Improving TCP End to End Performance in Wireless LANs with Snoop Protocol

Dejan Jaksic, Zeljko Ilic and Alen Bazant

Department of Telecommunications, Faculty of Electrical Engineering and Computing
Unska 3, Zagreb, Croatia

{dejan.jaksic, zeljko.ilic, alen.bazant}@fer.hr, http://www.fer.hr/

Abstract— Many attempts have been made to improve TCP (Transmission Control Protocol) behavior in wireless environment. Networks with wireless links and mobile hosts violate many of the assumptions made by TCP, and consequence is low throughput and degraded performance. Snoop is simple protocol for improving TCP performance by modifying network-layer software only at access point without touching transport layer [1]. Only modification is taken at access point by putting something called snoop module. This paper includes a simulation-based performance analysis one of the versions of TCP protocol. The main idea is to develop our own TCP Snoop protocol and suggest some improvements for calculation of local timer at access point.

For avoiding unnecessary retransmissions and for improving TCP end-to-end performance on wireless link we have introduced a simple network layer addition called Improved Snoop protocol (ISp). In our version main goal is to find best estimation of time for local timer at access point. The idea is to send locally a few probe segments in wireless channel to the wireless client. In that way we calculate round trip time on wireless channel. Our method shows good results to determine TCP local timer value of snoop module and it is used in our simulation scenario. In original Snoop protocol the timer is set to fixed value.

Simulations show some improvments when the TCP-Snoop module is implemented. This mean that there are less segment losses and the system is more robust.

Key words: TCP, IP, Snoop protocol, WLAN.

1. INTRODUCTION

Wireless networks are becoming much spread and popular way of connecting various network devices. Despite that, problems with TCP (Transmission Control Protocol) in WLANs (Wireless Local Area Networks) still put them into not very reliable solutions. Communications over wireless links show quite different characteristics compared to traditional LANs. The bandwidth in WLANs is not so high like in wired networks and the bit error ratio (BER) is more exhibit, [4]. Most installed WLANs work in radio access mode and use the cell structure for covering and communication between wired and wireless domain. Earlier mentioned handover causes frequently delays and dropping packets. This time is between few milliseconds and few seconds, [4].

In wired networks TCP performs very well by adapting number of sending packets into a network with the congestion state on the network links. In that way TCP optimize network end-to-end delay. Basic principles of TCP are given in Section 3 but here are some short explanations. TCP is reliable transport layer protocol and provides reliability by maintaining a running average of estimated round-trip delay and by retransmitting any packet whose acknowledgement is not received before timeout, [4]. This assumption in WLANs causes a significant number of retransmissions and bad performance for the wireless clients.

Various link layer mechanisms have been proposed to improve the performance of TCP over wireless links. One of them is Snoop protocol. It provides reliable solution by maintaining TCP end-to-end connection while recovering the wireless link errors locally. There are many different versions of original Snoop, [1], and here is the one of them. For all these solutions one thing is common and that is a low price. Some of most familiar snoop architectures are explained in [7]-[9].

Disadvantage of snoop approach is to suffer from not being able to completely “hide” the sender from the losses in wireless channel. Also there can be some problems between transport layer and link layer. Some packets can be retransmitted at both layers. But the biggest complain about the Snoop protocol can be in increasing delay and jitter between hosts in wireless network.

This paper is organized as follows. In Section 2 we present an overview of TCP protocol. Section 3 describes the TCP-snoop protocol (ISp). Next section (Section 4) describes the TCP-snoop implementation and simulation results. Section 5 is conclusion and future work.

2. OVERVIEW OF THE TCP

TCP is the basic transport protocol for Internet network and LANs. It uses Go-Back-N protocol and a timer based retransmission mechanism. The timer period, (called timeout interval) is calculated based on the estimated round-trip-time (RTT). RTT is time between sending segment and getting back acknowledgment (ACK) for that segment. Basic mechanism for reliability is in using acknowledgments and retransmissions of segments. Segments whose acknowledgements are not received before the timer expires are retransmitted. In the presence of frequent retransmissions, TCP assumes that there is congestion in the network and invokes its congestion algorithms. These algorithms reduce congestion window size (CWND). Congestion window size is measure for capacity of the network. As the congestion window size is reduced, the transmission rate is also reduced. This window size adjustment technique prevents the source from overwhelming the network with an excessive number of segments. In described way TCP makes a flow control and avoids congestions in the network. Three basic TCP algorithms for avoiding congestion are:

- Slow start;
- Congestion avoidance, and
- Fast retransmission.

More about TCP mechanisms can be found in [3]. In the presence of high bit error rates in wireless links, TCP reacts the same way as in a wired link. It reduces the congestion window size before segment retransmission. TCP assumes that there is congestion in the network but that is not always true because most likely reason for loosing segments is unstable wireless link. This adjustment results in unnecessary reduction of the bandwidth causing significant performance degradation. When we say performance we are meaning on throughput and delays between source and destination node. There are few versions of TCP protocol like Tahoe TCP, Reno TCP, New Reno TCP, SACK TCP, Vegas TCP, [3]. In this paper our aim is to study how Tahoe TCP performs in WLANs. Reason for that is in fact that this is the version of TCP protocol with biggest degradation of performance in interaction with wireless links.

For good performance of TCP in WLANs there are three solutions:

- End to end solutions;
- Link layer solutions, and
- Splitting connection solutions.

TCP-snoop is the link-layer solution.

3. TCP-snoop PROTOCOL

The TCP-snoop protocol is Performance Enhancing Proxy (PEPs) method for reducing performance degradation caused with link characteristics, [2]. The main of TCP-snoop is to improve the performance of communication over unstable wireless channel without triggering retransmission and window reduction mechanisms at the transport layer.

The snoop module runs at the access point and provides a reliable solution maintaining the end-to-end semantics of the transport layer connection. TCP-snoop protocol uses link level buffers for storing passing segments, makes necessary local retransmissions for unacknowledged segments and in that way avoids false congestion. Furthermore, it filters duplicate acknowledgment from receiver (mobile host) to avoid unnecessary timeouts at the sender (fixed host). The module monitors every segment passing through access point forwards segments to their destinations and handles the corresponding acknowledgments.

Every TCP segment has a sequence number written in the first byte of segment header. Each sequence number is associated with acknowledgment segment (ACKs). This number informs the sender with the sequence number of last byte successfully delivered to receiver. If the receiver receives the same acknowledgment sequence number more then once, usually three times, it supposes that the segment on which the ACKs is showing is lost. An ACK that contains a sequence number that is smaller than the sequence of the last received ACK is called duplicate ACK (DUPACK). When snoop module receives DUPACKs it retransmits lost segments locally without forwarding ACKs to the sender. Transport layer is not aware these lost and no congestion control algorithms are triggered by sender. Very important thing for snoop module is local timer for each TCP connection. When the timer expires snoop module retransmits the segments that have not been acknowledged yet. In most cases this timer has a fixed value but more effective is when is calculated depending on segment propagation time in wireless channel.

In other words, the idea s to intercept segments sent to mobile host and performs local retransmissions. Snoop module works with two methods:

- snoopData () and
- snoopAck ().

These two methods are used in our simulation model but with some modifications. For example, we assume that there are no losses in wired domain.

3.1. snoopData()

Method for caching and processing the data segments going to mobile host. It takes care of retransmitted segments as well as out of sequence segments. When segment arrives in sequence and has a sequence number greater than the previous segment, snoop module caches it at access point and forwards the segment to mobile host. For every segment RTT is calculated by using a local timer. If an out of sequence segment with sequence number smaller than the one already ACKs is obtained, this means a retransmitted segment from the sender reached the access point. If this segment has already been sent to the mobile host, snoop sends ACK back to the sender so that the sender timeout does not expire. Otherwise, it forwards the segment to the mobile host and caches this segment as being retransmitted by the sender.

3.2. snoopAck()

This method processes the ACKs received from the mobile host and performs retransmissions. When ACK arrives from mobile host, it can be spurious, genuine or dupack. If it is genuine it is regular (new) ACK, snoop clears ACKs buffer, estimates RTT and forwards the ACK to the sender (fixed host). If it is ACK with the sequence number smaller than the one previous received it is spurious ACK and is discarded. If a DUPACK arrives for a segment that is not in data buffer it is forwarded to the fixed host. If a DUPACK for a segment in the data buffer arrives it retransmits segment to the mobile host. If expected DUPACK arrives and access points knows that the segments is lost it is discarded preventing unnecessary retransmissions.

Simulations which are will be shown in rest of the paper shows Tahoe TCP performance in WLANs in different way.

4. TCP-snoop IMPLEMENTATION AND SIMULATION RESULTS

Network topology used in simulation of protocol TCP-snoop is shown in ‘’Fig. 1’’.

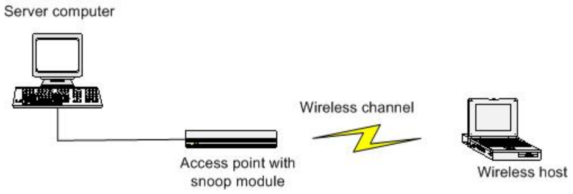
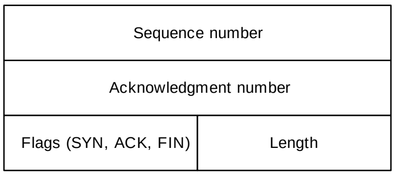


Figure 1: Simulation topology

In the role of the sender is server computer and receiver is wireless host. All entities of network topology are implemented in simulator as Java classes. Reason for choosing Java for simulator implementation is in object-oriented approach and possibility of interactive representation each object of network topology as program objects. In the simulation we used TCP segment shown in ‘’Fig. 2’’.



Sequence number	
Acknowledgment number	
Flags (SYN, ACK, FIN)	Length

Figure 2: TCP segment structure

Most complex component of simulator is server computer, where are retransmission and dynamic counter calculation mechanisms implemented. Snoop module is implemented via two earlier mentioned methods in access point class. In simulation we focused our attention on download direction of TCP segments. Reason for that is in fact that protocol TCP-snoop shows best characteristics for this direction and asymmetry of WLAN traffic. Also, we suppose that the wired link is error free.

For simulation the wireless channel is shown like two state Markov chain, so called Fritchman model, [5]. It has a good state (ON) and a bad state (OFF). During the good state all the segments are transmitted without losses and in bad state segments are dropped. ACKs are also lost during the bad state of wireless channel and losses of segments and ACKs are independent. Transfer to bad state is possible only at the end of the time slot, we do not considered BER. Errors are occurred at segment level, not at the bit level. Furthermore, simulator works in time cycles. Time is represented with timer counter which is incremented after each time cycle. Assumption is that in one time cycle each node (wireless client, access point and server computer) can receive process and send only one segment. All the segments used in simulation are set to MSS (Maximum Segment Size) given by user. Only last segment can be smaller than MSS depending on file size for sending. MSS is calculated like in ‘’(1)’’:

$$MSS = MTU - IP_{header} - TCP_{header} \tag{1}$$

MTU is maximum transmission unit for transfer over specified network. MSS in simulation is set to 40 octets, wireless channel propagation for all segments is set to 100 ms and file for transfer between wired and wireless domain is set to 1600 octets. Propagation time on wired link between server computer and access point is set to 10 ms. Loss probability on wireless channel is set to 10 %. Maximum number of retransmissions for server computer is set to 12. In this way probability of dropping connection is minimal. If number of retransmissions reaches the threshold connection is lost and the message of timer expiration is written in simulation log.

For better simulation results large number of simulations was made. In the simulation stack we do not modify network-layer except at the access point. Only one sender and one receiver are implemented in simulator because of simplicity. The extension to multiple receivers and senders is left for future improvements. These improvements can be in analyzing a problem of *handover* between two access points when snoop module is running at both access points.

4.1.1. *Calculation of the local timer*

In regular version of Snoop protocol, [1], local timer is called *persist timer* and set to fixed value. In our simulation there are two mechanisms for ISp:

- With random calculation of timer value, and
- with round-trip delay (round-trip timer) estimated from the time of sending a data segment to the time the acknowledgment is received.

In this paper only results concerning the second method are shown because it gives better performance of CWND and segment sequence number as a function of time. We calculate the timer using Karn's algorithm, but with some modifications. Principle used in simulation environment is given in figure (''Fig. 3'').

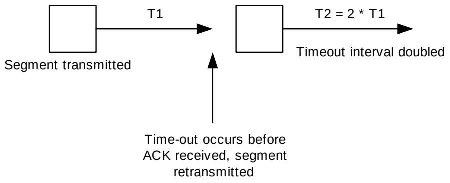


Figure 3: Calculation of local counter at access point

Probe segments are used initially for determing the maximum retransmission time for client. Basic principle is same as for regular TCP (without snoop module), [1], but we define maximum value for the local timer (retransmission timer of unacknowledged TCP segments) at access point which is smaller than server maximum retransmission value. When a new segment arrives from fixed host, the snoop module adds it to its buffer and forwards the packet to mobile host. Also, when a segment loss is detected (with duplicate ACKs or by a local timeout), it retransmits the lost segment to the mobile host. In that way the sender is shielded at access point from losses in wireless channel and no congestion control algorithms are triggered. This assumption works well only for simulation purpose and is taken for simplicity of implementation. For real wireless environment, propagation of each segment depends on state of wireless link. It is important to say that there is threshold for number of local retransmissions after which connection is dropped. One of characteristic of wireless link is very complex distribution of channel state in time. Taken wireless channel model maybe is not the best one, but it can simulate some principles of real wireless channel. Original approach for timer problem can be find in [1].

Some elements that can effect on time of propagation are:

- path loss;
- multipath propagation;
- shadow fading, and delay dispersion of signal.

''Fig. 4'' shows improvement on CWND when snoop module is active in access point and when is not for slow start phase of TCP protocol. Simulation environment is the same for both cases. As expected, when snoop module is activated congestion window size improvement concerning number of segments that sender can send is significant.

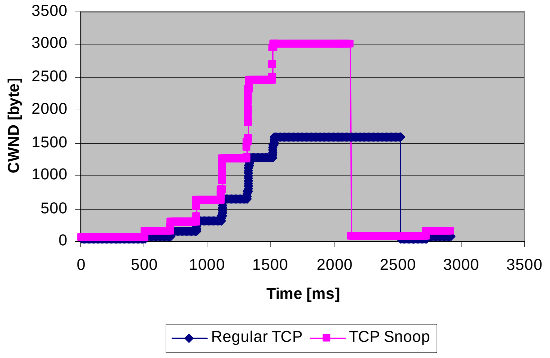


Figure 4: Congestion widow size vs. time

Furthermore figure ("Fig. 4") shows the difference between the values of timer value when snoop module is active and when is not. For case when snoop is activated the smaller value of calculated retransmission timer causes earlier beginning of slow start phase, but the size of congestion window is bigger for all the period of time.

For active snoop module, sender can send about 75 TCP segments while for opposite case this number is about 38. Segments caching at access point prevents congestion window from shrinking. It is the most critical factor in improving the performance of data transfers, [2]. Shrinking the congestion window has a ''cyclic' effect. It causes TCP sender to send less payload in each segment, which implies sending more data segments across the network. Since sending more data segments increases the chances of segments getting corrupted, this, in turn will shrink the congestion size further, and the ''cycle'' repeats.

Figure (''Fig. 5'') shows segment sequence number as a function of time.

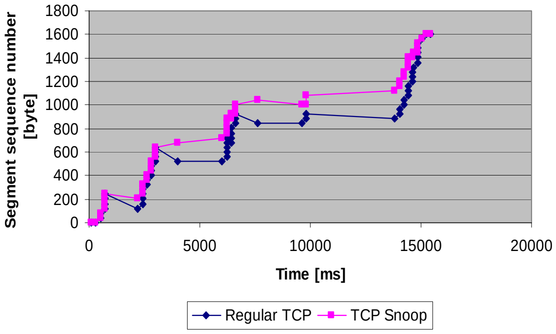


Figure 5: TCP segment sequence number vs. time

The sequence number ends with 1602 because the sequence number starts with 1 and is incremented for every byte of data sent across the wireless link. Also there is one extra byte for SYN and FIN segment. It is obvious that without active snoop module, the TCP layer at sender side has to retransmit the same sequence number several times. In ideal situation (without loss of segments) the graph will be linear.

It is important to say that for ISp local timeout interval is calculated absolutely separately from retransmission timeout at server side. One of disadvantage of ISp is in increasing of simulation time. Reason for that is in slow search of specific segments in local buffers at access point. In comparison with other similar work, [10], approach for calculation value of local timer ISp shows greater efficiency making the communication between two domains ''more natural'' and more flexible to the wireless link.

Figure ("Fig 6") shows one more comparison of congestion window size as a function of time. For this case timer value when snoop module is activated is almost the same when is not. Important thing to say is that the all mentioned results are product of few simulations and the best results are taken.

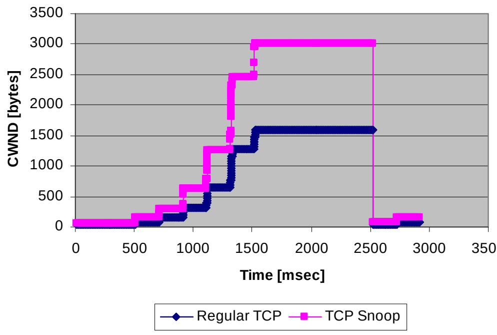


Figure 6: Congestion window size vs. time

5. CONCLUSIONS AND FUTURE WORK

Our simulation results indicate that the proposed improved Snoop protocol (ISp) is improvement in various snoop approaches. It is not tested for multiple connections and this is left for future work. Goal of the paper was to propose a solution for better calculation of local timer at access point, and we succeeded.

The ISp improves TCP performance by preventing the transport layer from reducing the congestion window size. Improvements to the current ISp implementation may include testing how ISp works on some other versions of TCP protocol in WLANs (for example TCP Vegas). It is interesting to say that the TCP SACK shows the best performance without snoop, but with snoop module it is the TCP version with worst performance in WLANs, [10].

This improvement in comparison with other related snoop solutions, [11], in some cases gives better results. The most important thing to determine the ISp efficiency is the model of wireless channel. So, the next thing to do will be to analyze behavior of ISp with some other known wireless channel models.

REFERENCES

[1] H. Balakrishnan, S. Seshan and R. H. Katz, ''Improving reliable transport and handoff performance in cellular wireless network'', ACM Wireless Networks, vol 1, no. 4, pp 460-481, Dec 1995.

[2] Performance Enhancing Proxy (PEP) Request for Comments: http://community.rocxen.com/developers/idocs

[3] W.R. Stevens, ''TCP/IP Illustrated'', vol. 1, Addison Wesley Press, 1994.

[4] E. Amir, H. Balakrishnan, S. Seshan and R.H. Katz, "Efficient TCP over Networks with Wireless Links," Proceedings of HotOS-V, May 2005

[5] J. Arauz, S. Banarjee and P. Krishanamurthy, ''MAITE: a scheme for improving the performance of TCP over wireless channels'', In Proceedings of Vehicular Technology Conference, pp 252-256, 2001.

[6] H. Balakrishnan, V. Padmanabhan, S. Seshan and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", ACM/IEEE Transactions on Networking, Vol. 5, No. 6:756-769, 1997.

[7] Hee-Jin Jang and Young-Joo Suh, "A Flow Control Scheme for Improving TCP Throughput and Fairness for Wireless Networks", IEEE Wireless Communications and Networking Conference, Vol. 4, pp 999-1003, March 2003.

[8] A. Patil, "A Snoop for Every Node", http://www-106.ibm.com/developerworks/wireless/library/wi-snoop/, June 2002.

[9] Jian-Hao Hu, K. L. Yeung, S. C. Kheong and G. Feng. "Hierarchical Cache Design for Enhancing TCP over Heterogeneous Networks with Wired and Wireless Links", IEEE Global Telecommunications Conferences, no. 1, pp 338-343, Nov 2000.

[10] S. Vangala and M. A. Labrador, "The TCP SACK-Aware Snoop Protocol for TCP over Wireless Networks", IEEE Semiannual Vehicular Technology Conference, 2003.

[11] C. Ho Ng and Lj. Trajkovic. "Performance Evaluation of TCP over WLAN 802.11 with the Snoop Performance Enhancing Proxy", OPNETWORK Conference Washington, DC, August 2002.