Throughput and Fairness of Multiple TCP Connections in Wireless Networks

Hong Zhou, John Leis
Electrical, Electronic and Computer Engineering
University of Southern Queensland
Toowoomba, Australia
hzhou, leis@usq.edu.au

Doan Hoang, Phuong Nhan
Department of Computer Systems
University of Technology, Sydney, Australia
Sydney, Australia
dhoang, pnhan@it.uts.edu.au

Abstract –**TCP suffers from poor throughput performance in wireless networks. Furthermore, when multiple TCP connections compete at the base station, link errors and congestion lead to serious unfairness among the connections. Although the issue of TCP performance in wireless networks has attracted significant attention, most reports focus only on TCP throughput and assume that there is only a single connection in a congestion-free network. This paper studies the throughput and fairness of popular improvement mechanisms (the Snoop [8] and ELN [5]) and TCP variants with multiple TCP connections. Simulation results show that the improvement mechanisms under investigation are effective to improve TCP throughput in a wireless network. However, they cannot provide fairness among multiple TCP connections. From the studies presented, it is concluded that mechanisms to enhance TCP fairness are needed in wireless networks.**

I. INTRODUCTION

Transmission Control Protocol (TCP) has been deployed widely in the majority of commercial networks and services. It is thus essential for any wireless networking device used to access these networks and their services to communicate via TCP/IP. Developed some two decades ago, TCP is based on assumptions applicable to wired networks [1]. TCP assumes that packet loss occurs primarily as a result of congestion somewhere in the network. Obviously, this assumption is irrelevant in unreliable high error-rate wireless networks. TCP interpreting all packet losses as being due to congestions results in the unnecessary triggering of the congestion slow start procedure. TCP performance in wireless environments has been shown nowhere near as efficient as in wired networks [2].

Furthermore, when multiple TCP connections are concentrated on a base station, it is likely that one connection will take more bandwidth than others due to random link errors. To make the situation worse, the base station is liable to become a bottleneck. Packet losses due to congestion at the bottleneck lead to more serious unfairness among the connections.

Many schemes have been proposed to address this issue [3,4]. Examples include Explicit Loss Notification (ELN) [5], Explicit Congestion Notification (ECN), Indirect TCP [7] and Snoop [8]. In general, these proposals to overcome the problem are based on two fundamental ideas: (i) decoupling the congestion control from retransmission of lost packets and (ii) retransmission of the lost packets as early/closely to the wireless link as possible. ELN and ECN can provide the TCP sender with the reasons for packet loss. Indirect TCP decomposes the TCP connection into two sub-connections for the wired and wireless parts of the path. The Snoop protocol, a so-called link layer proposal, introduces a snoop agent at the base station. The agent monitors every packet that passes in both directions and maintains a cache of TCP segments that have not yet been acknowledged by the receiver. It detects packet loss by duplicate ACKs or a local Retransmission TimeOut (RTO), and retransmits the lost packet.

Over the years, a number of TCP variants to improve congestion control while maintaining good user throughput have been developed [10]. They are differentiated from each other mainly in the use of fast retransmission algorithm. Tahoe TCP includes Slow-Start, Congestion Avoidance, and Fast Retransmission algorithms [1]. Reno TCP modified Tahoe to include Fast Recovery [11]. New-Reno TCP includes a small change in the operation of retransmit timer at the sender. Vegas TCP extends Reno’s retransmission mechanisms. It reads and records the system clock each time a segment is sent [12,14]. SACK TCP is a conservative extension of Reno. The main difference is in the behavior when multiple packets are dropped from one window [12].

In this paper, we focus our interest on the throughput and fairness performance of two improvement mechanisms -- Explicit Loss Notification (ELN) [5] and the Snoop protocol [8] and four TCP variants (Tahoe, Reno, New-Reno and Vegas). We compare their performance in a wireless network with multiple TCP connections. Although it was compared comprehensively in wired networks [11], the performance of various TCP versions has not been studied in wireless networks, especially with multiple TCP connections competing for a fair share of channel bandwidth. Simulation results show that the Snoop protocol and ELN are effective mechanisms to improve the throughput performance of TCP versions in a wireless network. However, neither of the improvements can guarantee fairness among the connections. Mechanisms which can not only improve the throughput but also provide the fairness among multiple TCP connections are needed.

This paper is organized as follows. Section II describes the network scenario, simulation model and associated parameters. Section III examines the performance of four scenarios for each TCP version, namely, TCP, TCP with Snoop, TCP with ELN and TCP with both Snoop and ELN. Simulation results for each mechanism and TCP version are presented for comparison. Section IV summarizes the results and concludes the paper.

II. NETWORK MODEL

The network scenario is shown in Figure 1. We assume that there are a number of *n* fixed hosts (FHs) sending TCP data to *n* mobile hosts (MHs), respectively. That is, there are *N* communication pairs. Without loss of generality, suppose the TCP packets traverse a source edge router (SER), a core router (CR) and a base station (BS) from FHs to the MHs. The MHs are assumed to be located in a wireless LAN. In this paper, mobility issues are not considered. We focus on the impacts of link loss on the performance of TCP versions and TCP improvement mechanisms. Hence, a lossy link is used to represent wireless environment.

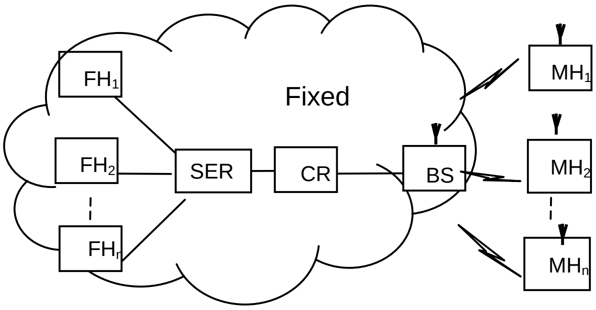


Figure 1. The network model

Network Simulator (ns2) [9] was used as simulation tool. Simulation models are set up for four versions of TCP and two TCP improvement mechanisms. Without loss of generality, there are five TCP connections sharing the wireless link. The propagation delay from the source to the destination is 26ms. The bandwidth at the lossy link is 2Mbps. The TCP packet size is 1000 bytes. We use a uniform error model to generate errors on the lossy links. The packet error rates changes from 0.001% to 10% (i.e. 0.0, 0.001%, 0.002%, 0.004%, …, 0.1%, 0.2%, …, 1.0%, …, 10.0%). The simulations run for 100 seconds with a TCP receiver window size of 10 packets.

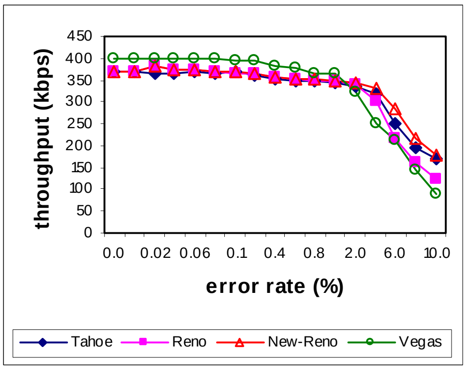
III. NUMERICAL RESULTS

Simulations run for each TCP version under four mechanisms, that is, *a.* TCP only; *b.* TCP with ELN; *c.* TCP with Snoop; *d.* TCP with both Snoop and ELN.

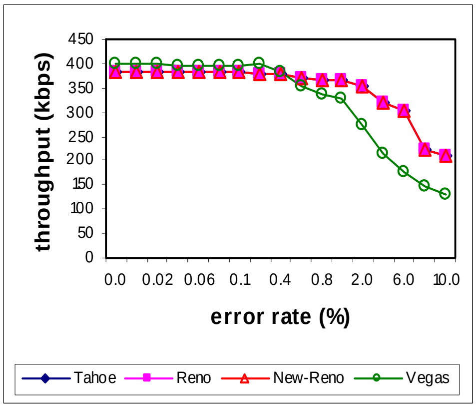
A. Throughput

Figure 4 (a) to (d) give the average throughputs of five sessions versus different packet loss rates for four mechanisms, respectively. Tahoe, Reno and New-Reno basically have very similar performance due to the small changes among their implementations. Vegas overall has better average throughput performance compared to other TCP versions when the error rate is not very high (less than about 0.5%). This is probably because Vegas uses more accurate RTT estimate to decide when to retransmit. However, when the error rate becomes higher, it is difficult to accurately estimate RTT and thus Vegas losses its advantages.

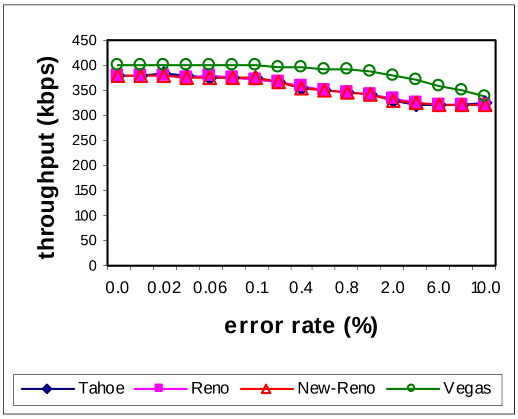
In Figure 2 (a), in the case of TCP only, Vegas provides better performance than Tahoe, Reno and New-Reno until packet loss rate increases to 2.0%. Performance of Vegas TCP becomes worse than other TCP versions when loss rate is higher than 2.0%. In Figure 2 (b), in the case of TCP with ELN, Vegas again provides better performance than the others when loss rate is less than around 0.4%. But after a packet loss rate of 0.6%, the average throughput of Vegas drops much more quickly than those of other TCP versions. In Figure 2 (c) and (d), in cases of TCP with Snoop and TCP with both ELN and Snoop, Vegas TCP consistently has better performance than the others though the performance gain is insignificant as the error rate increases to 10%.



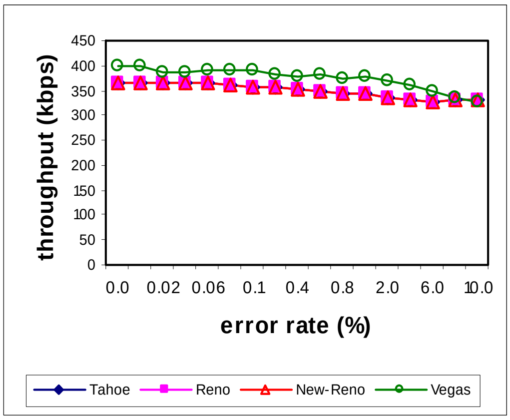
(a) TCP only



(b) TCP with ELN

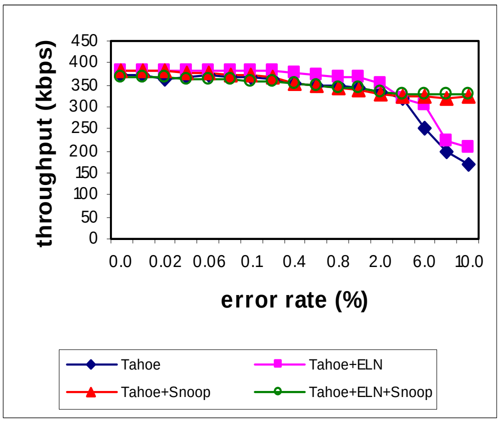


(c) TCP with Snoop

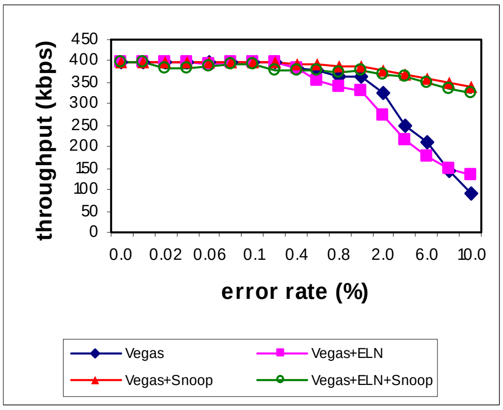


(d) TCP with Snoop and ELN

Figure 2. Comparision of throughputs for different TCP variants



(a) Tahoe



(b) Vegas

Figure 3. Comparison of throughputs for different mechanisms

Figures 3 (a) and (b) compare the average throughputs of different mechanisms for Tahoe and Vegas, respectively. As the performance of Reno and New-Reno are similar to Tahoe, their performance is represented by the performance of Tahoe in Figure 3 (a). Figure 3 shows that, for any TCP versions, both the Snoop protocol and ELN are effective mechanisms to improve throughput performance of TCP. With loss rates less than 2.0%, Snoop and ELN have similar performance. However, when the packet error rate is higher than 2.0%, Snoop has better performance than ELN. With Snoop, the throughput drops much more slowly than TCP or TCP with ELN. In addition, as shown in Figure 3 (a), Tahoe TCP with both ELN and Snoop has the best performance compared to other mechanisms (the same for Reno and New-Reno). This is because Snoop cannot shield all packet losses over the wireless link.

B. Unfairness

The throughput for individual TCP connections has been obtained through simulation using ns2. In this paper, we define *unfairness* as the difference between the maximum and minimum throughputs among five sessions divided by the minimum throughputs. That is,

$$unfairness = \frac{throughput_{\max} - throughput_{\min}}{throughput_{\min}} \qquad (1)$$

Figure 4 gives the unfairness measure for Tahoe and Vegas. It shows that Vegas treats TCP connections somewhat unfairly. The *unfairness* of Vegas is much worse than other TCP versions in general (Reno and New-Reno have similar performance with Tahoe). At an error rate of 10%, one connection takes 300% more bandwidth than another. This means some MHs enjoy their high throughput while others are starving. Figure 4 also shows that the unfairness of Tahoe among individual sessions becomes worse as the packet loss rate becomes higher. At an error rate of 10%, the unfairness of Tahoe is 70%.

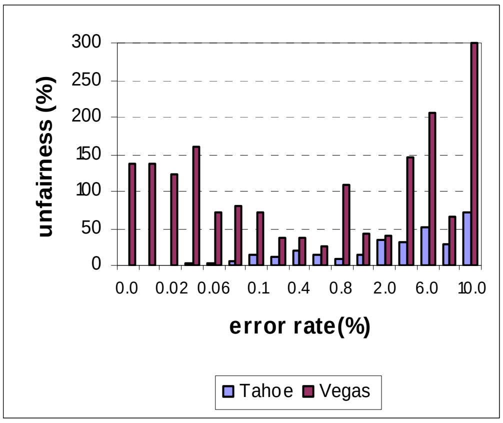


Figure 4. Unfairness of TCP versions

Figure 5 gives the unfairness measure when improvement mechanism Snoop is employed for Tahoe and Vegas. Figure 5 shows that the Snoop cannot guarantee the fairness among TCP connections though the unfairness situation is slightly improved. At an error rate of 0.8%, the unfairness of Vegas is 50% and the unfairness of Tahoe is 8%. It should be noted that we use the same Round Trip Times (RTTs) in the simulations. With different RTTs, the unfairness behavior will become worse. The unfairness with other improvement mechanisms (i.e. with ELN and with both ELN and Snoop) has similar behavior as Snoop as shown in Figure 5.

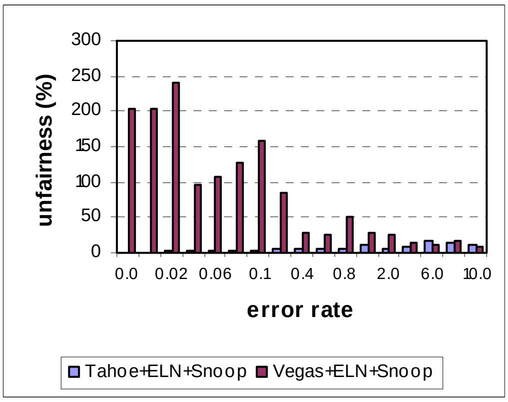


Figure 5. Unfairness of TCP versions with ELN and Snoop

IV. CONCLUSION

In this paper, the performance of various combinations of four TCP versions with two improvement mechanisms in a wireless network is examined. The results show that TCP Tahoe, Reno and New-Reno have very similar performance and perform more stably in terms of throughput and fairness than Vegas. Vegas TCP has slightly better throughput performance than others while its fairness performance is much worse than the others. The results also show that Snoop and ELN can effectively improve the throughput of TCPs. Significantly, Snoop performs better than ELN when the packet loss rate becomes higher than around 2.0%. However, the improvements cannot guarantee the fairness among multiple TCP connections. Link error and congestion could lead to serious unfairness among the connections. Mechanisms which can not only improve the throughput but also provide the fairness among multiple TCP connections are needed.

REFERENCE

[1] V. Jacobson, “Congestion Avoidance and Control,” Proceedings of the ACM SIGCOMM 88, August 1988.

[2] F. Khafizov and M. Yavuz, "Running TCP over IS-2000," Proceedings of the IEEE International Conference on Communications, 2002, pp. 3444-3448.

[3] G. Xylomenos, G. Polyzos, P. Mahonen, and M. Saaranen, “TCP Performance Issues over Wireless Links,” IEEE Communication Magazine, April 2001, pp. 52-58.

[4] H. Balakrishnan, V. N Padmanabhan, S. Seshan, and R. H. Kaze, “A Comparison of Improving TCP Performance over Wireless Links”, IEEE/ACM Trans. on Networking, Vol. 5, no. 6, pp. 756-769, December 1997.

[5] H. Balakrishnan and R. H. Kaze, “Explicit Loss Notification and Wireless Web Performance”, IEEE Globecom’98, Sydney, Australia.

[6] J. Pan, J. Mark, and X. Shen, “TCP Performance and Its Improvement over Wireless Links”, Proceedings of IEEE Globecom 2000, pp. 62-66.

[7] A. Bakre and B. R. Badrinath, “I-TCP: Indirect TCP for Mobile Hosts”, Proceedings of the 15th Int. Conf. Distributed Computing Syst., May 1995.

[8] H. Balakrisnan, S. Seshan, and R. H. Katz, “Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks,” ACM Wireless Networks, vol. 1, Dec. 1995.

[9] NS-2 simulation tool home page. http://www.isi.edu/nsnam/ns/, 2002.

[10] W.R. Stevens, “TCP/IP Illustrated, Volumn 1: The Protocols”, Addision-Wesley Publishing C., New York, 1994.

[11] K. Fall and S. Floyd, “Simulation-based Comparisons of Tahoe, Reno, and Sack TCP”, available online, http://www.icir.org/floyd/papers/sacks.pdf.

[12] L. Brakmo and L. Peterson, “TCP Vegas: End-to-end congestion Avoidance on a global Internet”, IEEE J. Select. Areas Communications, vol. 13, pp. 1465-1480, Oct. 1995.

[13] H. Jang, Y. Suh, “A Flow Control Scheme for Improving TCP Throughput and Fairness for Wireless Networks”, Proceedings of IEEE Wireless Communications and Networking Conference, 2003.

[14] Hengartner, U., Bolliger, J., and Gross, T., “*TCP Vegas revisited*”, Proc. of IEEE Infocom 2000, pp. 1546-1555, rch 2000.