On the Performance of Congestion Control Protocols in Lossy Wireless Networks

(Invited Paper)

Hiroki Nishiyama[1,*], Tarik Taleb[1,†], Nirwan Ansari[2], Yoshiaki Nemoto[1,‡], and Nei Kato[1,*]

[1]Graduate School of Information Sciences
Tohoku University, Japan
*{bigtree,kato}@it.ecei.tohoku.ac.jp
†talebtarik@ieee.org, ‡nemoto@nemoto.ecei.tohoku.ac.jp

[2]Advanced Networking Laboratory
Department of Electrical and Computer Engineering
New Jersey Institute of Technology, USA
Nirwan.Ansari@njit.edu

***Abstract*— To cope with link errors in wireless networks, end-to-end solutions (e.g., TCP Westwood and TCP Veno) as well as router feedback-based solutions have been proposed. In this paper, we focus on the merits behind the use of router feedback-based approaches to tackle congestion and link errors in wireless environments. Firstly, some of the end-to-end improvements to TCP are presented and their strengths and drawbacks are discussed. Secondly, router feedback-based methods are introduced. It is shown that proactive window adjustment methods based on the fed back information are able to achieve high throughput even in high link error environments. Finally, we discuss the importance of router feedback-based mechanisms and discuss the required features for a scalable and feasible feedback mechanism.**

I. Introduction

In recent years, new wireless access technologies have been rapidly developed for easy connection to the Internet. Wireless LAN, Worldwide Interoperability for Microwave Access (WiMAX), and 3^{rd} Generation cellular systems (3G) are few examples. However, the current TCP cannot exploit the full advantage of these high speed links. TCP was initially designed for wired networks. It does not take into account the packet losses caused by wireless link errors. All packet losses are considered due to congestion. In the event of a packet loss, the sending rate is decreased. In wireless networks, a packet loss does not necessarily indicate network congestion as a number of packets can be dropped due to interference, multipath fading, and so forth. In such environments, the sender unnecessarily drops its transmission rate upon a loss event due to error, resulting in a waste of bandwidth and ultimately poor link utilization.

To improve the performance of TCP, some end-to-end approaches have been proposed, namely TCP Westwood [1], TCP Vegas [2], TCP Veno [3], and Jitter-based TCP (JTCP) [4]. In these schemes, additional functions to estimate the congestion state or the cause of a packet loss are introduced to achieve higher throughput even in high bit error environments. However, the performance of these methods largely depends on the estimation accuracy which depends, in turn, on network conditions (e.g., heavy network congestion, random packet losses, etc.).

Several TCP enhancements use router feedback to obtain more accurate information on network conditions. For example, TCP New Jersey [5], [6] is able to differentiate non-congestive loss from congestive loss according to the feedback notified by Explicit Congestion Notification (ECN) [7]. Explicit Loss Notification (ELN) [8] is another example. In ELN capable networks, a packet loss due to a wireless link error is notified to the source via Acknowledge (ACK) packet.

To further improve the TCP performance, several rate feedback approaches have been developed. Enhanced TCP (ETCP) [9] and Terrestrial-Recursive, Explicit, and Fair Window Adjustment Plus (T-REFWA+) [10] are some examples. In these schemes, an appropriate transmission rate (window size) of each flow is calculated at the router side and is fed back to each source. It becomes accordingly possible to achieve high throughput even in error prone environments by adjusting congestion window based on the feedback from the routers.

On the other hand, non-TCP transport layer protocols using router feedback have been proposed, e.g., eXplicit Control Protocol (XCP) [11]. Most of these protocols are different from TCP as they adopt different rate control policies and assume different network/packet types.

In this paper, we focus on router feedback-based proactive window control approaches that improve the performance of transport protocols in wireless networks. The advantages and the shortcomings of these schemes are discussed. The rest of the paper is organized as follows. We describe the existing congestion control protocols in Section II. They are categorized into three groups, namely, end-to-end, explicit notification, and rate feedback approaches. In Section III, the benefits of router feedback-based approaches are demonstrated via simulation. Features of efficient router feedback mechanism are discussed in Section IV. Section V concludes the paper.

II. Congestion Control Protocols

A. End-to-End Approaches

1) TCP Westwood: TCP Westwood is one of the most notable variants which improve TCP performance, particularly in wireless environments with high bit error rates. TCP Westwood adjusts the congestion window based on the available bandwidth estimated at the sender side. The estimated available bandwidth, $\hat{b}_k$, is calculated using the following equation when a sender receives the k^{th} ACK:

$$\hat{b}_k = \alpha_k \cdot \hat{b}_{k-1} + (1 - \alpha_k) \cdot \frac{b_k + b_{k-1}}{2} \qquad (1)$$

where b_k is the instantaneous available bandwidth. It is calculated as the amount of data acknowledged by the k^{th} ACK divided by Δ_k which represents the interarrival time between the k^{th} and $(k-1)^{th}$ ACKs. α_k is defined as $\{(2\tau - \Delta_k) / (2\tau + \Delta_k)\}$ where $(1/\tau)$ is the cutoff frequency of the discrete-time low-pass filter. The estimated available bandwidth $\hat{b}$ is used to set the slow start threshold, $ssthresh$, and the congestion window, $cwnd$, upon the detection of packet losses. When three duplicate ACKs are received, $ssthresh$ is set to w and $cwnd$ is updated to $\min(cwnd, w)$. w indicates the most appropriate window size and is defined as the product of $\hat{b}$ and the smallest RTT value observed over the duration of the connection. After a timeout expires, $ssthresh$ is adjusted as in the case of duplicate ACKs, and $cwnd$ is reduced to one packet.

Congestion window control relying on the estimated available bandwidth makes the performance of TCP Westwood less sensitive to random packet losses. However, its performance is heavily degraded in case of inaccurate bandwidth estimation.

On the other hand, it has been widely reported that the ACK compression leads to the overestimation of available bandwidth. To cope with this issue, TCP Westwood+ [12] has been proposed. Instead of computing the available bandwidth every time an ACK arrives, TCP Westwood+ computes the available bandwidth every RTT. It can thus mitigate the effect of ACK compression. However, errors in the bandwidth estimation remain a significant factor that affects the performance.

2) TCP Vegas: TCP Vegas handles its congestion window size according to changes in RTT. The sender computes *Diff* defined by Eq. (2) based on two RTT values: firstly the minimum RTT, rtt_{min}, which has been experienced since the beginning of the connection, and secondly the latest value of measured RTT, rtt_{cur}.

$$Diff = \frac{cwnd}{rtt_{min}} - \frac{cwnd}{rtt_{cur}} \qquad (2)$$

where $cwnd$ indicates the size of the congestion window at that time. The amount of backlogged packets in the buffer of a bottleneck link is represented as $(Diff \cdot rtt_{min})$.

TCP Vegas also has two thresholds, α and $\beta(>\alpha)$, which correspond to too little and too much backlogged data, respectively. When $(Diff < \alpha)$, TCP Vegas considers the network to be underutilized and increases its congestion window during the next RTT. On the other hand, the congestion window is linearly decreased (to avoid a possible congestion) when $(Diff > \beta)$. In the other case, $(\alpha < Diff < \beta)$, the congestion window retains the same value. By doing so, TCP Vegas tries to stabilize its congestion window. Also, the performance over lossy wireless networks can be improved because random packet drops do not directly affect the window control operation. However, when a bottleneck link is on the reverse path, TCP Vegas is inferior to standard TCP because its window control mechanism matches its transmission rate to the bottleneck capacity [13].

3) TCP Veno: Similar to TCP Vegas, TCP Veno adopts Eq. (2) to estimate the amount of backlogged packets and uses it as a congestion indicator. In other words, the congestion window control algorithm of standard TCP is modified by using the congestion indicator in TCP Veno. In the fast recovery phase, if the number of backlog packets, N, is less than a threshold θ, TCP Veno assumes that the loss is due to bit errors in the wireless link and sets the new congestion window to 80% (not half) of the current congestion window size. Therefore, the source can quickly recover its transmission rate after the retransmission. In the congestion avoidance phase, if N exceeds θ, TCP Veno deems the available bandwidth is fully utilized and decreases the growing speed of the congestion window by half to expand the duration of the congestion avoidance phase. However, the performance improvement by TCP Veno diminishes as the random packet loss rate increases because the reduction of congestion window frequently occurs along with high loss rates.

4) JTCP: To distinguish the causes of packet losses, JTCP uses the average jitter ratio, Jr, which is defined as the ratio of the sum of interarrival jitters during one RTT to RTT. If Jr is larger than $(k/cwnd)$, a loss is assumed to be due to network congestion. In contrast, a value of Jr smaller than the threshold indicates that the packet loss is caused by link errors. $cwnd$ indicates the current window size and if k is equal to 1, then just use $(1/cwnd)$.

Upon receiving three duplicate ACKs, JTCP invokes an immediate recovery instead of the fast recovery if the loss is regarded as a non-congestive loss. In the immediate recovery phase, both the slow start threshold and the new congestion window are set to $(d \cdot cwnd)$. The decrease factor d is set to one in [4]. After the expiration of the retransmit timer, JTCP halves the congestion window and enters the congestion avoidance phase instead of entering the slow start phase if Jr is below the threshold. JTCP can achieve good performance in wireless environments by using the jitter-based congestion detection mechanism. JTCP does not provision a mechanism to estimate the ideal window size.

B. Explicit Notification Approaches

1) ELN: In ELN schemes such as in [8] and [14], the receiver or the intermediate node informs the sender of the reception of a corrupted packet due to wireless link errors via an additional ELN option in ACK packets. Using this ELN information, the sender is able to distinguish the cause of the packet loss; network congestion or wireless link errors. Upon receiving three duplicate ACKs, the sender retransmits the lost packet without any window control if the loss is a non-congestive loss. Otherwise, the sending rate is halved as in standard TCP. By doing so, the sender can prevent the performance degradation caused by wireless link errors.

2) TCP New Jersey: TCP New Jersey has two key components, namely, the Congestion Warning (CW) and the Available Bandwidth Estimator (ABE). The TCP New Jersey source differentiates wireless losses from congestive losses based on the feedback generated by CW at the router, and accordingly adjusts its sending rate to the bandwidth estimated by ABE. A router marks all packets when the average queue length exceeds a predetermined threshold. For packet marking, the original ECN is employed to convey the congestion warning information. Upon receiving an ACK (including a duplicate ACK) packet with the CW mark, TCP New Jersey invokes the rate control procedure to adjust the window size. On the other hand, if the received duplicate ACKs are not marked with CW, the sender retransmits the lost packet without any rate control. In the rate control procedure, the slow start threshold is updated to an appropriate congestion window size and then the congestion window is set to the new slow start threshold if the current window size is larger than it. The appropriate congestion window size is defined as the product of RTT and the estimated bandwidth by ABE. The advantage of TCP New Jersey consists in the fact that senders can know the precursor of congestion using CW and accordingly prevent packet losses. A TCP New Jersey sender also has the ability to distinguish the cause of packet losses by employing CW. So, TCP New Jersey can outperform TCP Westwood which uses an end-to-end bandwidth estimator similar to ABE.

The ABE algorithm computes the available bandwidth, R_n, upon the reception of the n^{th} ACK as follows:

$$R_n = \frac{RTT \cdot R_{n-1} + L_n}{(t_n - t_{n-1}) + RTT} \tag{3}$$

where L_n denotes the amount of data acknowledged by the ACK. Instead of using the received time of the n^{th} ACK at the sender side as t_n, TCP New Jersey uses the arrival time of the n^{th} data packet at the receiver side. By applying the timestamp option [15], t_n is available at the sender. In general, the performance of the end-to-end estimation mechanisms is sensitive to corruptions on backward paths including the compression, delay, or loss of ACKs. However, by using the arrival time of data packets, these impacts may be removed and the estimation can be made more accurately. Other viable estimators can readily be adopted.

C. Rate Feedback Approaches

1) ETCP: ETCP uses the feedback generated by Explicit Window Adaptation (EWA) [16] (or other similar schemes including Fuzzy EWA [17]) to calculate the new congestion window size. Here, we first outline the EWA scheme and then describe the ETCP approach.

In EWA, a router informs the sources of traversing TCP connections about the appropriate window size via the receiver's advertised window (RWND) field in the TCP header of an ACK packet. By doing so, the maximum value of the congestion window is bounded by the feedback value according to standard TCP congestion control mechanism. The feedback value at time t, $F(t)$, is periodically calculated based on the buffer occupancy, $Q(t)$, as follows:

$$F(t) = \alpha \cdot \log_2 (B - Q(t)) \tag{4}$$

where B denotes the total buffer size and α is a variable scaling factor. α is also periodically updated based on the average queue length according to the policy of Additive-Increase and Multiple-Decrease (AIMD). The AIMD algorithm allows the system to reach the steady state where the buffer occupancy is within the desired range. EWA equally treats all TCP connections and leads to an efficient utilization of network resources; both buffer and link capacities. However, EWA results in unfair bandwidth allocation if connections have a high variance in their RTT distribution because all connections receive the same feedback regardless of their differences in RTTs.

Instead of slow start and congestion avoidance, ETCP updates its congestion window according to the function of the current congestion window size $cwnd$ and the feedback value F. The following three functions, f_1 to f_3, are used in [9].

1) $f_1 = F$
2) $f_2 = \min\left(cwnd \cdot 2^{1/cwnd},\ F\right)$
3) $f_3 = \min\left\{cwnd \cdot (F/cwnd)^{1/cwnd},\ F\right\}$

F is directly set to the new congestion window with f_1. In the case of f_2, the congestion window converges to its doubled value after receiving as many ACKs as $cwnd$. Thus, the congestion window continues to grow exponentially until it reaches the fed back value. On the other hand, the congestion window is rapidly increased and converges to the fed back value when f_3 is used. In all these ETCP variants, a pacing mechanism is used to avoid bursty transmissions caused by rapid increase in the congestion window. After a packet loss, ETCP carries out the fast retransmit and fast recovery mechanisms as standard TCP does. However, ETCP is able to quickly bring up the sending rate to the highest level by its aggressive window control algorithm relying on the feedback generated by EWA. This is why ETCP can achieve good performance gain, especially in wireless environments with high bit error rates.

2) T-REFWA+: T-REFWA+ is a sender side modification based on the feedback generated by the T-REFWA [18], [19] scheme. While the feedback is notified to sources in the same way as in EWA (i.e., via the RWND field), the control mechanisms in T-REFWA and T-REFWA+ are quite different from those in EWA and ETCP. Here, a brief overview of the original T-REFWA scheme is presented followed by a description of the major operations of T-REFWA+.

A T-REFWA-capable router periodically computes the effective Bandwidth-Delay Product (BDP) of the network using spare bandwidth and free buffer size. It then divides among all existing active flows in proportion to their RTTs. By matching the aggregate traffic rate of all active flows to the effective BDP, T-REFWA is able to make a highly efficient utilization of network resources while avoiding congestive packet losses. Moreover, the unfairness issue among competing flows with different RTTs is alleviated by taking RTT values into account in the bandwidth allocation process. The RTT value of each flow is sent from each source to routers via the Type of Service (ToS) field of the TCP packet header [18]. In T-REFWA, the computed feedback changes upon only changes in network conditions (e.g., flow count). Especially, the feedback value gets largely degraded due to network congestion when a new flow comes in. Using this feature, T-REFWA+ predicts the cause of packet losses by tracking changes in feedbacks.

Upon reception of an ACK packet, a T-REFWA+ sender compares the feedback value W_c signaled by the currently received ACK and the previous feedback value W_p recorded by the sender. The sender deems that the network is congested when

$$W_c < (r \cdot W_p) \tag{5}$$

where r is a constant parameter from within [0,1]. It reflects how much decrease in feedback values can be seen as an indicator of network congestion. If the sender gets no such sign for $(k \cdot RTT)$ time period before and after a reception of the third duplicate ACK, the packet loss event is treated as non-congestive loss and the congestion window is set to the most recently received feedback value. Otherwise, the sender behaves as standard TCP. k is a constant parameter handling the relation between a packet loss and the decrease in the feedback value. In case of the expiration of the retransmit timer, the sender checks whether any ACK packet has recently been received as well as if there is any sign of congestion. If at least one ACK packet reaches the sender and network congestion is not detected, the congestion window is set to the latest feedback value. Moreover, in order to improve TCP performance in heavy loss environments, T-REFWA+ freezes the RTO backoff mechanism. As described above, T-REFWA+ adjusts the sending rate more aggressively based on the appropriate rate signaled by the T-REFWA mechanism, and so it can improve the TCP performance in lossy wireless networks.

3) XCP: XCP is one of the most notable non-TCP congestion control protocols. In XCP, Congestion Header (CH) is introduced to exchange valuable information for congestion control between a XCP sender and XCP routers along the communication path. CH consists of three fields: current congestion window size, estimated RTT value, and feedback value. The router computes the feedback value of each flow based on the congestion window size and the RTT value of all traversing flows which are available from CH of packets. The calculated value is fed back to each sender via the feedback value field of the packets. Upon receiving a new acknowledgement packet, the sender increases or decreases its congestion window by the received feedback value. By doing so, XCP is able to quickly increase its sending rate even in networks with high bandwidth-delay product.

The XCP router performs two functions, namely, Efficiency Controller (EC) and Fairness Controller (FC). EC computes the aggregate feedback, F, based on the spare bandwidth, Bw and persistent queue size, Q, as follows:

$$F = \alpha \cdot d \cdot Bw - \beta \cdot Q \tag{6}$$

where α and β are constants, and d is the average RTT value of all active flows. After the computation of F, FC allocates it

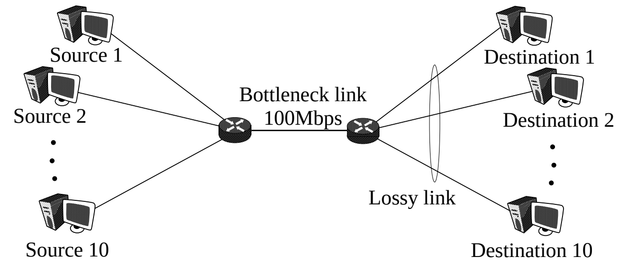

Fig. 1. Simulation network.

to each flow according to the policy of AIMD. In other words, the throughput is increased by the same value in each flow if F is positive. On the other hand, the throughput of each of the flows is decreased proportionally to the current throughput if F is negative. It should be noted that XCP can prevent the unfair bandwidth allocation which is substantial in TCP. This is because FC is designed by taking into account that the throughput depends not only on the congestion window size but also on the RTT value. As mentioned above, XCP maximizes the link utilization and minimizes packet losses by EC. It achieves fair bandwidth allocation by FC.

When a packet drop is detected, the XCP source transits to standard TCP behavior. Since congestive packet drops are almost zero in pure XCP networks as shown in [11], XCP sources consider that a loss implies the presence of non-XCP router in the path and behaves as a standard TCP. The performance of XCP is also degraded by this conservative response to packet losses in high bit error environments. However, under the assumption that all routers are XCP-capable, the performance can be improved by means of a simple modification in which the congestion window is not decreased upon the receipt of the third duplicate ACK or timeout occurrence. The concept of this modification is similar to the idea proposed in [20] and this XCP variant is referred to as "Enhanced XCP" in this paper.

III. Performance Evaluation

To evaluate the performance of congestion control protocols, we perform computer simulations. We consider a simple network topology as shown in Fig. 1. Ten connections having equal RTT values share the single bottleneck link and packet drops occur due to buffer overflow at the bottleneck queue. Also, some packets are dropped in the lossy links, corresponding to wireless losses due to random link errors. Six protocols are used: TCP NewReno, TCP Westwood+, TCP New Jersey, T-REFWA+[1], XCP, and Enhanced XCP[2]. Fig. 2 plots the average individual goodputs of each protocol for different packet error rates, ranging from 10^{-5} to 10^{-1}.

The three protocols that employ router feedback mechanisms, TCP New Jersey, T-REFWA+, and Enhanced XCP, show dramatic performance improvements in high bit error rate environments. They can sustain high throughputs because

[1]The two parameters, r and k, are set to 0.9 and 0.5, respectively.

[2]In case of XCP and Enhanced XCP, a pure XCP network is considered.

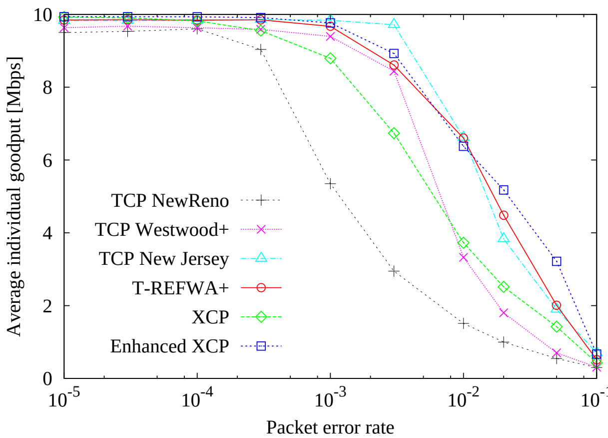

Fig. 2. Performance of protocols over high bit error rates.

the congestion window is controlled according to the feedback information from routers. In contrast, the performance of Westwood+, one of the end-to-end control protocols, is degraded due to errors in bandwidth estimation when the packet error rate exceeds 10^{-2}. Comparing between the graphs of XCP and the Enhanced XCP, it becomes clear that the use of router feedback in the congestion control operation yields significant improvements in the performance. The results show that the router feedback approaches have an important potential to drastically improve the congestion control performance even in wireless networks with high bit error rates.

IV. Future Direction

Table I shows the properties of the introduced congestion control protocols. End-to-end solutions have two crucial problems regardless of the types of their used techniques, bandwidth estimation or loss differentiation. The first shortcoming is that the performance heavily depends on the accuracy of the estimation. At worst, they could be inferior to standard TCP due to incorrect estimate. Secondly, there is no guarantee of fairly sharing the network resources among competing flows. This is because the rate estimated at an end-host may not be a fair rate. In explicit notification approaches, accurate information about the presence of wireless losses or network congestion is conveyed to sources, thus facilitating a suitable rate control. However, they have the same problems as end-to-end solutions because they cannot get any useful knowledge about the ideal rate. On the other hand, the issue on prediction accuracy can be avoided in rate feedback approaches because an appropriate window size is signaled to each source. Also, the fairness issue can be solved in T-REFWA+ and XCP even if competing flows have a high variance in their RTT distribution. However, in these rate feedback schemes, the mechanisms for computing and conveying the optimal window size increase the system complexity at the router side.

While the router feedback approach is a viable solution to improve the performance of congestion control protocols, the following issues should be addressed.

1) *Which information should be fed back?* Routers can get useful information by using measurements such as spare bandwidth, buffer occupancy, queuing delay and the number of traversing flows. Also, some secondary information, e.g., the ideal rate or window size, can be obtained from the primary ones.
2) *How is the information fed back to the source?* This can be facilitated by employing ECN-like mechanisms, via ACK packet headers, or other methods. Introducing a new header field is not preferred. Owing to the limitation of the header field, any encoding algorithm may be employed.
3) *How is the information used for rate control at the source?* According to [21], only a simple modification is effective enough to improve the performance of TCP even with random losses. Nevertheless, the rate control algorithm should be designed to take the full advantage of the information in order to greatly enhance the performance of the congestion control protocol.

Among these issues, the key component is how to transmit the feedback from the router to the source. If any packet marking scheme relying on the ECN-like framework is adopted, the amount of transmittable information is limited (a few bits). In such a case, the rate control mechanism at the sender side has to be robust to handle the feedback error because the fed back information would have already been deteriorated due to compression. It is not easy to design such a robust and efficient window control algorithm. On the other hand, the ideal window size is explicitly signaled to the source with no error if the RWND field is used as a carrier. However, the router has to look into the TCP header (not IP header), and hence the processing load increases. Also, RWND-based methods are not able to transmit any information except the window size. The straightforward solution to convey any information without an error is the introduction of an additional header like in XCP. This, however, also introduces more processing load.

So far, we have only considered congestion control protocols. The information observed at intermediate nodes may be highly useful for other protocols including application layer protocols such as Real-time Transport Protocol (RTP). Considering the wide use of feedback techniques, it is preferred that the router feedback is performed within the IP layer. Indeed, when IP-security (IPsec) is in use, such a feedback mechanism will not violate the IPsec semantics, unlike many RWND-based methods. From this point of view, Deterministic Packet Marking (DPM) approach is one of the most dominant feedback generators. In some recently proposed DPM schemes [22], [23], the IP identification field, which is designed for IP fragmentation, is used as a marking area. Since the size of the field is 16 bits, which is equal to that of RWND, it seems to be enough to contain the necessary information. However, the development of marking schemes compatible with IP fragmentation is a remaining issue, though it has been reported in [24] that over 99% of the total IP traffic is non-fragmented. To engineer the router feedback-based rate control, it is indeed important to develop a feasible

TABLE I

COMPARISON AMONG MAJOR WIRELESS SOLUTIONS FOR CONGESTION CONTROL.

	End-to-End Approach				Explicit Notification Approach		Rate Feedback Approach		
	TCP Westwood	TCP Vegas	TCP Veno	JTCP	ELN	TCP New Jersey	ETCP	T-REFWA+	XCP
Bandwidth estimate	Sender	None	None	None	None	Sender	Router	Router	Router
Loss differentiation	No	No	Yes	Yes	Yes	Yes	No	Yes	No
Modification	Sender	Sender	Sender	Sender	Sender, Router or Receiver	Sender, Router	Sender, Router	Sender, Router	Sender, Router, Receiver
Required field	–	–	–	–	ELN option	ECN	RWND	RWND, ToS	CH
Fairness control	No	No	No	No	No	No	No	Yes	Yes
Complexity	Low	Low	Low	Low	Medium	Medium	High	High	High
Expandability[3]	Low	Low	Low	Low	Low	Medium	High	High	High

and effective method for transferring the feedback.

V. CONCLUSION

In this paper, we discussed the issues of TCP in wireless environments with high bit error rates. Some effective solutions for congestion control in wireless environments were presented and categorized into three groups. The performance improvements by end-to-end and explicit notification approaches are limited due to the lack of detailed knowledge regarding network conditions. On the other hand, rate feedback approaches are able to drastically improve the network performance even in lossy networks because useful information on the appropriate sending rate is given by the feedback from the router. However, the complexity of the feedback notification techniques prevents their wide use. Although we suggested DPM as a candidate for the feedback generator, it is not a complete solution. Further developments are required for a feasible and smart feedback transmission scheme.

ACKNOWLEDGMENT

This research was partially supported by Grant-in-Aid for JSPS[4] Fellows.

REFERENCES

[1] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," *Wireless Networks*, vol. 8, no. 5, pp. 467–479, Sep. 2002.

[2] 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.

[3] C. P. Fu and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 216–228, Feb. 2003.

[4] E. H.-K. Wu and M.-Z. Chen, "JTCP: Jitter-Based TCP for Heterogeneous Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 4, pp. 757–766, May 2004.

[5] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for Wireless IP Communications," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 4, pp. 747–756, May 2004.

[6] K. Xu, Y. Tian, and N. Ansari, "Improving TCP performance in integrated wireless communications networks," *Computer Networks*, vol. 47, no. 2, pp. 219–237, Feb. 2005.

[7] K. K. Ramakrishnan, S. Floyd, and D. L. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," IETF, RFC 3168, Sep. 2001.

[8] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, Dec. 1997.

[9] M. Savoric, "Improving Congestion Control in IP-based Networks Using Feedback from Routers," Telecommunication Networks Group, TKN Technical Report TKN-04-008, Jul. 2004.

[10] H. Nishiyama, T. Taleb, Y. Nemoto, A. Jamalipour, and N. Kato, "Enhancements of T-REFWA to Mitigate Link Error-Related Degradations in Hybrid Wired/Wireless Networks," *Journal of Communications and Networks*, vol. 8, no. 4, pp. 391–400, Dec. 2006.

[11] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 89–102, Oct. 2002.

[12] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, and G. Piscitelli, "Performance evaluation of Westwood+ TCP congestion control," *Performance Evaluation*, vol. 55, no. 1/2, pp. 93–111, Jan. 2004.

[13] C. Fu, L. C. Chung, and S. C. Liew, "Performance Degradation of TCP Vegas in Asymmetric Networks And Its Remedies," in *Proc. of IEEE ICC*, Helsinki, Finland, Jun. 2001, pp. 3229–3236.

[14] J.-H. Yun and S.-W. Seo, "New Explicit Loss Notification for TCP in Wireless Networks," in *Proc. of IEEE Vehicular Technology Conference (VTC)*, Milan, Italy, May 2004, pp. 2271–2275.

[15] V. Jacobson, B. Braden, and D. Borman, "TCP Extensions for High Performance," IETF, RFC 1323, May 1992.

[16] 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.

[17] M. Savoric, "Fuzzy Explicit Window Adaptation: Using Router Feedback to Improve TCP Performance," Telecommunication Networks Group, TKN Technical Report TKN-04-009, Jul. 2004.

[18] 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, pp. 1921–1926.

[19] T. Taleb, N. Kato, and Y. Nemoto, "REFWA: An Efficient and Fair Congestion Control Scheme for LEO Satellite Networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 1031–1044, Oct. 2006.

[20] K. Zhou, K. L. Yeung, and V. O. K. Li, "P-XCP: A Transport Layer Protocol for Satellite IP Networks," in *Proc. of IEEE GLOBECOM*, Dallas, TX, Nov./Dec. 2004, pp. 2707–2711.

[21] A. Karnik and A. Kumar, "Performance of TCP Congestion Control With Explicit Rate Feedback," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 108–120, Feb. 2005.

[22] A. Belenky and N. Ansari, "On deterministic packet marking," *Computer Networks*, vol. 51, no. 10, pp. 2677–2700, Jul. 2007.

[23] R. W. Thommes and M. J. Coates, "Deterministic Packet Marking for Congestion Price Estimation," in *Proc. of IEEE INFOCOM*, Hong Kong, China, Mar. 2004, pp. 76–85.

[24] C. Shannon, D. Moore, and K. C. Claffy, "Beyond Folklore: Observations on Fragmented Traffic," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 709–720, Dec. 2002.

[3] Expandability for further enhancements

[4] The Japan Society for the Promotion of Science