International Journal of Emerging Technologies in Engineering Research (IJETER)
Volume 5, Issue 9, September (2017) www.ijeter.everscience.org

Delayed ACK Approach for TCP Performance Improvement for Ad Hoc Networks Using Chain Topology

Prashant Kumar Gupta
M.Tech. Scholar, Computer Networks, Bhilai Institute of Technology, Durg (C.G.), India

Dinesh Bhawnani
Sr. Assistant Professor, Department of Computer Science and Engineering, Bhilai Institute of Technology, Durg (C.G.), India.

Dr. M.V. Padmavati
Professor, Head of Department, Computer Science and Engineering, Bhilai Institute of Technology, Durg (C.G.), India

Abstract – TCP suffers from poor wireless channel characteristics due to which performance is decreased in an ad hoc network. In previous studies we have found that TCP data consume as much as Wireless Recourses as TCP ack packet which is used during transmission by the sender and receiver. In TCP transmission for each data packets, receiver generates one ack and if the ack and data packets uses the same path, than they can cause collision at some point of time. It will directly affect the throughput of the TCP. TCP delayed acknowledgment is a technique used by some implementations of the Transmission Control Protocol in an effort to improve network performance. Before explaining what Delayed ACK is, let's start by clarifying some concepts.

Index Terms – Ad hoc network, TCP throughput, Delay Acknowledgement, Chain topology, MAC protocol.

1. INTRODUCTION

In standard TCP, the receiver generates one ack for each data packet, or more popularly, two in order data packets with the standard delayed ack option [1]. This mechanism works well in wired networks [1]. In multi-hop wireless networks, however, this mechanism can be further improved due to:

- Since the data and ack packets usually take the same route (or spatially close routes), they interfere with each other. The interference increases with the number of acks generated.
- Generating acks wastes scarce wireless resources. Though acks are essential to provide reliability, generating more acks than necessary is not desirable in wireless networks. Ideally, the receiver should generate minimal number of acks required for reliable data recovery.

TCP delayed acknowledgment is a technique used by some implementations of the Transmission Control Protocol in an effort to improve network performance. TCP/IP is used to refer to two different things. On one side TCP/IP is a suite of protocols that rule the Internet containing protocols like HTTP, Telnet, ICMP, SSH and many others. On the other side TCP/IP is also a pair of protocols themselves. TCP as a protocol in the transport layer, and IP as a protocol in the Internet Access Layer.

2. RELATED WORK

There is significant amount of literature available on performance of TCP over ad hoc networks employing the 802.11 protocol. Several proposals for improving TCP performance or replacing its mechanisms, over Adhoc networks have emerged in recent years. The strategy of these proposals is to enhance the TCP sender to react properly to lost packets caused by reasons other than congestion.

The research work from well-known researchers can be presented as: -

May Zin and Othman [2], present the rate-based pacing TCP with the delayed acknowledgement policy in the multi-hop wireless network. Through the simulation analysis they found the effects of delayed acknowledgement on TCP variants say Reno and Vegas with rate-based pacing. Although ACKs are used by TCP to ensure window flow control and reliability, generating more ACKs than necessary is not a desirable characteristic in wireless networks. Hence, a TCP receiver may enable the delayed ACK option to generate the optimal number of ACKs required for reliable delivery of data segments and improved TCP performance

Allman [5], showed that TCP performance might be hurt by delayed ACKs [10] during the slow start phase. Two mechanisms were presented to handle the side effects of delayed ACKs: delayed ACKs after slow start and byte counting. The former aims to speed up data rate recovery during slow start. The receiver only delays ACKs when slow start is over. This requires signaling between sender and receiver to inform the receiver about whether the slow start is active or not. Byte counting allows the sender to increase its cwnd based on number of bytes acknowledged by each ACK instead of the amount of ACKs. The results showed that both mechanisms can improve performance for implementations using delayed ACKs.

R. Oliveira and T. Braun [8], conformed that TCP should limit its congestion window as a function of the number of hops in place to achieve optimal performance. In particular, they showed that a short chain of nodes of up to 10 hops should have a congestion window limit of approximately 3 packets. This was shown to be caused by the limited spatial reuse property inherent in multi-hop networks relying on the IEEE 802.11 standard as the MAC protocol. In fact, this limitation is imposed by the hidden node problem present in such environments. The simulation results were quite encouraging by showing substantial improvements over various scenarios. As mentioned above, this paper proposes an enhancement to the initial mechanism that was not intended to highly noisy environments.

3. DELAYED ACK

A host that is receiving a stream of TCP data segments can increase efficiency in both the Internet and the hosts by sending less than one ACK segment per data segment received; this is known as a "delayed ACK". The delayed ACK strategy can be used by a TCP receiver. The delayed ACK strategy specifies that ACKs should be delayed in hopes of piggybacking the ACK on return traffic. When used, a TCP receiver must not excessively delay acknowledgments. Specifically, an ACK should be generated for at least every second full-sized segment, and must be generated within 500 ms of the arrival of the first unacknowledged packet. A delayed ACK gives the application an opportunity to update the window and perhaps to send an immediate response. In addition, a delayed ACK can substantially reduce protocol processing overhead by reducing the total number of packets to be processed.

The delayed acknowledgement strategy has been found to have a positive impact on TCP performance for bulk transfers and in wireless multi-hop networks. However, delay ACKs can also reduce performance in certain situations. TCP sender increases the amount of outstanding data based on the number of received ACKs. Therefore, the amount of data injected into the network is reduced as a result of reducing the number of ACKs.

4. CHAIN TOPOLOGY

The chain topology is the simplest representation of a wireless Adhoc network. Therefore, a theoretical analysis [6] of this observation should give a better understanding of TCP performance in Adhoc wireless networks. TCP throughput is independent of the maximum window size when the number of hop is fixed.

To understand how such collision may occur, let's take a scenario with the geographical range of interference and reception. Suppose the transmission range is about 250m and carrier sensing range as well as the interference range is about 550m.

Fig. 1 Chain Topology

5. RESULT AND DISCUSSIONS

The simulations scenario of chain topology of nodes consists of n nodes over a line separated by a distance of 200m. For each wireless node, the transmission range is 250m, the carrier sensing range is 550m and the interference range is about 550m.

We use the standard two-ray ground propagation model, the IEEE802.11 MAC, an Omni-directional antenna model of ns and an interface queue length of 50 at each node. We tested the NewReno version of TCP, which is the most deployed one. We tested four scenarios: 3and 9 nodes. The cases of 3 and 9 nodes required 150 secs per simulation (to obtain stationary behavior). A TCP data packet is taken to be of size 1040 bytes.

We have used ns2 to simulate and xgraph plotter to study and analyze the results. To manipulate and filter the required information from the trace files, generated in simulations, awk script has been used.

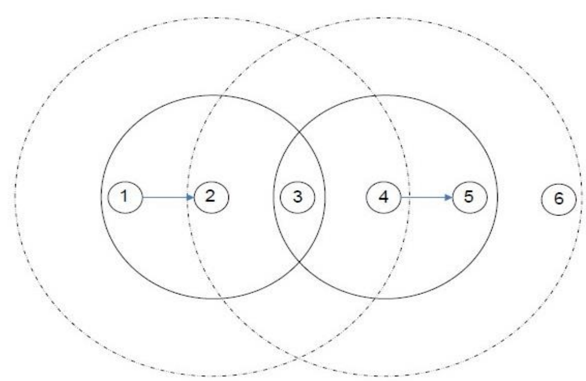

Fig. 2 Chain Scenario

In order to measure performance of standard TCP and Delay ACK TCP in 802.11 WLANs, the throughput and window size of each scenario is calculated and compared. We have done the simulations for two kinds of scenario, based on the number of nodes n, one for n=3 and another for n=9. We have implemented the standard TCP and delayed Ack TCP in both the cases and compared the window sizes putting maximum window size 2000 for n=9.

Fig. 3 shows the Simulations Scenario at Time 15.020819 sec 9 nodes

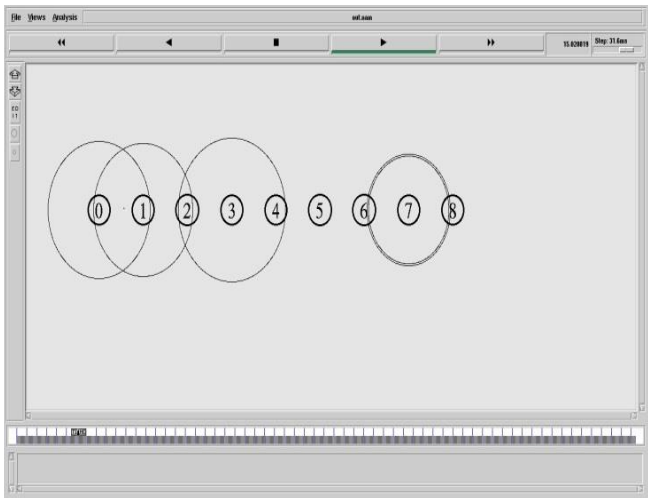

Fig. 3 Simulations Scenario at Time 15.020819 sec

Fig. 4 shows the window size evolution for standard TCP with maximum window size of 2000 and 9 nodes while Fig. 5 shows the window size evolution for DelAck TCP with maximum window size of 2000 and 9 nodes.

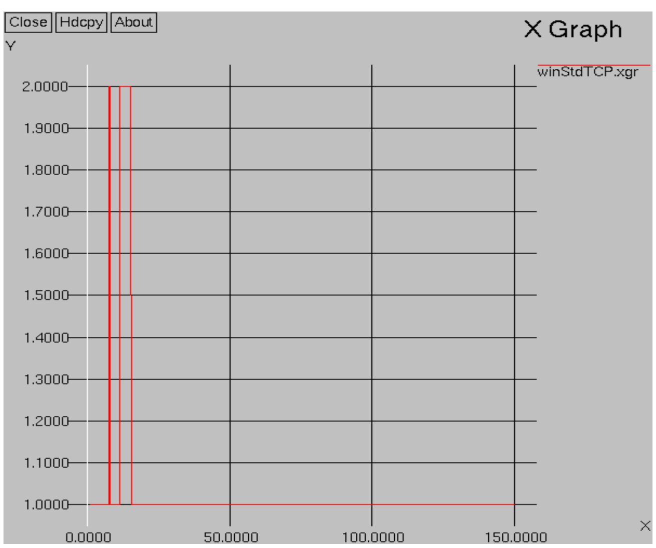

Fig. 4 Window Size Evolution for Standard TCP with Maximum Window Size of 2000 and 9 Nodes

Yet the most important conclusion from the curves is the robustness of the Delayed Ack options. In practice, when we do not know the number of nodes, there is no reason to limit the maximum window size to a small value, since this could deteriorate the throughput considerably.

When choosing large maximum window, the delayed ACK versions considerably outperform standard TCP. They achieve almost the optimal value that the standard TCP could achieve if it knew the number of nodes and could choose accordingly the maximum window.

We see that the standard Delayed Ack option slightly outperforms the standard TCP (yet with another value of maximum window size) for n = 9. A further improvement is obtained by the Delayed Ack with d = 3 (for n = 9). But the most important improvement that we see is that all delayed ACK versions are better than the standard TCP for maximum window sizes of more than 10, with the options of d = 3 or d = 4 outperforming the standard delayed ACK option. For n = 9, the Delayed ACK version with d = 3 is seen to yield between 30% to 40% of improvement over standard TCP for any maximum window sizes larger than 10. An even better performance of delayed ACK can be obtained by optimizing over the timer duration of the Delayed Ack option.

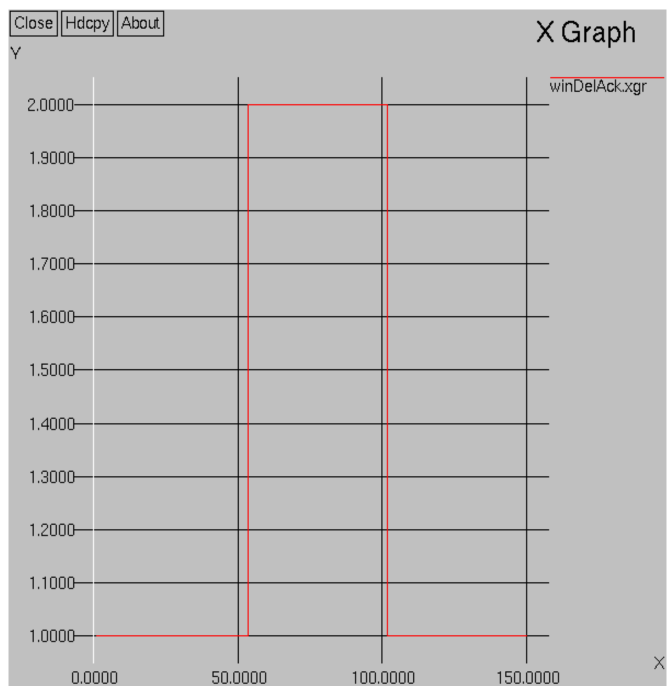

Fig. 5 Window Size Evolution for DelAck TCP with Maximum Window Size of 2000 and 9 Nodes

Fig. 5 shows the window size evolution for standard TCP with 3 nodes while Fig. 6 shows the window size evolution for delayed Ack TCP with 3 nodes.

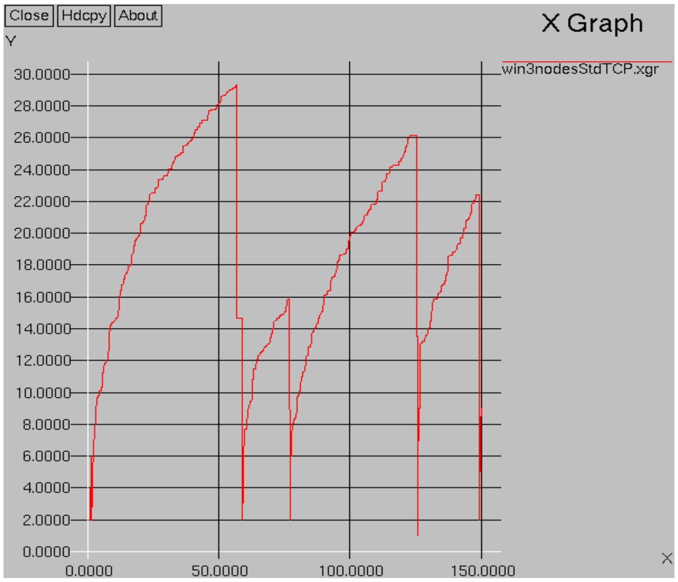

Fig. 6 Window Size Evolution for Standard TCP with 3 Nodes

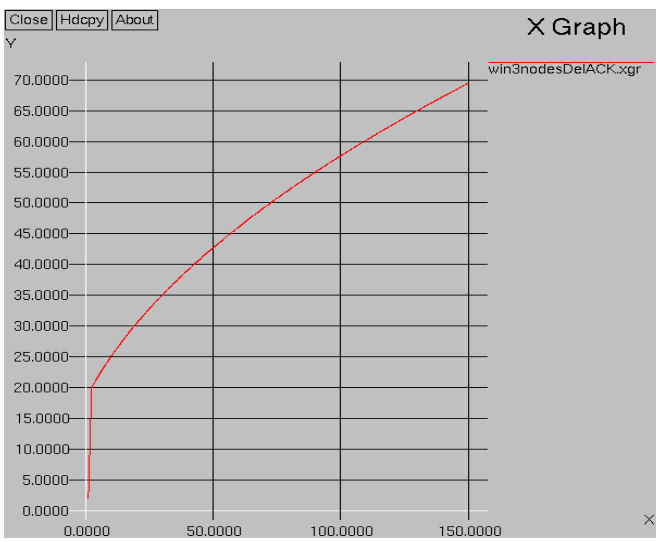

Fig. 7 Window Size Evolution for Delayed ACK TCP with 3 Nodes

From the Fig. 7 and Fig. 8, it is found that in standard TCP, losses are more frequent and more severe (resulting in timeouts) whereas the d=2 version of delayed ACK does not give rise to timeouts.

We observe from the Fig. 8 that the Delayed Ack option slightly outperforms the standard TCP (yet with another value of maximum window size) for n = 3, and largely outperforms (more than 10%) the standard TCP for n = 9. The Delayed ACK version is seen to yield between 30% to 40% of improvement over standard TCP for any maximum window sizes of 2000 in that range, it also outperforms standard TCP.

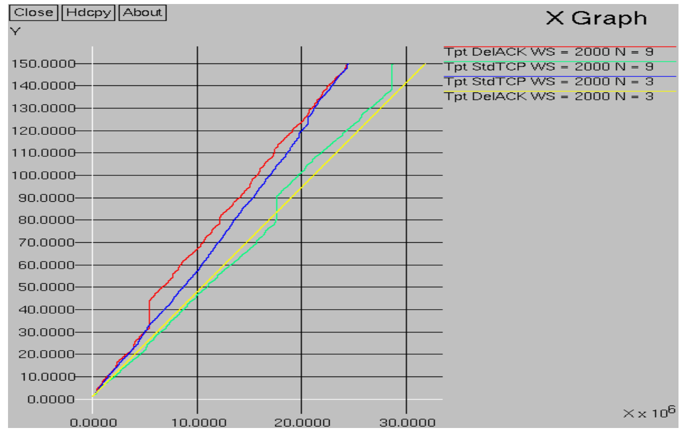

Fig. 8 Comparison of Delayed Ack and Standard TCP Throughput

6. CONCLUSION AND FUTURE WORK

In this paper, our focus is on a static ad hoc network that uses the IEEE 802.11 protocol for access. For such scenarios, the TCP performance is mainly determined by the hidden terminal effects (and not by drop probabilities at buffers) which limits the number of packets that can be transmitted simultaneously in the network (this is called the "spatial reuse"), in particular, for the chain topology. In this thesis, we have shown that Delaying Ack improves the performance of TCP over Adhoc networks that uses IEEE802.11 as MAC. The improvements are due to the fact that ACKs and TCP packets contend over the same channel, so decreasing the throughput of ACK can improve the throughput of TCP.

Delayed Ack inevitably triggers burst transportation at the sender. The burstiness increases the packet loss and potentially hurts TCP performance. Since TCP-DCA does not use large delay window except for short paths, the burstiness is limited. It is possible to further reduce the burstiness.

Our idea still holds and could be applied with different implementations. Existing work, which usually focuses on short-path networks, our evaluations include long-path networks that are valuable for large sensor networks. Our method could be further improved to suit different types of networks.

REFERENCES

[1] T Lang and D. Florani. 2001. The impact of delayed Acknowledgements on TCP performance over Satellite Links. Proc. ACM Workshop on Wireless Mobile Internet. ACM New York, USA. Pp. 56-61.

[2] M. Z. Oo, M. Othman and T. O'Farrell. 2016. A proxy acknowledgement mechanism for TCP variants in mobile ad hoc networks. Journal of Communications and Networks, vol. 18, no. 2, pp. 238-245.

[3] Altman, Eitan, and Tania Jiménez. 2003. Novel delayed ACK techniques for improving TCP performance in multihop wireless networks. IFIP International Conference on Personal Wireless Communications. Springer Berlin Heidelberg.

[4] S. R. Johnson. 1995. Increasing TCP Throughput by Using an Extended Acknowledgment Interval. Master's thesis, Ohio University, USA.

[5] M. Allman. 1998. On the Generation and Use of TCP Acknowledgements. In ACM Computer Communication Review, volume 28, pages 1114–1118.

[6] R. Oliveira and T. Braun. 2001. TCP in Wireless Mobile Ad Hoc Networks. In University of Bern, Technical Report IAM-02-003.

[7] K. Chen, Y. Xue, and K. Nahrstedt. 2003. On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks. In Proceedings of IEEE International Conference on Communications (ICC 2003), Anchorage, Alaska.

[8] R. Oliveira and T. Braun. 2005. A Dynamic Adaptive Acknowledgment Strategy for TCP over Multihop Wireless Networks. In Proceedings of IEEE Infocom, Miami, USA.

[9] Ammar Mohammed AI-Jubari and Mohamed Othman. 2010. A New Delayed ACK Strategy for TCP in Multi-Hop Wireless Networks, 978-1-4244-6716-7/10/ IEEE.

[10] V. Jacobson. 1990. Compressing TCP/IP headers for low-speed serial links. RFC1144, Feb 1990.