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

Hiroki Nishiyama, Tarik Taleb, Yoshiaki Nemoto, Abbas Jamalipour, 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 appropriate 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 disabling 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:*** **Congestion control, hybrid wired/wireless network, link error, transmission control protocol (TCP), T-REFWA.**

## I. INTRODUCTION

Along with the rapid globalization of the mobile telecommunications industry, Internet services have 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 4-th 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. Given the current dominance of the transmission control protocol (TCP) among IP data transmission protocols, effects of wireless link errors on TCP should be investigated [1]. This forms the focus of the research work outlined in this paper.

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 time (RTT) distribution. As an attempt to solve the unfairness issue of TCP, the authors have proposed the T-REFWA [2] scheme. In this paper, we further enhance the working of T-REFWA by extending its design to environments with link errors. With the proposed enhancements, 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 work 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 the 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 [3], wireless transmission control protocol (WTCP) [4], TCP Vegas [5], and TCP Westwood [6]). 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 conges-

Manuscript received May 2, 2006.

H. Nishiyama, T. Taleb, Y. Nemoto, and N. Kato are with the Graduate School of Information Sciences, Tohoku University, Sendai, Japan, email: 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, email: a.jamalipour@ieee.org.

tion is increasing or decreasing in both the forward and reverse paths of the connection based on the inter-arrival time of data packets at the receiver side. 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 [7], 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 accordingly adjusts the congestion window. When the actual throughput largely deviates from the expected throughput, the sender linearly decreases the congestion window 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 linearly increases the congestion window during the next RTT as this indicates more available bandwidth at the network. By doing so, 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, 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 rates. 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. According to analysis in [8], rate estimation-based congestion control schemes relying on ACK arrival rates, such as TCP Westwood, can result in significant overestimation of forward bottleneck link capacity when the backward path experiences congestion.

Similar to TCP Westwood, TCP New Jersey [9] proactively adapts the sending rate using available bandwidth estimation (ABE) algorithm. Meanwhile, TCP New Jersey has another algorithm, referred to as congestion warning (CW), which enables routers to alert end terminals by marking all packets, similar to explicit congestion notification (ECN) [10], when the average queue length exceeds a given threshold. A TCP New Jersey sender refers to CW to distinguish packet losses caused by network congestion from those caused by wireless link errors and accordingly adjusts its transmission rate using ABE.

While most enhancements to TCP carry out the transmission rate calculation at end terminals, some other approaches have considered the equipment of network elements, such as routers and gateways, with rate controllers. Explicit control protocol (XCP) [11], explicit window adaptation (EWA) [12], and window tracking and computation (WINTRAC) [13] are few notable examples. In these methods, network elements along the path of a TCP source to a TCP destination signal the most appropriate transmission rates (window sizes) to the source. These schemes are efficient in making full use of network resources. However, apart from XCP which assumes a pure XCP network and requires significant modifications at the end-system, they 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) [1], [14], [15] 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 appropriate 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 with bit error rates. 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 into an integer value, $\alpha$, within the range $[0, 63]$. For a detailed description of this transformation procedure, the interested reader is referred to [2]. At routers, flows are grouped according to their RTT indicator $\alpha$. Each group is defined as the set of flows having the same $RTT_\alpha$ 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 of all active flows traversing the router, $RTT_{avg}$.

Similar 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 $\alpha$, $W_\alpha(n)$, is computed as in (1) (shown at the bottom of the page). In (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 $(t = n \cdot RTT_{avg})$. $\phi$ and $\psi$ are constant parameters. It should be recognized that $\phi$ and $\psi$ play a significant role in exploiting well the unused bandwidth and free buffer size, respectively. In case of optimum setting of $\phi$ and $\psi$, changes in flows' RTTs and packet drops may happen only when the number of flows sharing the same bottleneck link increases or decreases. Upon such an occurrence, the system returns back immediately to its steady state where all flows exhibit stable RTTs and no packet is dropped. Details on the setting of $\phi$ and $\psi$, and on the working of the T-REFWA scheme can be found at [2] and [16]. Similar to the original REFWA scheme, $\phi$ and $\psi$ are set to 1.5 and 0.5, respectively.

So far, we have considered the case of only TCP flows. However, our studies can be easily extended to more general scenarios where nonresponsive traffic such as user datagram protocol (UDP) coexists with TCP flows. In such case, by using classification based on port number and per-class queuing, application of the proposed scheme is relevant to only the TCP queuing class.

### *B. Major Operations 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 receiver's advertised window (RWND) field of ACK packets and computed at bottleneck links. 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 although it may be slightly affected by changes in queue length due to link errors. On the other hand, when packet drops occur due to network congestion, flows experience changes in their RTTs and the queue length of the bottleneck link dynamically fluctuates. Therefore, the feedback value gets largely degraded due to network congestion during the reception of duplicate ACKs or expiration of timeout. Using this feature, it is possible to predict the cause of packet losses by tracking changes in feedbacks signaled by ACKs. The remainder of this section explains how this is implemented in T-REFWA plus.

Upon reception of an ACK packet, a T-REFWA plus sender compares the feedback value $W_c$ signaled by the currently received ACK and the previous feedback value $W_p$ recorded by the sender. If the ratio of $W_c$ to $W_p$ is less than a threshold $r$, the sender records the occurrence time of this event as $t_d$. Hence, $t_d$ refers to the most recent time when

$$\frac{W_c}{W_p} < r \tag{2}$$

where, $r$ is a parameter within [0,1]. The threshold $r$ decides how much decrease in feedback values can be a sign of network congestion. This process is the first judgment for distinguishing packet drops due to congestion from packet losses due to link errors.

If $t_d$ is $k \cdot RTT$ to or past the reception time of the third duplicate ACK, the sender judges the packet loss event as due to congestion. In contrast, if a sign of network congestion is not found, the sender interprets duplicate ACKs as a result of a link error. This second judgment condition is summarized in Fig. 1 with the first judgment condition. After a packet loss is judged as due to link error, the sender exits the congestion avoidance phase. In such case, congestion window ($cwnd$) and slow start threshold ($ssthresh$) are set equal to the most recently signaled feedback. On the other hand, the sender remains in the congestion avoidance phase as in case of standard TCP if the network is judged congested. As shown in Fig. 2, this operation enables the TCP sender to promptly return back to the congestion window size ($W$) prior to the reception of duplicate ACKs. Hence, in case of packet losses due to link errors, the time till $W$ is

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

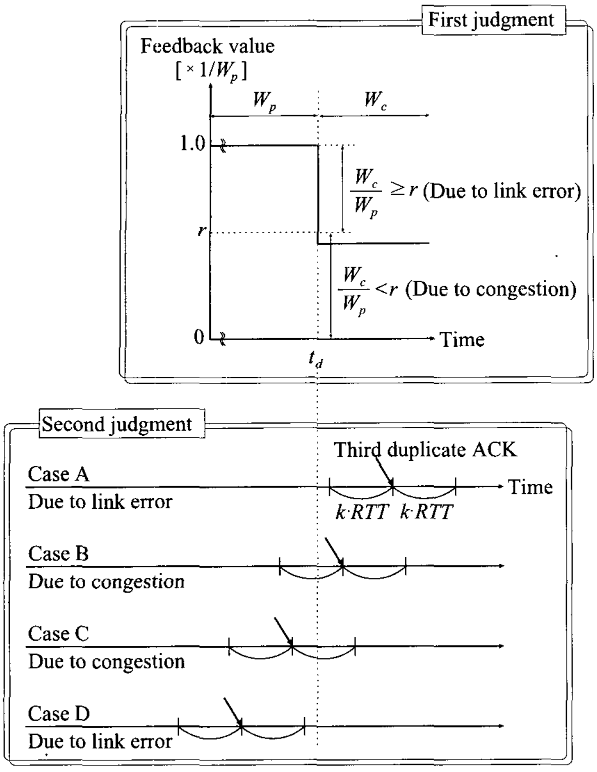



Fig. 1. Judgment conditions in the first and second steps.

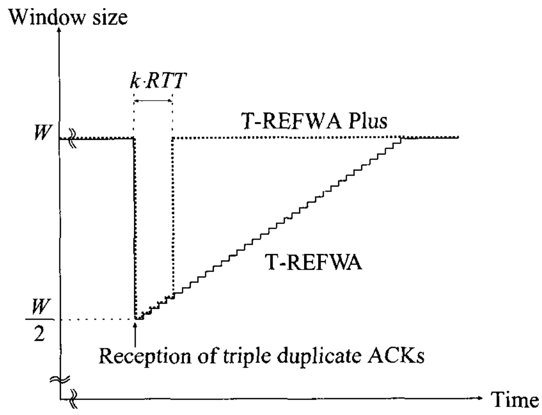



Fig. 2. Window size variations in T-REFWA and T-REFWA plus.

reached becomes $k \cdot RTT$ in contrast to $(RTT \cdot W/2)$ in case of original T-REFWA or standard TCP. It is clear from Fig. 2 that the parameter $k$ indicates the time interval during which the sender remains in congestion avoidance even if a packet loss is due to link error.

Upon occurrence of timeout, the sender checks whether any ACK packet has been received and whether condition of (2) happened $k \cdot RTT$ before timeout occurs. To trace the arrival of ACK packets, the time value $t_a$ which indicates the arrival time of the most recent ACK packet is held by the sender. If ACK packets reach the sender and network congestion is not inferred, $cwnd$ and $ssthresh$ are set to $W_p$ in the same manner as in case of duplicate ACKs due to link errors. Moreover, in order to improve TCP performance in heavy loss environments, T-REFWA plus freezes the RTO backoff algorithm. This algorithm, used in most TCP variants, doubles the RTO value when coarse timeouts occur in succession. In the 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 [17]. The major procedures of T-REFWA plus are illustrated in Fig. 3.

As stated above, the performance of T-REFWA plus depends on the setting of $r$ and $k$. Indeed, small values of $r$ might cause misinterpretation of a packet loss due to network congestion as due to link errors, and vice versa. As for $k$, high values of $k$ make a sender wait for a long time till it adjusts its congestion window to the signaled feedback, while small values of $k$ may affect the accuracy of the proposed error recovery mechanism in finding out the real cause of a packet loss. Considering these trade-offs, in our simulations, $r$ and $k$ are set to 0.9 and 3, respectively. 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) [18]. We consider a simple network topology with a single bottleneck link as shown in Fig. 4. 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 20 Mbps to 200 Mbps. Wireless link delays are almost zero and each group has different propagation delays. 60 ms, 120 ms, and 180 ms are set for Groups 1, 2, and 3, respectively. All groups consist of equal number of mobile users and the size of each group $N$ is varied from one to 50. To avoid bursty drops at the simulation launch time, all mobile users are randomly 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 17.5 Mbps in Group 1 (with the smallest RTT $=$ 60 ms). 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 Newreno with random early detection (RED) [19] as packet discarding policy. Simulations are all run for 100s. The packet loss probability for link errors in the wireless part is varied within the range $[10^{-5}, 0.5]$. Table 1 shows a complete list of the simulation parameters and the range of their values.

Three TCP variants are used as comparison terms: TCP Newreno, TCP Westwood, and T-REFWA. As TCP Newreno achieves faster recovery from multiple losses within the same

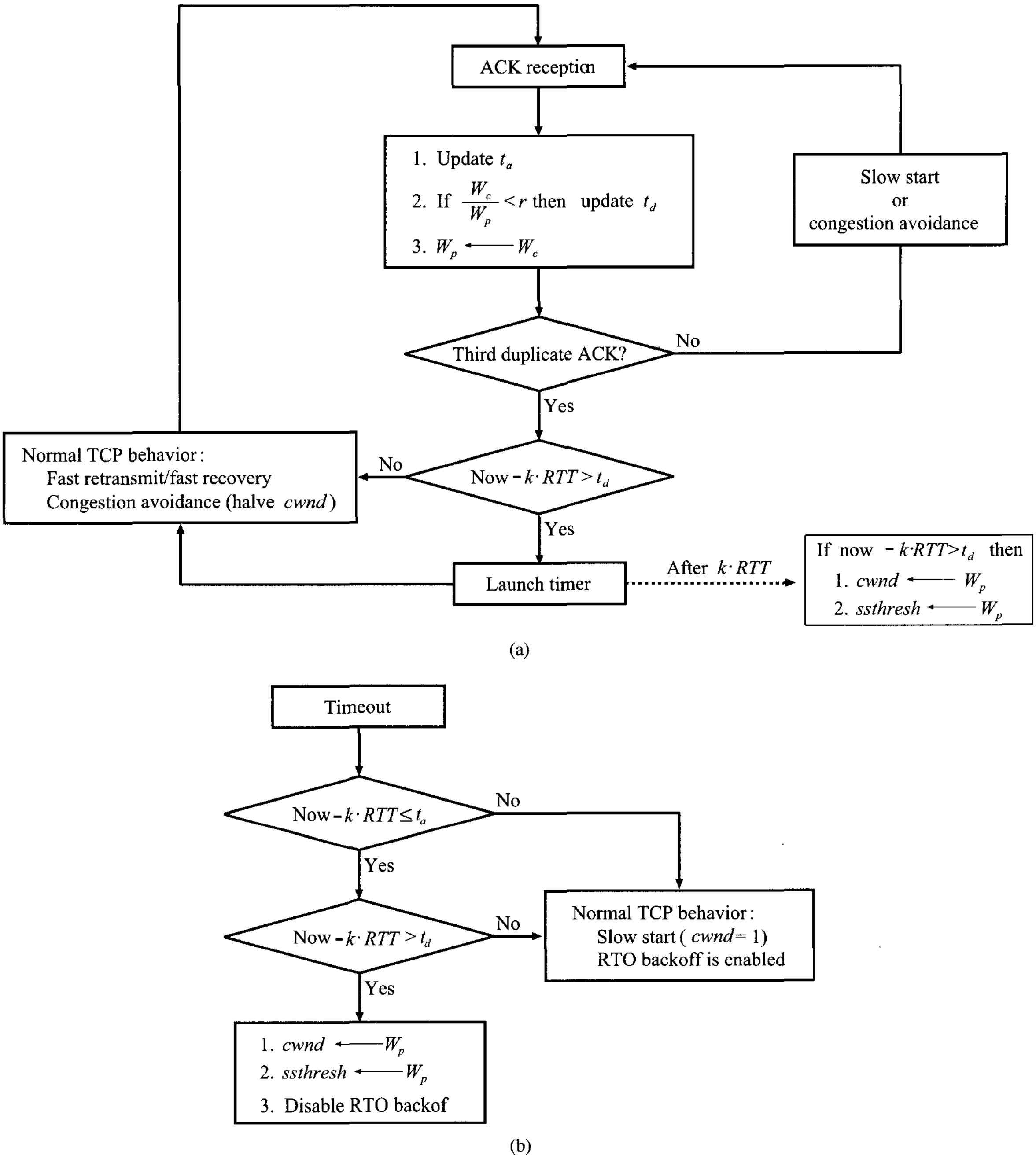



Fig. 3. Major procedures in the T-REFWA plus scheme; (a) ACK reception flowchart, (b) timeout flowchart.

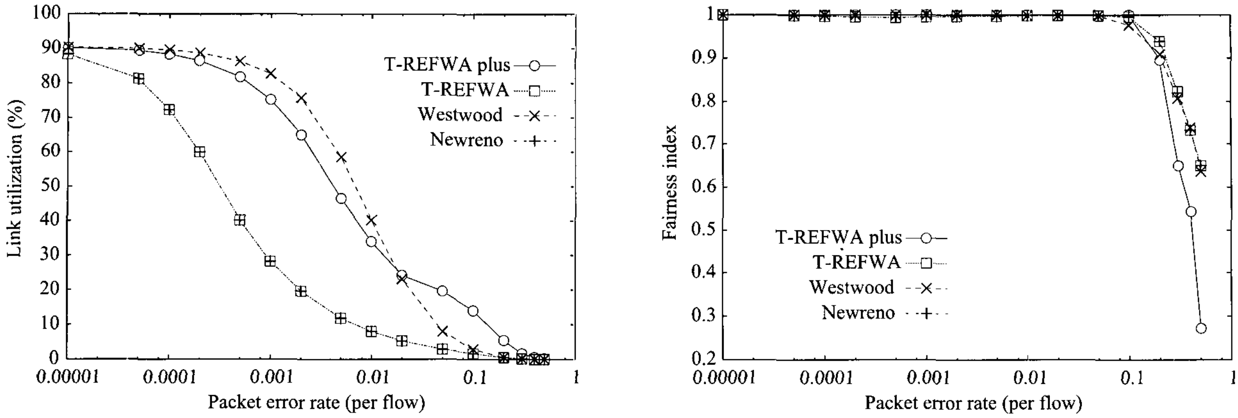
window and has the potential of significantly improving TCP performance over bursty losses, we consider the TCP Westwood based on TCP Newreno. While T-REFWA and T-REFWA plus can be implemented on any TCP variant, we consider their implementations on TCP Newreno for the same reason as mentioned above.

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 [20]:
  To investigate the fairness of the schemes, we use the fairness index shown in the following equation.

$$F(x) = \frac{(\sum_{i=1}^{M} x_i)^2}{M \sum_{i=1}^{M} (x_i)^2} \tag{3}$$

where $M$ and $x_i$ denote the total number of flows and the ac-




(a) (b)

Fig. 5. System performance with no packet drops due to congestion (flows with equal RTTs, $bw = 200$ Mbps, $N = 10$); (a) bottleneck link utilization, (b) fairness.

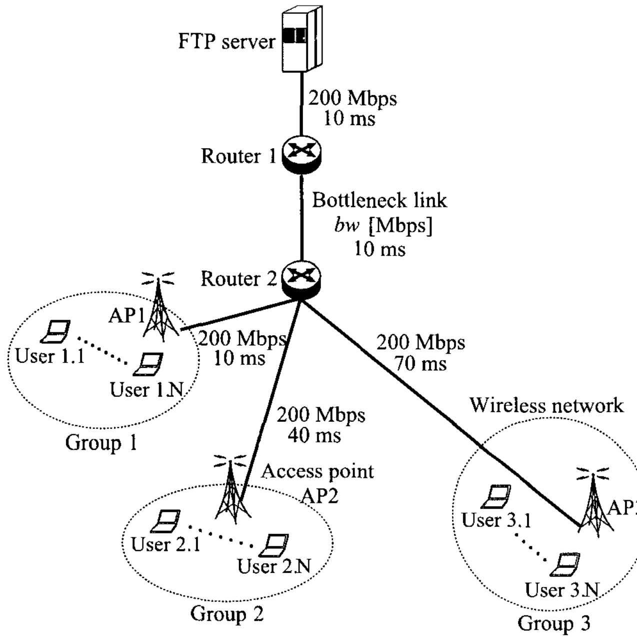



Fig. 4. Simulation topology.

Table 1. Simulation parameters.

| Parameters | Range of values |
|---|---|
| Bottleneck link capacity $bw$ | 20 Mbps – 200 Mbps |
| RTTs of each group | 60 ms, 120 ms, 180 ms |
| Packet size | 1024 bytes |
| RWND | 128 packets |
| Size of each group $N$ | 1 – 50 |
| Packet error rate (PER) | $10^{-5}$ – 0.5 |
| Simulation run time | 100 s |

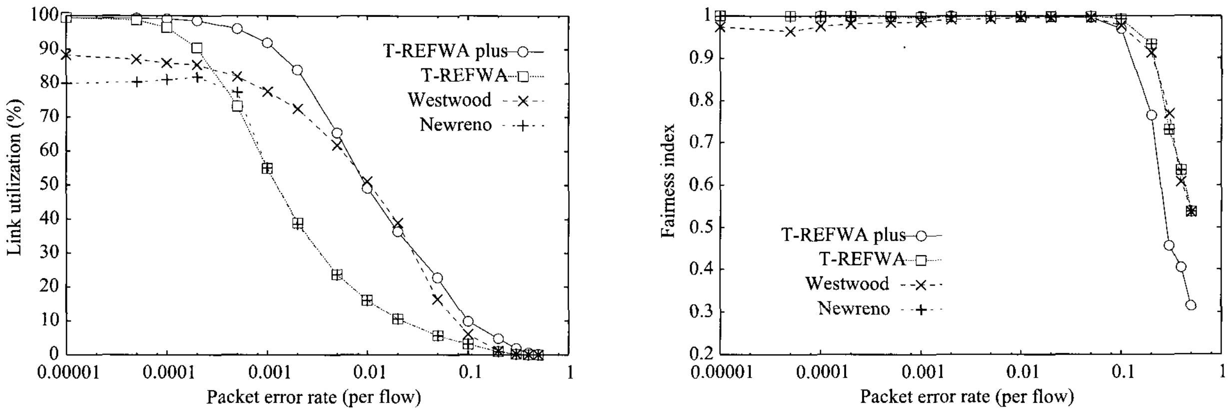
tual goodput of the $i$-th flow, respectively. The fairness index ranges from zero to one. Lower values of the fairness index represent poor fairness among the competing flows.

## *B. Robustness to Link Errors (No Network Congestion)*

To describe how the proposed scheme is efficient in environments with high bit error rates, we run a simple simulation using the topology as shown in Fig. 4. Only ten users in Group 1 are active and download data from the server. The bottleneck bandwidth $bw$ is set to 200 Mbps. Since the maximum transmission rate of each flow is set to 17.5 Mbps as mentioned above, the network does not experience congestion and all packet losses are due to link errors in the wireless part.

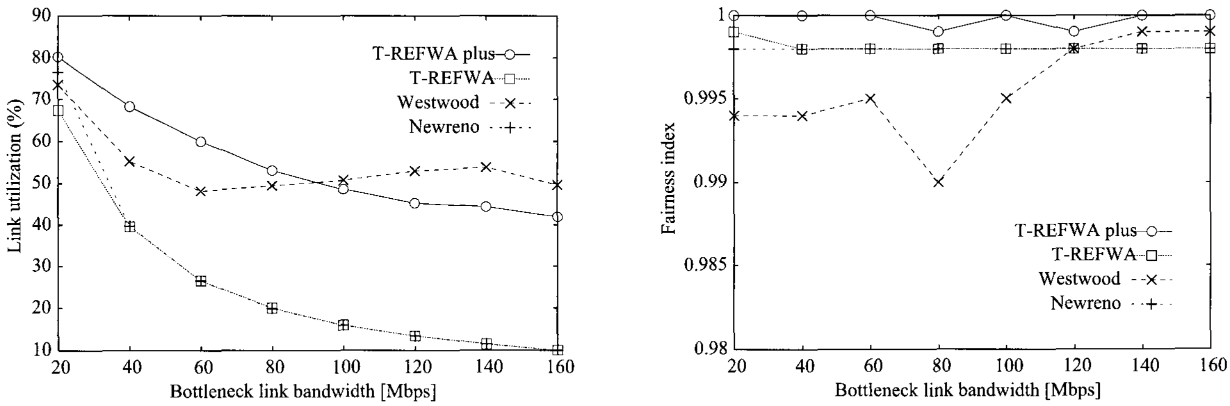
Fig. 5(a) shows the bottleneck link utilization in case of using the four schemes for different packet error rates (PERs). In this figure, the link utilization obtained using T-REFWA plus and TCP Westwood are always higher than that of T-REFWA and TCP Newreno. This is because the former two methods do not halve their congestion window sizes when senders receive duplicate ACKs generated by packet drops due to link errors (only T-REFWA plus temporarily halves its congestion window size). But the latter two protocols misinterpret duplicate ACKs as a notification of network congestion and reduce their sending rates by mistake. In the former two protocols, it is shown that TCP Westwood slightly outperforms the proposed scheme which temporarily decreases its sending rate after the reception of duplicate ACKs. But the advantage of TCP Westwood comes from the non existence of network congestion. We discuss about this point in the following subsections.

In heavy PERs, we can see that T-REFWA plus achieves higher link utilization than the other schemes. This is due to the fact that T-REFWA plus disables the RTO backoff mechanism so that the source is able to send new packets without waiting for a long time. In T-REFWA plus scheme, only the flows which consider that a timeout is caused by a link error, promptly increase their sending rates. So, the fairness of the proposed scheme gets lower values in heavy PERs as shown in Fig. 5(b). In other schemes, because timeouts cause a diverse distribution in throughput due to backoff algorithm, the fairness index decreases in high PERs. However, this degradation is not an issue as achieving high throughput is more critical than fair-




(a) (b)

Fig. 6. Overall performance in terms of link utilization and fairness index (flows with equal RTTs, $N = 10$, and possible congestion at bottleneck link $bw = 100$ Mbps); (a) bottleneck link utilization, (b) fairness.




(a) (b)

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

ness in case of high PER.

### C. Performance When Drops Are Due to Both Link Errors and Congestion

Here, we investigate the behavior of the four 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 Mbps. As the maximum aggregate data traffic generated by all connections is around 175 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. 6(a) and 6(b) plot the bottleneck link utilization and the fairness index, respectively. Fig. 6(a) indicates that TCP Newreno and TCP Westwood do not achieve efficient link utilization in low PERs. TCP Newreno reduces its congestion window size when duplicate ACKs are received and so a long time is required to reach the transmission rate prior to the reception of duplicate ACKs. Although TCP Westwood controls congestion window size using an estimate of the available bandwidth, it fails to make an accurate estimation. Therefore, TCP Westwood is not able to fully utilize the bottleneck link bandwidth even in low bit error rate environments.

On the other hand, The T-REFWA plus and T-REFWA schemes achieve near perfect performance in low PERs. In these protocols, the transmission rate of each flow is computed based on the bandwidth-delay product of the network and the buffer occupancy in a way that the aggregate throughput of flows matches the available network resources. Consequently, the number of packets dropped at the bottleneck link due to network congestion becomes almost zero and the bottleneck link utilization is maintained near 100 %. 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 immediately gets back to its previous sending rate when packet drops are due to link errors.

### D. Influence of the Bottleneck Link Capacity

To clarify how packet drops due to network congestion affect the TCP performance in wireless environments with link

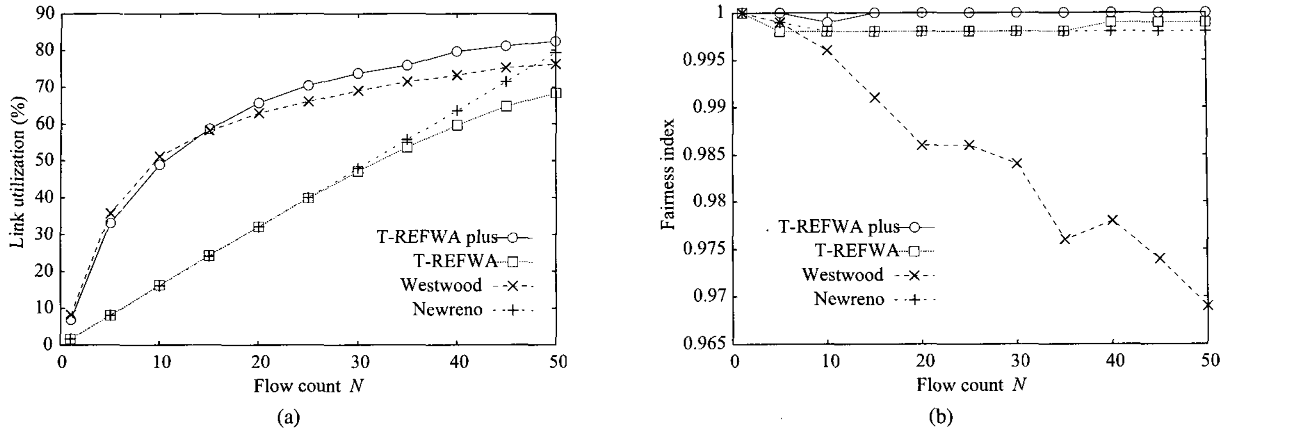



Fig. 8. System performance for different flow count (PER $= 0.01$, $bw = 100$ Mbps, flows with equal RTTs); (a) bottleneck link utilization, (b) fairness.

errors, we investigate the performance of each protocols with changes in the bottleneck link bandwidth $bw$. $bw$ is decreased from 160 Mbps to 20 Mbps. The PER is set equal to 0.01 where TCP Westwood and T-REFWA plus achieve the highest link utilization under $bw = 100$ Mbps as shown in Fig. 6(a). The number of sources is set to ten and all sources belong to Group 1.

Due to the high bit error rate, TCP Newreno and T-REFWA could not make efficient use of the network resources even in broadband environments as shown in Fig. 7(a). Although TCP Westwood exceeds the T-REFWA plus in terms of 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. Indeed, due to lack of buffer size, TCP Westwood becomes unstable and performs poorly in narrow bandwidth. To stabilize the working of TCP Westwood, large queue buffers are required.

In contrast to the fluctuations of TCP Westwood which adjusts the transmission rate based on the estimated available bandwidth, the performance of T-REFWA plus is stable when the network congestion level is changed by varying the bottleneck link capacity. 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.

### *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$= 0.01$, $bw = 100$ Mbps, and a number of active flows with equal RTTs (e.g., Group 1). Fig. 8(a) shows that TCP Newreno and T-REFWA do not efficiently utilize the network resources. In these protocols, the employed window control algorithms frequently reduce the sending rate 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, the other two schemes achieve high link utilizations even in case of few flows. The good performance of TCP Westwood comes, however, at the price of poor fairness. In TCP Westwood employing rate estimation algorithm based on inter-arrival time of ACK packets, the network has to increase its buffer size when the number of traversing flows increases in order to remain stable. Otherwise, the algorithm becomes unstable and leads to a diverse distribution in throughput [8]. This is the reason why the fairness index of TCP Westwood decreases when the number of flows increases despite the fact that they have equal RTTs as shown in Fig. 8(b). 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. 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. 4. The bottleneck link bandwidth is set to 100 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. 9(a) and 9(b), respectively. Comparing Figs. 6(a) and 9(a), it can be observed that the four protocols exhibit similar performance. This demonstrates that the RTT distribution does not largely affect the link utilization. Throughput degradation of TCP Westwood in low PERs is due to unstable issue caused by larger number of users sharing the bottleneck link as described in Section IV-E.

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 PERs, the fairness of TCP Newreno and TCP Westwood are below that of T-REFWA and T-REFWA plus schemes which assign the bottleneck bandwidth to competing flows at a rate proportional to their RTT values. In high PERs, the fairness of T-REFWA however degrades. Indeed, 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.

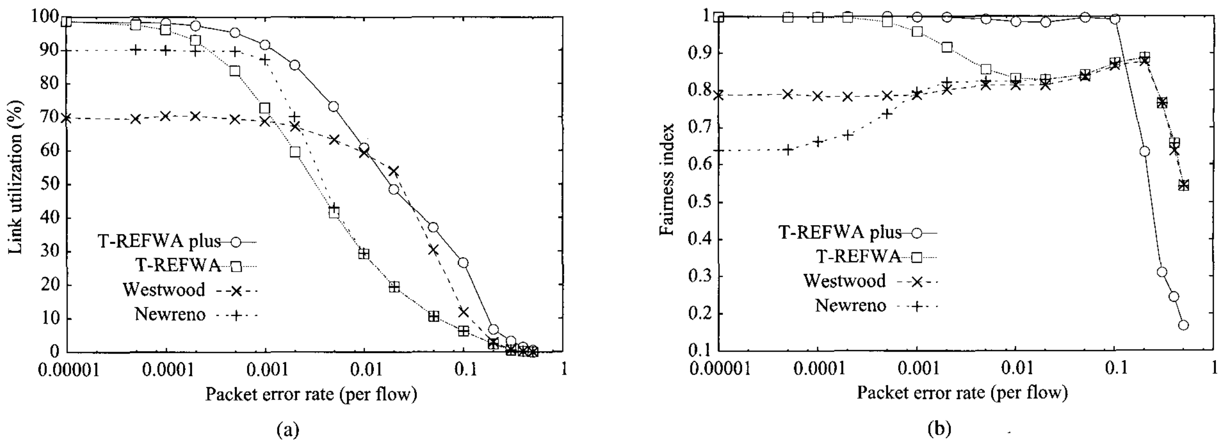



Fig. 9. Overall performance in terms of link utilization and fairness index (flows with different RTTs, $bw$ = 100 Mbps, $N$ = 10, total flows = 30); (a) bottleneck link utilization, (b) fairness.

In addition, the graphs indicate slight increases in the fairness index of all protocols except for T-REFWA plus when PER is around 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 higher PERs, the fairness rapidly degrades due to coarse timeout. 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 appropriate 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, N. Kato, and Y. Nemoto, "Recent trends in IP/NGEO satellite communication systems: Transport, routing, and mobility management concerns," *IEEE Wireless Commun.*, vol. 12, no. 5, pp. 63–69, Oct. 2005.
[2] 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. IEEE ICC 2006*, Istanbul, Turkey, Jun. 2006.
[3] C. Parsa and J. J. Garcia-Luna-Aceves, "Improving TCP congestion control over Internets with heterogeneous transmission media," in *Proc. IEEE ICNP'99*, Toronto, Canada, Oct./Nov. 1999, pp. 213–221.
[4] 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.
[5] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," *IEEE J. Select. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.
[6] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. ACM MOBICOM 2001*, Rome, Italy, Jul. 2001, pp. 287–297.
[7] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options," RFC 2018, IETF, Oct. 1996.
[8] K. Xu and N. Ansari, "Stability and fairness of rate estimation-based AIAD congestion control in TCP," *IEEE Commun. Lett.*, vol. 9, no. 4, pp. 378–380, Apr. 2005.
[9] K. Xu, Y. Tian, and N. Ansari, "Improving TCP performance in integrated wireless communications network," *Computer Networks*, vol. 47, no. 2, pp. 219–237, Feb. 2005.
[10] K. K. Ramakrishnan, S. Floyd, and D. L. Black, "The addition of explicit congestion notification (ECN) to IP," RFC 3168, IETF, Sept. 2001.
[11] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proc. ACM SIGCOMM 2002*, Pittsburgh, PA, Aug. 2002, pp. 89–102.
[12] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "Explicit window adaptation: A method to enhance TCP performance," *IEEE/ACM Trans. Networking*, vol. 10, no. 3, pp. 338–350, Jun. 2002.
[13] 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, Sept. 2001.
[14] 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. IEEE ICC 2004*, Paris, France, Jun. 2004, pp. 4268–4274.
[15] T. Taleb, N. Kato, and Y. Nemoto, "REFWA plus: Enhancement of REFWA to combat link errors in LEO satellite networks," in *Proc. IEEE HPSR 2005*, Hong Kong, China, May 2005, pp. 172–176.
[16] T. Taleb, N. Kato, and Y. Nemoto, "REFWA: An efficient and fair congestion control scheme for LEO satellite networks," *IEEE/ACM Trans. Networking*, Oct. 2006.
[17] 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.
[18] The Network Simulator–ns-2 [online], available at http://www.isi.edu/nsnam/ns/.
[19] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.
[20] R. Jain, D.-M. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report, Tech. Rep. 301, Sept. 1984.

**Hiroki Nishiyama** received his B.E. from Tohoku University, Sendai, Japan in 2005. He has been engaged in research on congestion control protocols, TCP performance, and network security. He is currently working toward his M.S. degree at the Graduate School of Information Sciences (GSIS), Tohoku University. He is an IEEE student member and a member of the Information Processing Society of Japan.

**Tarik Taleb** is currently working as assistant professor with the Graduate School of Information Sciences (GSIS), Tohoku University, Japan. From Oct. 2005 till Mar. 2006, he was working as research fellow with the Intelligent Cosmos Research Institute, Sendai, Japan. He received his B.E. degree in Information Engineering with distinction, M.E. and Ph.D. degrees in Computer Sciences from GSIS, Tohoku Univ., in 2001, 2003, and 2005, respectively. His research interests lie in the field of wireless networking, satellite and space communications, congestion control protocols, and network security. Dr. Taleb is on the editorial board of the IEEE Wireless Communications Magazine. He serves also as secretary of the Satellite and Space Communications Technical Committee of the IEEE Communications Society (ComSoc). He has been on the technical program committee of different IEEE conferences, including Globecom, ICC, and WCNC, and chaired some of their sessions. He is a recipient of the Niwa Yasujirou Memorial award (Feb. 2005) and the Young Researcher's Encouragement award from the Japan chapter of the IEEE Vehicular Technology Society (VTS) (Oct. 2003). Dr. Taleb is an IEEE member.

**Yoshiaki Nemoto** received his B.E., M.E., and Ph.D. degrees from Tohoku University in 1968, 1970, and 1973, respectively. He is a full professor with the Graduate School of Information Sciences, and served as director of the Information Synergy Center, Tohoku University. He has been engaged in research work on microwave networks, communication systems, computer network systems, image processing, and handwritten character recognition. He is a co-recipient of the 1982 Microwave Prize from the IEEE Microwave Theory and Techniques Society. He is a member of IEICE, and a fellow of the Information Processing Society of Japan.

**Abbas Jamalipour** received the Ph.D. degree in electrical engineering from Nagoya University, Nagoya, Japan. He is a professor at the School of Electrical and Information Engineering, University of Sydney, Australia, where he is responsible for teaching and research in wireless data communication networks, wireless IP networks, network security, and satellite systems. He is the author for the first technical book on networking aspects of wireless IP, *The Wireless Mobile Internet - Architectures, Protocols and Services* (New York: Wiley, 2003). In addition, he has authored another book on satellite communication networks, *Low Earth Orbital Satellites for Personal Communication Networks* (Norwood, MA: Artech House, 1998) and coauthored four other technical books in wireless telecommunications. He has authored over 180 papers in major journals and international conferences, and given short courses and tutorials in major international conferences. He is currently the editor-in-chief of the IEEE Wireless Communications, a technical editor of the IEEE Communications, and the International Journal of Communication Systems, and several other journals. Professor Jamalipour is a voting member of IEEE GITC and has been a vice chair of IEEE WCNC 2003–2006, chair of IEEE GLOBECOM 2005 (Wireless Communications), and a symposium co-chair at IEEE ICC 2005–2007, and IEEE GLOBECOM 2006. He is a fellow member of IEEE; a fellow member of IEAust; past chair of IEEE Communications Society Satellite and Space Communications Technical Committee; chair of Asia-Pacific Board, Chapters Coordinating Committee; and vice chair of Communications Switching and Routing Technical Committee. He is a distinguished lecturer of the IEEE Communications Society.

**Nei Kato** received his M.S. and Ph.D. degrees from Tohoku University, Japan in 1988 and 1991, respectively. He has been working with Tohoku University since then and is currently a full professor at the Graduate School of Information Sciences. He has been engaged in research on computer networking, wireless mobile communications, image processing, and neural networks. He has published over 100 papers in journals and peer-reviewed conference proceedings. Nei Kato has served as TPC member on a large number of IEEE international conferences including ICC, GLOBECOM, WCNC, and HPSR. From 2006, he is a technical editor of IEEE Wireless Communications. He is the recipient of the 2005 Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee. Nei Kato is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and a senior member of IEEE.