Improving Wireless Snoop Performance Using Fake ACK Technique

By: Furat Asmat Moojid, Mohamed Hadi Habaebi, Dr. Borhanuddin Mohd Ali
Department of Computer and Communications Systems, Faculty of Engineering, Universiti Putra Malaysia

Abstract

Because of the burgeoning increase in data communication and multimedia services over wireless links, many approaches have been undertaken to find effective integrated protocols that satisfy these demands. One of them is the Snoop protocol that uses buffers at base stations to cache and copy packets passing in both directions and to retransmit lost packets locally, therefore avoiding congestion control mechanisms. Snoop was originally proposed by the University of California at Berkeley for improving the performance of TCP over wired and wireless LANs.

In this paper we modified the snoop ACK procedure using a fake ACK technique to improve its performance. This technique avoids end-to-end waiting time for lost packets and TCP slow start initiation at the source, resulting in more efficient utilisation of network resources. Simulation results show that using this fake ACK technique leads to improved throughput and a decreased number of dropped packets compared to the original Snoop.

Keywords: Snoop Protocol, Fake Acknowledgment.

I. Introduction

One of the most important transport protocols, Transmission Control Protocol (TCP), detects packet loss using sender timeouts or by receiving duplicate acknowledgments. When this loss occurs, TCP will invoke its congestion control mechanism, whatever the reason of the losses; e.g. congestion in wired links or errors in wireless links. This mechanism will degrade the performance of TCP because of a dropping transmission rate and as a result, uses less than the available bandwidth and throughput. (e.g. Slow Start [10]) and backing off its retransmission timer (Karn's Algorithm [11]). Recently, several schemes have been proposed to alleviate the effects of non-congestion-related losses on TCP performance over networks that have wireless or similar high loss links [12, 13, 14]. These schemes choose from a variety of mechanisms, such as local retransmissions, split-TCP connections and forward error correction, to improve end-to-end throughput.

One of the proposed schemes to solve this problem on wireless networks was the Snoop protocol that makes the sender unaware of the losses that occurred in the network [1].

In this paper, we propose a technique called Fake ACK (Acknowledgment) to replace the Snoop protocol ACK procedure at the base station to improve TCP performance. This technique limits the retransmission path to the wireless link and leads to a decrease in sender waiting time and a reduction in the probability of congestion control mechanism initiation. It also reduces the total number of dropped packets and increases throughput. Another feature of the modified Snoop is that the end-to-end TCP semantics would still be unchanged.

The rest of this paper is organised as follows. Section II presents the original Snoop protocol and Section III describes the proposed Fake ACK technique. Section IV introduces the simulation models over NS-2 together with the simulation results. The paper is then concluded in Section V.

II. Original Snoop Protocol

It is desirable to improve TCP performance in a wireless network without any modification to the fixed hosts. Snoop does that efficiently [2], by modifying only the base station and adding a module called a Snoop agent. The modifications are made only to the routing code at the base station without changing TCP semantics as shown in Figure 1. The Snoop protocol is a link layer protocol which uses buffers at the base station to cache packets passing in both directions, to retransmit unacknowledged packets and to avoid unnecessary timeouts.

There are two procedures that are used in the Snoop protocol: the Snoop data procedure that processes packets from a fixed host, and the Snoop ACK procedure that monitors and processes acknowledgements sent from the mobile host. We will focus on the second procedure in our work.

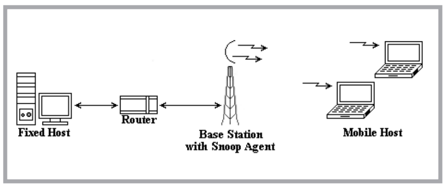

Figure 1: Modified module at base station

When the Snoop agent receives a data packet from the fixed host, it will cache a copy of this packet, buffer it

—continued from page 18

and then forward it to the mobile host. The same process will occur when the snoop agent receives an ACK from the mobile host but in the reverse direction. When the data packet is lost over the wireless link, the receiver will send a duplicate ACK to the Snoop agent. In this case the Snoop agent will cache this duplicate ACK and suppress it. At the same time, the Snoop agent will retransmit the lost packets locally (over the wireless link only) without making the sender aware of this loss as shown in Figure 2.

As shown in Figure 2, when the sender sends packet D0 to the mobile host and once this packet reaches the Snoop agent at the base station, it will be cached, copied and buffered, and then forwarded to the mobile host. When this packet is received at the mobile host, it will reply with ACK0 to the fixed host, and the process will be repeated in the reverse direction.

Suppose that packets D1 and D2 are sent at the same time but D1 is lost over the wireless part of the network.

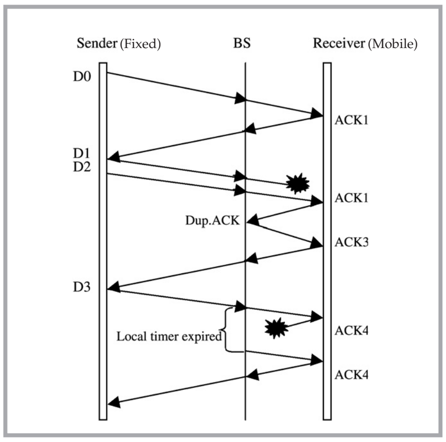

Figure 2: Original snoop protocol

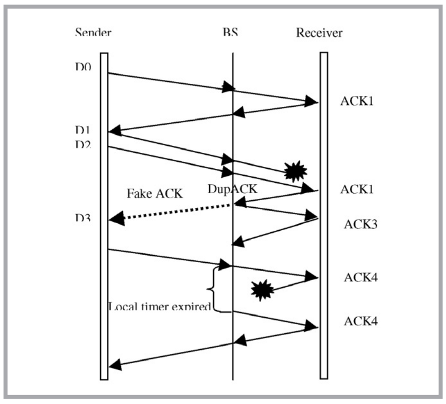

Figure 3: Fake ACK technique

This means the mobile host will receive D2 only and in this case it will send a duplicate acknowledgment ACK1 to the sender. But when the Snoop agent receives this duplicate ACK, it will be suppressed and the Snoop agent will retransmit the lost packet at the same time if it is cached to the mobile host with a reset local timer. This retransmission process will continue until the fixed host receives the right ACK.

Now suppose that packet D3 was sent from the fixed host and was received at the mobile host, and ACK4 was sent back from the mobile host and lost. When the Snoop agent's local timer expires, the agent will retransmit the lost packet until it receives the right ACK, which is then forwarded to the sender.

This method makes the sender unaware of this loss and therefore, prevents congestion control mechanism initiation.

III. Fake ACK Technique

The proposed modification to the Snoop ACK procedure is as follows (see Figure 3).

Consider when the fixed host has sent packets D1 and D2 at the same time. Suppose that D1 was lost over the wireless link. The mobile host receives packet D2 only and will reply back with a duplicate acknowledgment ACK1. When the Snoop agent receives this duplicate acknowledgment, it will forward a fake ACK to the fixed host and at the same time retransmit the lost packet until it receives the right ACK from the mobile host and therefore allowing the fixed host to continue its transmission.

The rest of the procedure is similar to the original Snoop including the Snoop agent local timer procedure. This technique improves TCP performance over a wireless link by utilising the bandwidth over wired links without invoking the TCP congestion control mechanism.

From Figures 1 and 2, the waiting time of the sender will be reduced drastically considering the time between packet D2's transmission time and packet D3's transmission time in

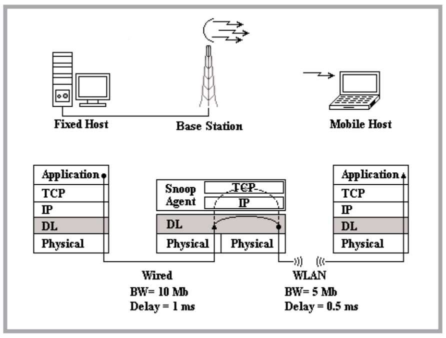

Figure 4: Simulation model

Table 1: Simulation parameters

Parameter	Value
Simulation time	10 sec
Network data type	FTP
Transport protocol	TCP Reno
Buffer queue size	100 packets
Max packet size	1000 bytes
TCP widow size	30 packets
Queue type	Droptail (FIFO)
Losses error rate	0.1

both cases. This will consequently increase the throughput of the link and more packets could be injected into the network.

The disadvantage of this technique is that the base station should have a bigger buffer because of the increased number of buffered packets (control and data alike).

IV. Simulation Models And Results

Network Simulator version 2 was used in this simulation study and the results were plotted using trace-graph. Our topology is divided into two parts, a wired link and a wireless link. The wired link uses a 10Mbps bandwidth with a 1ms delay and the wireless part uses a 5Mbps bandwidth with a 0.5s delay, as shown in Figure 4.

The rest of the simulation parameters are shown in Table 1.

This paper focuses on three factors as performance indicators: the link throughput or in other words the cumulative sum of sent packets, the total number of lost or dropped packets and the buffer size in both protocols, original Snoop and Fake ACK Snoop.

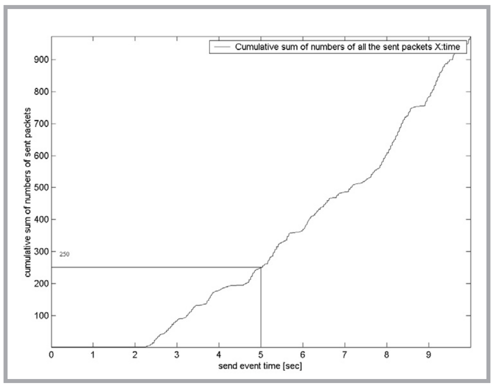

Figure 5a: Cumulative sum of sent packets for 5Mbps original snoop

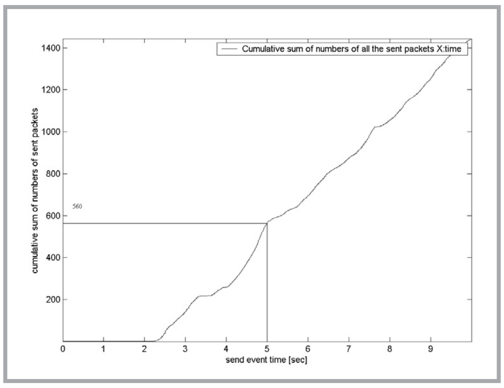

Figure 5b: Cumulative sum of sent packets for 5Mbps Fake ACK technique

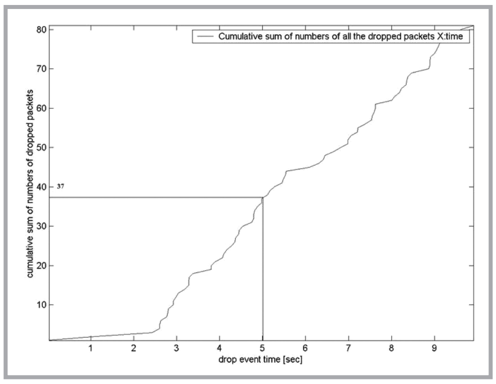

Figure 6a: Cumulative sum of dropped packets for 5 Mbps original snoop

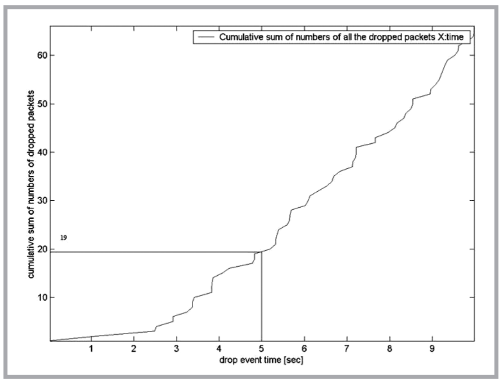

Figure 6b: Cumulative sum of dropped packets for 5Mbps Fake ACK technique

By applying the proposed Fake ACK technique, the throughput or the cumulative sum of sent packets increases as shown in Figures 5(a) and 5(b). We have chosen the 5th second as a comparison point. The results show that the original Snoop protocol throughput for 5Mbps bandwidth is about 250 packets, while the Fake ACK Snoop throughput for the same bandwidth reaches 560 packets. Changing the link bandwidth leads to similar improved results.

This throughput improvement minimises waiting time, and as a result, gives more packets a chance to be injected through the network.

Figures 6(a) and 6(b) show the number of lost or dropped packets plotted against their drop event time. The cumulative sum of the dropped packets reduces with the use of the Fake

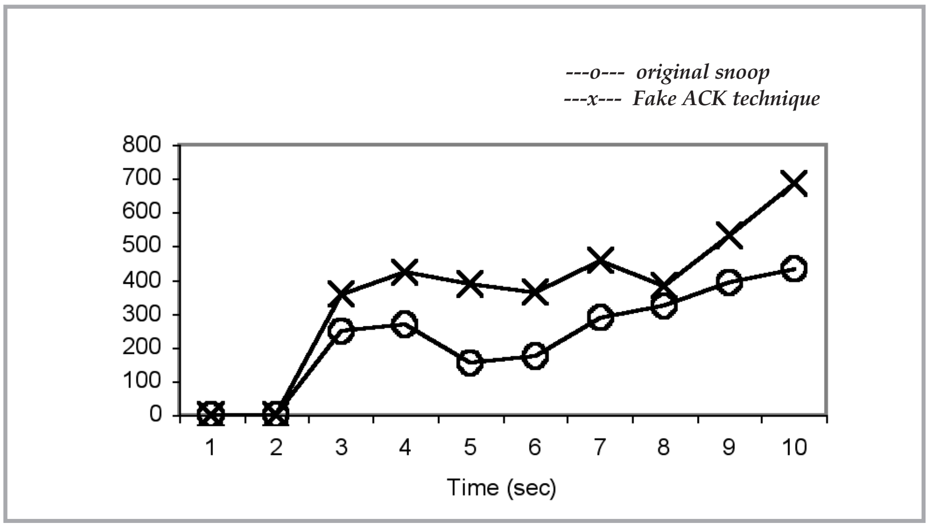

Figure 7: Comparison between buffered packets in original snoop and Fake ACK technique with 5Mbps.

ACK technique compared to the original Snoop. When a packet is dropped, TCP invokes its congestion control mechanism resulting in a lower system throughput. With the Fake ACK technique, the chance of initiating the congestion control mechanism is reduced because of the lesser number of dropped packets resulting in a higher system throughput.

In Figure 6(a) the cumulative sum of dropped packets at the 5th second is 37 packets with the 5Mbps original Snoop, while for the Fake ACK Snoop in Figure 6(b), the sum is 19 packets for same bandwidth.

Unfortunately, the Fake ACK is not an optimum technique because of the buffer size needed at the base station as illustrated in Figure 7. On average, Fake ACK Snoop requires twice as much buffer as the original Snoop. This might be acceptable from a designer's point of view when considering the savings in terms of accumulated throughput and the transparency of the protocol to end-to-end TCP semantics.

V. Conclusion

Many protocols have been proposed to minimise losses on wireless links by trying to make the sender unaware of these losses, but by using the TCP congestion control mechanism, TCP performance is degraded on wireless networks.

In the proposed Fake ACK technique, the Snoop agent located at the base station tries to reduce the sender waiting time by forwarding a Fake ACK and at the same time keeps resending the lost packet until it receives the right ACK, leading to improvement in the performance of wireless link throughput and a reduced dropped packets rate but with an increase in buffer size. ■

REFERENCES

[1] Hari Balakrishnan, Srinivasan Seshan, Elan Amir and Randy H. Katz, "Improving TCP/IP Performance over Wireless Networks," 1st ACM Int'l Conf. on Mobile Computing and Networking (Mobicom), November 1995.

[2] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan and Randy H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," IEEE/ACM Transactions on Networking, Vol. 5, No. 6, December 1997.

[3] George Xylomenos, and George C. Polyzos, "TCP Performance Issues over Wireless Links," IEEE Communications Magazine, April 2001.

[4] Amol Shah, "TCP Performance over Wireless Links," EE359 – Wireless Communications, Stanford University, December 12, 2001.

[5] Syed Natif Nawaz, and Joseph Toney, "Protocols for Improving Performance of TCP over Wireless Links." URL: http://www.cs.buffalo.edu/~qiao/cse620/wirelessTCP.ppt

[6] Janey C. Hoe, "Start-up Dynamics of TCP's Congestion Control and Avoidance Schemes," Masters thesis, Electrical Engineering and Computer Science, University of California at Berkeley, June 1995.

[7] Chi-ho Ng and Jack Chow, "Performance of TCP Protocol Running over Wireless LAN Network using the Snoop Protocol," Final Project Presentations, Spring 2001.

[8] Sarma Vangala and Miguel A. Labrador, "Performance of TCP over Wireless Networks with the Snoop Protocol," in Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN 2002), November 2002, pp. 600-601.

[9] Sally Floyd, "A Report on Recent Developments in TCP Congestion Control," IEEE Communications Magazine, April 2001.

[10] V. Jacobson, "Congestion Avoidance and Control," in Proc. ACM SIGCOMM88, August 1988.

[11] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and Sack TCP," Computer Communications Review, 1996.

[12] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," in Proc. 15th International Conf. on distributed computing systems (ICDCS), May 1995.

[13] H. Blakrishnan, S. Seshan, and R.H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," ACM Wireless Networks, 1(4), December 1995.

[14] R. Yavatkar and N. Bhagwat, "Improving End-to-End Performance of TCP over Mobile Internetworks," in Mobile 94 Workshop on Mobile Computing Systems and Applications, December 1994.

[15] Anne Aaron and Susan Tsao, "Techniques to Improve TCP over Wireless Links".