TCP WestwoodVT: A Novel Technique for Discriminating the Cause of Packet Loss in Wireless Networks[⋆]

Jahwan Koo, Sung-Gon Mun, and Hyunseung Choo

School of Information and Communication Engineering, Sungkyunkwan University
Chunchun-dong 300, Jangan-gu, Suwon 440-746, South Korea
`jhkoo@songgang.skku.ac.kr, msgon,choo@ece.skku.ac.kr`

Abstract. Conventional TCP in wireless environment cannot differentiate packet losses caused by network congestion from those caused by wireless link errors, thus, resulting in severe performance degradation. Accordingly, efficient operation of TCP in wireless networks is a critical issue in the context of differentiation between packet loss. Towards this issue, we proposes a novel technique, WestwoodVT (WestwoodNR based on TCP Vegas buffer Thresholds), which is a sender-based TCP congestion control mechanism for discriminating the cause of packet loss to enhance the performance of TCP in wireless environment. Simulation results show that, under various wireless link error rates, it achieves a maximum of 41% and 118% improvements in goodput over WestwoodNR and TCP Reno, respectively. WestwoodVT only requires changes at the send-side. Thus, it eliminates any changes to the intermediate routers unlike TCP New Jersey. Therefore, it is cost effective to implement in already deployed networks. Moreover, its fairness and friendliness are also satisfied.

Key words: Transmission control protocol, wireless networks, congestion control

1 Introduction

Conventional transmission control protocol (TCP) is a reliable transport protocol that is designed to perform well in wired networks. However, it may suffer from severe performance degradation in wireless networks due to atmospheric conditions, high transmission errors, temporal disconnections, and signal fading [1]. The main reason for the TCP performance degradation over wireless networks is that the conventional TCP (such as TCP Reno [2]) cannot distinguish between packet losses caused by transmission errors and those caused by network congestion, thus, blindly reacting to these losses by reducing its congestion window. Consequently, these inappropriate reductions of the congestion window lead to unnecessary throughput degradation for TCP applications [3].

[⋆] Corresponding author: H. Choo.

To overcome such limitation and degradation, several schemes have been proposed and are classified in [4] and [5]. Generally, current research work has been classified into two directions: (1) end-to-end TCP modifications such as congestion control mechanism and (2) link layer approaches that include intermediate router mechanisms. However, End-to-end TCP modifications do not distinguish the cause of packet loss whether it is due to network congestion or wireless link error. Link layer approaches require more time and cost to be deployed in real wired and wireless networks. Therefore, without supporting any intermediate router mechanisms, the discrimination of the cause of packet loss using the modified end-to-end TCP mechanism is a critical issue in wireless environment.

In this paper, we propose a novel technique, WestwoodVT (WestwoodNR based on TCP Vegas buffer Thresholds), which is a sender-based TCP congestion control mechanism for discriminating the cause of packet loss to enhance the performance of TCP in wireless environment. WestwoodVT uses the flow control concept of TCP Vegas [6] in order to discriminate the cause of packet loss. If the cause is due to network congestion, WestwoodVT retransmits packets using the available bandwidth estimator of WestwoodNR [7] [8], which is a very efficient TCP scheme in wired and wireless networks. Meanwhile, if the cause is due to wireless link errors, WestwoodVT ignores the loss and retransmits the packet as if the packet loss has never occurred. The result from the simulation using *NS-2* demonstrates that, under various wired and wireless link error rates, WestwoodVT achieves a maximum of 41% and 118% improvements in goodput over WestwoodNR and TCP Reno, respectively. Furthermore, WestwoodVT and TCP New Jersey [9] show very similar performance. Since TCP New Jersey is supported by an intermediate router mechanism, WestwoodVT, which is required only a sender-side modification, outperforms WestwoodNR, TCP Reno, and TCP New Jersey in terms of efficiency, adaptation, deployment, simplicity, and cost in wired and wireless networks.

2 Related Works

2.1 TCP Reno

TCP Reno [2] is a well-known standard TCP scheme. It has four transmission phases: slow start, congestion avoidance, fast recovery, and fast retransmit. At the beginning of a TCP connection, the sender enters the slow start phase, in which the congestion window size (*cwnd*) is increased by one for every acknowledgment (ACK) received. When *cwnd* reaches the slow start threshold (*ssthresh*), the TCP sender enters the congestion avoidance phase, in which it increases its *cwnd* by the reciprocal of the current window size every time it receives an ACK. This increases the window by one in each round-trip time (RTT). When packet loss occurs at a congested link due to buffer overflow at the intermediate router, either the sender receives duplicate ACKs (DUPACKs), or the sender's retransmission timeout (RTO) timer expires. These events activate TCP's fast retransmit and recovery, by which the sender reduces the size of its *cwnd* to half and linearly increases *cwnd*, resulting in a lower transmission rate.

2.2 TCP Vegas

Congestion detection in TCP Reno assumes the loss of packets as a signal of congestion. Since there is no mechanism in TCP Reno to early detect the congestion before the loss occurs, it is reactive rather than proactive. On the other hand, TCP Vegas [6] measures and controls the amount of extra data that a connection carries, where extra data means data that would not have been sent if the bandwidth used by the connection exactly matches the available bandwidth of the link. The goal of TCP Vegas is to maintain the proper amount of extra data in the network. Obviously, if a connection has too much extra data, it will result in congestion. If it has too little extra data, it is not able to respond fast enough to transient increase in the available bandwidth. TCP Vegas estimates the maximum expected transmission rate and measures actual transmission rate when received every ACK. It sets a variable, *diff*, which is calculated by the difference in actual from expected transmission rate. It also defines two thresholds, α and β, roughly corresponding to having too little and too much extra data in networks. When $\mathit{diff} < \alpha$, TCP Vegas increases the congestion window linearly during the next RTT, and when $\mathit{diff} > \beta$, TCP Vegas decreases the congestion window linearly during the next RTT. TCP Vegas leaves the congestion window unchanged when $\alpha < \mathit{diff} < \beta$.

2.3 WestwoodNR

WestwoodNR [7] [8] is a sender-side modification based on TCP New Reno, which is intended to better handle large bandwidth delay product (BDP) with potential packet loss due to network congestion or wireless link errors, and with dynamic load. It is a rate based end-to-end proactive approach, in which the sender estimates the available network bandwidth dynamically, by measuring and averaging the rate of returning ACKs. It relies on returning the ACK stream, and this information helps it better set the congestion control parameters: *ssthresh* and *cwnd*. In WestwoodNR, the transmission rate using the available bandwidth estimator is evaluated and used by the sender to update *ssthresh* and *cwnd* upon loss indication for efficient throughput in wired and wireless networks. However, WestwoodNR does not discriminate the cause of packet loss. It will adjust retransmission rates constantly upon experiencing packet loss. Therefore, it decreases the throughput in high wireless link error rate. This is also a problem that the accuracy of the estimated available bandwidth depends on network conditions, which is changed by network traffic in links.

2.4 TCP New Jersey

TCP New Jersey [9], known as the best existing scheme in terms of throughput and goodput, aims to improve wireless TCP performance using the available bandwidth estimator (ABE) at the sender and the congestion warning (CW) at the intermediate router. ABE estimates the bandwidth available to the TCP connection and guides the sender to properly adjust its transmission rate. CW is a congestion notification implemented at the intermediate routers that helps the sender to effectively distinguish packet losses caused by network congestion from those caused by transmission errors. Based on the bandwidth estimation from ABE and the congestion indication implied by CW, TCP New Jersey calculates the size of the congestion window. After that, the actual size of transmission window is determined as the minimum of the receiver-advertised window size and the sender-calculated congestion window size. Consequently, the joint approach with the combination of ABE and CW allows to improve TCP performance in high bit-error-rate (BER) wireless networks. However, TCP New Jersey may experience the decreasing of throughput depending on background traffic pattern. It cannot increase the reduced sending rates effectively according to the cause of packet loss. Consequently, if packet loss consistently occurs due to the high wireless link error, TCP New Jersey will reduce its throughput. In addition, CW is based on intermediate router mechanism. It is not easy to deploy TCP New Jersey into real networks because it requires the addition and modification of both the sender-side and the intermediate router-side modules. Thus, it consists of implementation, deployment, and management complexity.

3 WestwoodVT

3.1 Motivation

WestwoodNR has more efficient ABE than other TCP schemes. However, it has no operation to discriminate the cause of packet loss. On the other hand, even if TCP New Jersey is known as the best existing scheme in terms of throughput and goodput, it requires the addition and modification of both the sender-side and the intermediate router-side modules. Thus, it consists of implementation, deployment, and management complexity.

A novel TCP scheme for discriminating the cause of packet loss while introducing minimum additional complexity is needed. Specifically, our goal is to develop a solution that, from an implementation complexity perspective, requires little more than a sender-side modification. As we shall see, the only addition we consider is the discriminating of the cause and the effective retransmission mechanism in a sender-side only. Concretely, the discrimination mechanism uses the buffer state of network nodes based on the flow control concept of TCP Vegas. According to the cause, the retransmission mechanism uses the ABE of WestwoodNR.

3.2 Discriminating the Cause of Packet Loss

WestwoodVT investigates the buffer state of network nodes between sender and receiver in order to discriminate the cause of packet loss. The discrimination inherits from the concept of TCP Vegas. WestwoodVT estimates the maximum *Expected* transmission rate and *Actual* transmission rate when received every ACK. The *Expected* transmission rate is given by:

$$Expected = \frac{WindowSize}{BaseRTT} \tag{1}$$

where *WindowSize* is the size of the current congestion window, equal to the number of bytes in transit, *BaseRTT* is the minimum value of all measured RTT. The *Actual* transmission rate is given by:

$$Actual = \frac{WindowSize}{RTT} \tag{2}$$

where *WindowSize* is equal to *WindowSize* of Eq. 1, *RTT* is the current calculated RTT. Additionally, WestwoodVT defines $\triangle$ is calculated by the difference of *Actual* from *Expected* transmission rate. The value of $\triangle$ presents the amount of current existing data in buffer of network nodes. Namely, $\triangle$ indicates the state of the current network. This is given by:

$$\triangle = (\frac{WindowSize}{BaseRTT} - \frac{WindowSize}{RTT}) \times BaseRTT \tag{3}$$

In addition, WestwoodVT defines buffer thresholds, α and β. These values indicate the lower and upper bound of network nodes' buffer. The discrimination of the cause of packet loss in WestwoodVT is operated in congestion avoidance phase. In this phase, the sender estimates $\triangle$ using Eq. 1, 2, and 3 when it receives 3-DUPACK. WestwoodVT then compares $\triangle$ to buffer thresholds α and β.

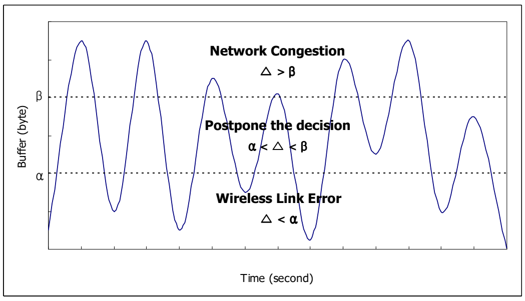


Fig. 1. Discriminating of the cause of packet loss in WestwoodVT.

Figure 1 presents main concept of the mechanism, which discriminates the cause of packet loss when the sender receives 3-DUPACK. If $\triangle$ is smaller than threshold α, WestwoodVT assumes that the current buffer state of network nodes is loose, and decides that the packet loss is due to wireless link error. If $\triangle$ is greater than threshold β, it assumes that the current buffer state of network nodes is tight. At this point, WestwoodVT considers the packet loss as that caused by network congestion. If $\triangle$ is greater than α and smaller than β, WestwoodVT does not decide the cause of the packet loss, maintains the current state, and retransmits lost packets. Therefore, WestwoodVT postpones the decision of cause until it receives 3-DUPACK for more accurate discrimination.

3.3 The Operation of WestwoodVT

WestwoodVT classifies the cause of packet loss using the buffer state of network nodes. When the cause is due to network congestion, WestwoodVT operates congestion control mechanism using the ABE of WestwoodNR, which is a very efficient TCP scheme in wired and wireless networks. However, when the cause is due to wireless link error, there is no requirement to use congestion control mechanism because there is no network congestion. In this case, WestwoodVT increases *cwnd* by one, and then retransmits lost packets as if transmitting a standard packet.

```
/* In Congestion Avoidance */
if  (3-DUPACK received)
    if ($\triangle > \beta$)
        Available bandwidth estimation of WestwoodNR;
        /* Due to congestion */
    else if ($\triangle < \alpha$)
        cwnd = cwnd + 1;
        /* Due to wireless link errors */
    else
        cwnd = cwnd;
        /* Postpone the decision */
    end if
end if
```

Fig. 2. Pseudo code of WestwoodVT.

The reason of increasing the congestion window is that the sender generally increases *cwnd* by one for every ACK received. At the packet loss is caused by wireless link error, when the sender receives third duplicated ACK, it does not increase the congestion window because of fast recovery mechanism; normally the sender increases the congestion window by one when it receives first or second duplicated ACK. This results in the loss of congestion window. In this situation, WestwoodVT modifies fast recovery mechanism using increasing the congestion window by one. Figure 2 shows the pseudo code of WestwoodVT.

4 Simulation Results

In this section, WestwoodVT is compared to WestwoodNR, TCP New Jersey, and TCP Reno using *NS-2* simulator in terms of goodput, fairness, and friendliness. These are metrics of TCP performance evaluation on wired and wireless networks. Table 1 presents parameters for simulations.

Table 1. Simulation parameters

Packet Size	Queue Size	Bandwidth		Delay	
726 B	20 MSS	wired	wireless	wired	wireless
		100 MB	2 MB	10 ~ 20 ms	1 ms

4.1 Goodput Performance

Goodput is the effective amount of data delivered through the network. It is a direct indicator of network performance. The simulation environment is depicted in Fig. 3.

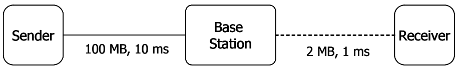


Fig. 3. Simulation topology for obtaining goodput performance.

A single TCP connection running a long-live FTP application delivers data from sender to receiver, under various wireless link errors: 0.1%, 1%, 2%, 3%, 4%, and 5%, respectively. In Fig. 3, all network nodes maintain the queue size of 20 MSS.

In TCP New Jersey, CW makes a decision the cause of packet loss, either network congestion or wireless link error. When amount of existing data over router's buffer is greater than 3/4 of amount of the buffer, if packet loss occurs, CW assumes that the cause of packet loss is due to network congestion. Otherwise, CW makes a decision that the cause is due to wireless link error.

Similarly, the buffer thresholds of WestwoodNR are configured to roughly three quarters of the queue size of network nodes. We setup the lower bound of WestwoodVT, α to 14. if $\triangle < 14$, the cause is due to wireless link error. The upper bound, β is fixed to 16, if $\triangle > 16$, the cause is due to network congestion. If $\triangle$ is equal to 15, WestwoodVT defers the decision of the cause. Finally, we run the simulation for TCP WestwoodVT, -WestwoodNR, -New Jersey, and -Reno, respectively. Figure 4 demonstrates the result of simulation, an average of 10 times, over 200 seconds.

As shown in the simulation results, WestwoodVT is superior over WestwoodNR and TCP Reno in all wireless link error rates. With wireless link error

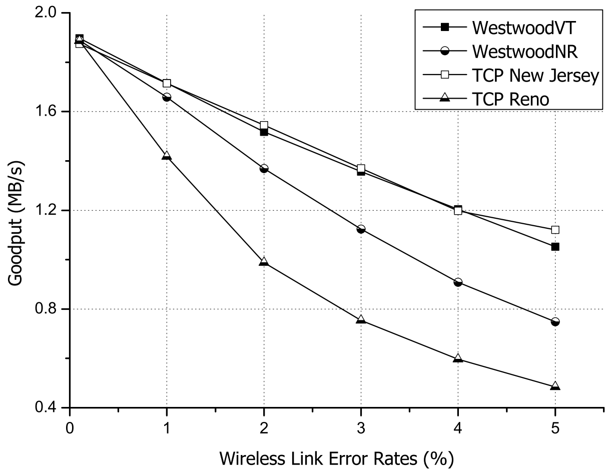


Fig. 4. Comparison of goodput of TCP WestwoodVT, -WestwoodNR, -New Jersey, and -Reno.

rate of 1%, WestwoodVT outperforms WestwoodNR by 3% and TCP Reno by 21%. WestwoodVT outperforms WestwoodNR by 11% and TCP Reno by 54% with wireless link error rate of 2%. Furthermore, WestwoodVT is almost identical to TCP New Jersey, until the wireless link error rate reaches 4%. WestwoodVT achieves a maximum of 41% and 118% improvements in goodput over WestwoodNR and TCP Reno, respectively.

The performance of TCP New Jersey is overwhelming other TCP schemes beyond wireless link error rate of 4%. However, because real wireless link error rate is 1% $\sim$ 2% and the performance of WestwoodVT is similar to TCP New Jersey. Therefore, WestwoodVT, a sender-side scheme, is generally better than TCP New Jersey and the other TCP schemes in terms of efficiency, deployment, adaptation, and cost.

4.2 Goodput Performance with Background Traffic

In order to simulate the performance of TCP WestwoodVT, -WestwoodNR, -New Jersey, and -Reno with background traffic of both forward and backward FTP traffic, we consider a more complex topology such as Fig. 5.

The queue size of all nodes are set to 20 MSS. A single TCP connection running a long-live FTP application delivers data from the sender to the receiver under various wireless link errors: 0.1%, 1%, 2%, 3%, 4%, and 5%, respectively. At the same time, cross-traffic flows are generated: forward traffic flows from

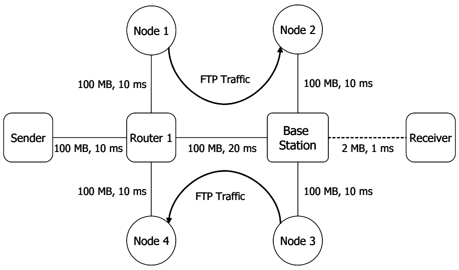


Fig. 5. Complex simulation topology with background traffic.

node1 to node2 via router1 and base station, and backward traffic flows from node3 to node4 via base station and router1, respectively.

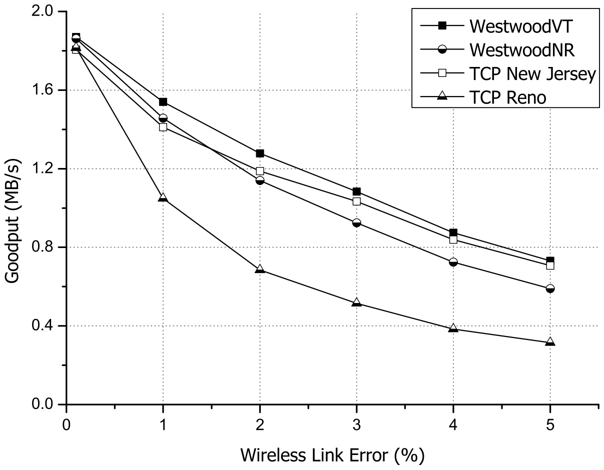


Fig. 6. Comparison of goodput of TCP WestwoodVT, -WestwoodNR, -New Jersey, and -Reno with background traffic.

Figure 6 depicts simulation results over 200 seconds, an average of 10 times. The buffer thresholds, α and β of WestwoodVT are configured to 14 and 16, respectively. These values are based on the queue size of network nodes. The performance of TCP Reno is worst. WestwoodVT achieves 6% $\sim$ 24% improvements in goodput over WestwoodNR, and it is overwhelming TCP New Jersey, under all wireless error rates. In this simulation, TCP New Jersey experiences the decreasing of throughput with background traffic. However, WestwoodVT keeps its performance constantly, regardless of wireless link error rates or background traffic.

4.3 Fairness of WestwoodVT

Another important issue of TCP is fairness. Multiple connection of the same TCP scheme must interoperate nicely and converge to their fair shares. We employ Jain's fairness index [10] function to justify the fairness of TCP schemes. The perfectly fair bandwidth allocation would result in a fairness index of 1.

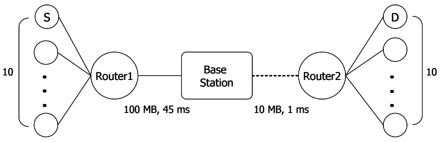
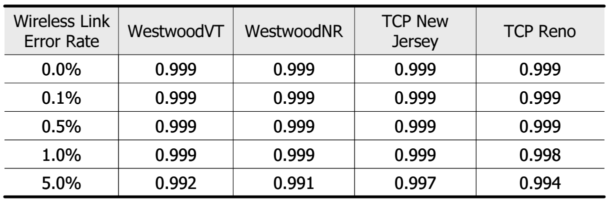


Fig. 7. Simulation topology for obtaining fairness index.

We setup the simulation environment as shown in Fig. 7. We run the simulation for different TCP schemes and compare their fairness index. The results of fairness are summarized in Table 2. All TCP schemes including WestwoodVT achieve a fairly satisfactory fairness index under various wireless link error rates.

Table 2. Fairness comparison



Wireless Link Error Rate	WestwoodVT	WestwoodNR	TCP New Jersey	TCP Reno
0.0%	0.999	0.999	0.999	0.999
0.1%	0.999	0.999	0.999	0.999
0.5%	0.999	0.999	0.999	0.999
1.0%	0.999	0.999	0.999	0.998
5.0%	0.992	0.991	0.997	0.994

4.4 Friendliness of WestwoodVT

Friendliness is another important property of TCP. A friendly TCP scheme should be able to coexist with other TCP schemes and not result in starvation. To verify the friendliness of WestwoodVT, we setup the simulation environment, as shown in Fig. 8, where WestwoodVT coexists with TCP Reno.

There are 10 pairs of TCP connections, of which m are TCP Reno connections and 10-m are WestwoodVT connections. All 10 connections are expected to share

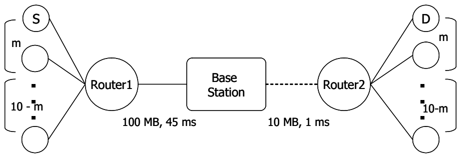


Fig. 8. Simulation topology for obtaining friendliness.

the 10MB link equally. We measure the throughput of each connection. The mean throughput of TCP Reno and WestwoodVT is calculated by summing up the throughput of the same TCP schemes and is divided by the number of connections, respectively. The results are presented in Fig. 9. WestwoodVT behaves more aggressively than nonwireless TCP such as TCP Reno. However, this behavior is anticipated since WestwoodVT is designed to perform better in lossy wireless environments.

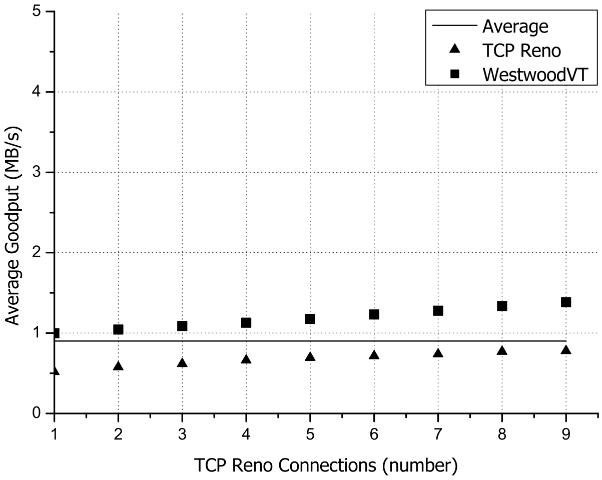


Fig. 9. Friendliness under 1% of wireless link error rate.

5 Conclusion

In this paper, we have proposed WestwoodVT which utilizes the sender-based transmission window control mechanism for discriminating the cause of packet loss. It checks the buffer state of network nodes between sender and receiver by using TCP Vegas operating mechanism and discriminates the cause of packet loss based on the buffer state. When the packet loss is due to network congestion, WestwoodVT uses the ABE of WestwoodNR for packet retransmissions. Otherwise, it retransmits packets without congestion control mechanism.

For performance evaluation, we setup WestwoodVT's buffer thresholds α and β to 14 and 16, respectively. The result from the simulation demonstrates that, under 1% of realistic wireless link error rate, WestwoodVT achieves a maximum of 3% and 21% improvements in goodput over WestwoodNR and TCP Reno, respectively. WestwoodVT and TCP New Jersey show very similar performance in wired and wireless networks. In our opinion, WestwoodVT is better than other TCP schemes such as WestwoodNR, TCP New Jersey, and TCP Reno in terms of efficiency, adaptation, deployment, simplicity, and cost in wired and wireless networks.

Acknowledgments.

This research was supported by the MIC, Korea, under the ITRC support program supervised by the IITA, IITA-2006-(C1090-0603-0046).

References

1. Xylomenos, G., Polyzos, G. C.: TCP Performance Issues over Wireless Links. IEEE Communications Magazine, Vol. 39. (2001) 52–58
2. Allman, M., Paxson, V., Stevens, W.: TCP Congestion Control. RFC 2581, (1981)
3. Lakshman, T. V., Madhow, U.: The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. IEEE/ACM Trans. Networking, Vol. 5, No. 3. (1997) 336–350
4. Balakrishnan, H., Padmanabhan, V. N., Seshan, S., Katz, R. H.: A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. IEEE/ACM Trans. Networking, Vol. 5, No. 6. (1997) 759–769
5. Tian, Y., Xu, K., Ansari, N.: TCP in Wireless Environments: Problems and Solutions. IEEE Radio Communications, Vol. 43. (2005) S27–S32
6. Brakmo, L. S., O'Malley, S. W., Peterson, L. L.: TCP Vegas: New Techniques for Congestion Detection and Avoidance. ACM/SIGCOMM Computer Communication Review, Vol. 24. (1994) 24–35
7. Casetti, C., Gerla, M., Mascolo, S., Sanadidi, M. Y., Wang, R.: TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. ACM/Mobicom. (2001) 287–297
8. Casetti, C., Gerla, M., Mascolo, S., Sansadidi, M. Y., Wang, R.: Westwood: End-to-End Congestion Control for Wired/Wireless Networks. Wireless Networks Journal, Vol. 8. (2002) 467–479
9. Xu, K., Tian, Y., Ansari, N.: Improving TCP Performance in Integrated Wireless Communications Networks. Computer Networks, Vol. 47. (2005) 219–237
10. Jain, R., Chiu, D., Hawe, W.: A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. Research Report TR-301, (1984)