Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org
Volume 7, Issue 5, May (2017)

A Survey on Delayed ACK Approach for TCP Performance Improvement in Ad hoc Networks

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

Abstract – In ad hoc wireless networks, TCP suffers from performance deterioration due to poor wireless channel characteristics. Earlier studies have shown that the small TCP ACK packets consume wireless resources as much as the long data packets. TCP receiver generates one acknowledgment ACK for each data packet that is received as an acknowledgement to the sender. If the data packet and ACK uses the same path, then there may be collision between them due to channel contention. Moreover, generating acknowledgment for each data packet reduces TCP throughput. The main factor affecting the TCP performance in ad hoc wireless networks is the contention and collision between ACK and data packets caused by taking the same path. To solve this, lowering the number of ACKs using the delayed acknowledgment will improve TCP throughout. In this paper, we describe and analyse techniques to improve the performance of TCP over an 802.11 based network. The analytical results suggest that there can be a significant gain in TCP performance due to ACK thinning if the TCP receiver uses the delayed acknowledgement option of TCP.

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

1. INTRODUCTION

The success of the Internet can largely be attributed to the strength of its protocols, TCP/IP, which is used for lots of applications. TCP/IP should have profound usage in wireless networks where the Internet applications have expanded. However, being designed for wired networks, TCP's performance in wireless networks needs investigation. In the literature [1], a lot of research efforts have been put into improving TCP performance in wireless cellular networks. Recently, some researchers turn to another kind of wireless networks, i.e., multi hop wireless network. Such networks have no fixed infrastructure no centralized administration entity and are usually referred to as ad-hoc networks.

Ad-hoc network is becoming a prominent topic in wireless communications. This network is a complex distributed system that can be freely and dynamically organized. Recently, the emergence of new techniques, such as Bluetooth and IEEE 802.11 [2], is making possible the deployment of adhoc wireless networks for commercial purposes. As the technology of ad hoc networks evolves more and more, supporting transport control by these networks seems a challenging issue.

The basic TCP/IP protocol suite has been instrumental in developing today's Internet. In particular, TCP has been successful due to its robustness in reacting dynamically to changing network traffic conditions and providing reliability on an end-to-end basis. This wide acceptance has driven the development of many TCP applications, motivating the extension of this protocol to wireless networks.

Unfortunately, the design of TCP's congestion control that made it successful in today's Internet makes it perform less than optimal in wireless adhoc networks. This is due to TCP's inability to distinguish wireless losses from congestion losses. Traditional transport protocols [3] like TCP face severe performance degradation over adhoc networks given the noisy nature of wireless media as well as unstable connectivity conditions in place.

A crucial challenge faced by TCP over these networks is how to operate smoothly with the 802.11 wireless MAC protocol which also implements a retransmission mechanism at link level in addition to short RTS/CTS control frames for avoiding collisions. These features render TCP acknowledgments (ACK) transmission quite costly. Data and ACK packets cause similar medium access overheads despite the much smaller size of the ACKs.

Numerous enhancements and optimizations have been proposed to make TCP survive in such environment. Earlier studies [4] have shown that the interference and the collision between data and ACK packets, which is caused by taking the same route, increases with the number of ACKs generated. Furthermore, Generating ACKs wastes 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. In the original TCP, the receiver generates one ACK for each data packet received. With the standard delayed ACK option TCP can generate one ACK for two in-order data packets. This mechanism is working well in wired networks, however in adhoc wireless networks this mechanism can be further improved. This paper This paper attempts to find the effects of delayed ACK on TCP for adhoc networks. This is achieved by dynamically adjusting the TCP receiver delay window to the optimal value based on channel condition.

2. AD HOC NETWORK MAC PROTOCOL

IEEE 802.11 [5] is the standard Medium Access Control (MAC) protocol for ad hoc networks. In the IEEE 802.11 MAC layer protocol, the basic access method is the distributed coordination function (DCF). DCF is based on the mechanism of carrier sense multiple access with collision avoidance (CSMA/CA).

CSMA/CA protocol works on a "listen before starting transmission" scheme. Whenever there is packet to transmit by station, a station first senses the medium and ensures that the medium is idle for the specified DCF interframe space (DIFS) duration. If such station initially senses the medium to be busy, then the station has to wait until the medium becomes idle for DIFS period, and then chooses a random "backoff counter".

This backoff counter determines the amount of time the station must wait until it is allowed to transmit its packet. During the period in which the medium is idle, the transmitting station decreases its backoff counter. If during this time the medium becomes busy, its backoff counter is frozen. It can decrease its backoff counter again only after the medium is idle for DIFS.

3. TCP MODELLING OVER END TO END CHAIN TOPOLOGY

The purpose of this paper is to give an analysis of this observation. 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.

Figure 1: Chain Topology

Figure 1 depicts an 802.11 wireless adhoc network with chain topology. There are 3 nodes numbered 1, 2 and 3. Node 2 is in the transmission range of both node 1 and 3. Nodes 1 and 3 are not in each other’s transmission range. Node number 1 is the source of a TCP session controlling a long file transfer between node 1 and node 3. The nodes are assumed to be static and the channels are assumed to be always in good condition so that the routing is also static. The MAC layer implements the four-way handshake (RTS/CTS/DATA/ACK) for data transmission.

Each transmission of a DATA packet at the MAC level is part of a four-way handshake protocol. The node that wishes to send a packet, which is M1, first sends an RTS (Request to Send) packet to destination node, M2. If M2 can receive the packet, it sends a CTS (Clear to Send) packet. If M1 receives the CTS it can send the DATA packet (e.g. TCP data or ACK packet). Finally, M2 sends a (MAC layer) ACK so that M1 knows that the data packet has been well received. This handshake protocol is intended to reduce the probability of “hidden terminal” collisions but it does not eliminate them.

4. LITERATURE REVIEW

An important problem in such scenario is that of predicting the interaction between TCP and the 802.11 protocol. 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. We focus here on proposals that aim to minimize traffic overhead caused by redundant ACKs.

Lang and Florani [5] have investigated the impact of using del ACK strategy on TCP performance over satellite links with propagation delay and bit-error rates high. Through the simulation, they have shown that the utilization of del ACK causes worse

TCP performance compared with the use of basic acknowledgements.

May Zin and Othman [6] 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.

Altman and Jimenez [7] investigated the impact of delaying more than two ACKs on TCP performance in adhoc wireless networks. They proposed a limited dynamic delay ACK scheme, in which the receiver begins delaying one ACK only and keeps increasing until four based on the sequence number of the acknowledged packet. The receiver keeps delaying four ACKs except at session startup. Although their result was positive, it has been obtained for a single flow only.

Johnson [8] investigated the impact of delaying the ACK for different number of packets ranging from 1 to 20. The main outcome was that the distance between nodes plays an important role in selecting the optimal value of ACK delay window. The idea was that, in short-range networks, delaying the ACK for large numbers of data packets is always beneficial. On the other hand, this might not be appropriate for long-distance networks, especially if congestion is occurring. The longer the distance the longer the time for TCP sender to detect lost packets obtained by delaying more ACKs.

Allman [9] 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 signalling 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.

Olivera and Braun [11] produced TCP-DAA in which the receiver adjusts the delay window according to the packet loss event. They set the delay window limit as <= 4. When there is no packet loss, the TCP-DAA receiver delays the ACK until it receives more data packets up to four, but reduces the number to two in case of out-of-order packet arrival.

Chen et al. [12] proposed a scheme called TCP-DCA to select different delay window based on path length (number of hops). They found that TCP does not always get throughput gain by delaying unlimited ACKs and the maximal TCP throughput is achieved at a certain delay window that balances decreasing ACK flow and burst loss. TCP-DCA has shown good performance in static adhoc wireless networks.

Beizhong at al. [13] suggested that the TCP performance can be gained by making the receiver waits until the ACK-delay timeout event occurs, no matter how many in-order packets received during the timeout period. They use large delay window to insure the occurrence of timeout.

5. DELAYED ACK

TCP receivers generate an ACK for each incoming segment. These ACKs are cumulative and acknowledge all in-order segments that have arrived at the receiver. The redundancy provided by cumulative ACKs protects against ACK loss. If an out-of-order segment arrives, an ACK is transmitted. However, the TCP receiver does not ACK the incoming segment, instead it generates a duplicate ACK for the last in-order segment that arrived. An optional delayed acknowledgment mechanism can improve TCP performance.

Delayed acknowledgment (DA) [7] mechanism allows the TCP receiver to transmit an ACK for every two incoming packets. Performance of TCP can be improved using algorithms which allows the TCP receiver to transmit an ACK for every two incoming packets. Instead of sending ack for every successfully transmitted TCP packet, DA decreases the flow of ACKs so as to give more bandwidth to TCP data packets. DA version is better than the standard TCP for maximum window sizes of more than 3.

DA allows a TCP receiver to refrain from transmitting an ACK for every incoming segment. However, the receiver must send an ACK for every second full-sized segment. Since ACKs are cumulative, using delayed ACKs has little impact on transmission reliability. Furthermore, DA conserves resources by decreasing the load on the network and the machines which must generate and process these segments.

6. IMPACT OF DA ON TCP PERFORMANCE

Literatures reveal that when a TCP connection runs over a static chain topology the TCP throughput measured decreases rapidly when the number of hops increases from one, and then stabilizes when the number of hops becomes large.

This paper focus on impact of this type of collision on TCP performance. This mechanism aims at preventing the well-known hidden node problem. 802.11 works efficiently for topologies of at most 3 hops between sender and receiver, for larger scenarios in terms of number of hops, the hidden node problem still exists due to the spatial reuse property inherent in the propagation model of such wireless networks.

Basically, the spatial reuse imposes that within a certain geographical area only one node can transmit at a time [3]. This causes adverse impact on traditional TCP since it is always probing the network for bandwidth by increasing its transmission rate until a lost packet is detected. Hence, unless an efficient coordination between MAC and transport protocols is in place, the end-to-end performance can be severely impaired. There has been a belief in the research community that TCP can improve its performance by employing delayed ACK mechanism.

7. CONCLUSION AND FUTURE SCOPE

The focus of this paper 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

This allowed the authors to conclude that there is an optimal size for the maximum window of TCP. Increase in the maximum window size results in decreasing TCP throughput, although the loss in performance is not large (around 20%). TCP reliability requires that transmitted packets are acknowledged by the receiver side. However, if the receiver it is evident that techniques for delaying ACKs can be indeed efficient in ad hoc environments.

There are two reasons why the DA option increases the TCP good-put in wireless ad hoc networks. First, it cuts the amount of ACK packets roughly in half. This allows the data packet from the sender to meet less collision in the wireless link. Since the wireless channel is a shared resource for all neighbour nodes, a decrease in ACKs saves more bandwidth for the data packets. In fact, by using DA, the reduction for the number of ACK packets traveling in the network, many more nodes will benefit. The overall throughput of the network will show greater improvement than in our simple chain topology. We believe that the results in this paper will be useful towards the understanding of more complicated scenarios such as the performance of TCP over multi-hop ad hoc networks.

REFERENCES

[1] Chandran, Kartik, et al. "A feedback-based scheme for improving TCP performance in ad hoc wireless networks." IEEE Personal communications 8.1 (2001): 34-39.

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

[3] Dyer, Thomas D., and Rajendra V. Boppana. "A comparison of TCP performance over three routing protocols for mobile ad hoc networks." Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing. ACM, 2001.

[4] Liu, Jian, and Suresh Singh. "ATCP: TCP for mobile ad hoc networks." IEEE Journal on selected areas in communications 19.7 (2001): 1300-1315.

[5] Holland, Gavin, and Nitin Vaidya. "Analysis of TCP performance over mobile ad hoc networks." Wireless Networks 8.2/3 (2002): 275-288.

[6] Vaidya, Nitin H., et al. "Delayed duplicate acknowledgements: a TCP-Unaware approach to improve performance of TCP over wireless." Wireless Communications and Mobile Computing 2.1 (2002): 59-70.

[7] Chen, Jiwei, et al. "TCP with delayed ack for wireless networks." Ad Hoc Networks 6.7 (2008): 1098-1116.

[8] Kuang, Tianbo, and Carey Williamson. "A bidirectional multi-channel MAC protocol for improving TCP performance on multihop wireless ad hoc networks." Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems. ACM, 2004.

[9] Al Hanbali, Ahmad, Eitan Altman, and Philippe Nain. "A survey of TCP over ad hoc networks." IEEE Communications Surveys and Tutorials 7.1-4 (2005): 22-36.

[10] Nyangaresi, Vincent O., Solomon O. Ogara, and Silvance O. Abeka. "Evaluation of TCP Congestion Control Modus Operandi in Mesh Networks." International Journal of Computer Networks and Applications (IJCNA), 4(1), 2017, PP: 15-26, DOI: 10.22247/ijcna/2017/41293.

[11] Singh, Ajay Kumar, and Kishore Kankipati. "TCP-ADA: TCP with adaptive delayed acknowledgement for mobile ad hoc networks." Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE. Vol. 3. IEEE, 2004.

[12] Gupta, Abhinav, Ian Wormsbecker, and C. Wilhainson. "Experimental evaluation of TCP performance in multi-hop wireless ad hoc networks." Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. (MASCOTS 2004). Proceedings. The IEEE Computer Society's 12th Annual International Symposium on. IEEE, 2004.

[13] Kherani, Arzad Alam, and Rajeev Shorey. "Performance improvement of TCP with delayed ACKs in IEEE 802.11 wireless LANs." Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE. Vol. 3. IEEE, 2004.