in *Proc. of ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002, pp. 89–102.

[8] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "Explicit window adaptation: a method to enhance TCP performance," *IEEE/ACM Transactions on Networking*, vol. 10, no. 3, pp. 338–350, Jun. 2002.

[9] J. Aweya, M. Ouellette, D. Y. Montuno, and Z. Yao, "WINTRAC: a TCP window adjustment scheme for bandwidth management," *Performance Evaluation*, vol. 46, no. 1, pp. 1–44, Sep. 2001.

[10] T. Taleb, N. Kato, and Y. Nemoto, "A Recursive, Explicit and Fair Method to Efficiently and Fairly Adjust TCP Windows in Satellite Networks," in *Proc. of IEEE ICC*, Paris, France, Jun. 2004, pp. 4268–4274.

[11] T. Taleb, N. Kato, and Y. Nemoto, "Recent Trends in IP/NGEO Satellite Communication Systems: Transport, Routing, and Mobility Management Concerns," *IEEE Wireless Communications*, vol. 12, no. 5, pp. 63–69, Oct. 2005.

[12] T. Taleb, N. Kato, and Y. Nemoto, "REFWA Plus: Enhancement of REFWA to Combat Link Errors in LEO Satellite Networks," in *Proc. of IEEE HPSR*, Hong Kong, China, May 2005, pp. 172–176.

[13] T. Taleb, N. Kato, and Y. Nemoto, "REFWA: An Efficient and Fair Congestion Control Scheme for LEO Satellite Networks," *to appear in IEEE/ACM Trans. on Networking*, Oct. 2006.

[14] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," *Wireless Networks*, vol. 8, no. 2–3, pp. 275–288, Mar. 2002.

[15] The Network Simulator - ns-2 [Online]. Available: http://www.isi.edu/nsnam/ns/.

[16] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[17] R. Jain, D.-M. Chiu, and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems," DEC, Tech. Rep. 301, Sep. 1984.