

A THROUGHPUT ANALYSIS OF TCP IN ADHOC NETWORKS

S.P.Valli¹,K.M.Mehata²

¹vallisp@yahoo.com

Department of Computer Science and Engineering
B.S.Abdur Rahman University,Chennai.

²kmmeahata@bsauniv.ac.in

Department of Computer Science and Engineering
B.S.Abdur Rahman University,Chennai.

ABSTRACT.

Transmission Control Protocol (TCP) is a connection oriented end-end reliable byte stream transport layer protocol. It is widely used in the Internet.TCP is fine tuned to perform well in wired networks. However the performance degrades in mobile ad hoc networks. This is due to the characteristics specific to wireless networks, such as signal fading, mobility, unavailability of routes. This leads to loss of packets which may arise either from congestion or due to other non-congestion events. However TCP assumes every loss as loss due to congestion and invokes the congestion control procedures. TCP reduces congestion window in response, causing unnecessary degradation in throughput. In mobile ad hoc networks multi-hop path forwarding further worsens the packet loss and throughput. To understand the TCP behavior and improve the TCP performance over mobile ad hoc networks considerable research has been carried out. As the research is still active in this area a comprehensive and in-depth study on the TCP throughput and the various parameters that degrade the performance of TCP have been analyzed. The analysis is done using simulations in Qualnet 5.0

KEYWORDS:

TCP, Ad hoc networks, Retransmissions, Throughput.

1. INTRODUCTION

An ad hoc network consists of a collection of individual devices that directly communicate with each other. A mobile ad hoc network is a type of ad hoc network that can change locations and configure itself on the fly and requires no infrastructure for communication. Every node acts as a router and performs route discovery and route maintenance. The responsibilities for organizing and controlling the network are taken care by the nodes themselves. Nodes within radio range communicate directly via wireless links. Since mobile Ad Hoc network communicates using wireless media, it has restriction of communication such as interference of signal, noise and data loss against environment.

The main objective of this paper is to give an insight into the various parameters that affect the TCP throughput in an ad hoc network. The paper is organized as follows. Section 1 gives an introduction to ad hoc networks and their characteristics. Section 2 concentrates on overview of

TCP. Section 3 discusses problems and issues of TCP in mobile ad hoc networks. Section 4 presents the related work in this area of research. Section 5 gives a detailed discussion on the results of the throughput analysis based on simulations. Finally, Section 6 gives the conclusion.

2. TCP OVERVIEW

TCP is a connection-oriented, reliable, end-to-end, byte stream transport protocol designed to be used over unreliable internetworks. TCP has got an extensive application in Internet because of its mechanisms of reliable transmission, traffic control and congestion control. TCP reduces the size of congestion window to ensure reliable data transmission and reduce the degree of network congestion when packet loss event occurs. The congestion control mechanism of TCP is very effective in the low Bit Error Rate (BER) link such as wired link, because the probability of packet loss or error is very small in low BER link, in other words, almost all packet loss events in low BER link are caused by network congestion. But in a high BER link such as wireless link, this criterion that is used to determine network congestion by the congestion control mechanism of TCP is wrong.

TCP provides reliability by a mechanism of acknowledgment and re-transmission. TCP assigns a sequence number to each byte transmitted and starts a timer which is a function of the estimated round trip time. This is called a Retransmission Timeout (RTO) interval. When the receiver receives a packet with sequence number 'n' it sends an acknowledgment 'n+1' indicating the next expected sequence number [1]. If an acknowledgment is not received within the timeout interval the sender assumes it as a lost packet and the data is retransmitted. A sender can send at most a window's worth of packets without receiving ACKs.

The current implementation of TCP includes four main congestion control algorithms [2] such as slow start, congestion avoidance, fast retransmit and fast recovery. TCP uses two main window-based mechanisms: the receive window and the congestion window. The receive window is the receive-side limit and congestion window is the number of segments that will be sent before waiting for an acknowledgement. The window size is adjusted dynamically based on two TCP state variables: the congestion window (cwnd), and the slow-start threshold (ssthresh). Initially 'cwnd' is one segment and it is increased upon every ACK received. The increase is exponential in the slow-start phase (i.e., doubling cwnd every RTT, until ssthresh is reached), and linear in the congestion avoidance phase (i.e., increasing cwnd by one segment for every complete window's worth of data exchanged). TCP uses packet loss as an implicit signal of network congestion. When a retransmission timeout occurs 'ssthresh' is set to half the window size before packet loss started and 'cwnd' is set to one segment and the slow start phase is initiated. In order to avoid this, another commonly used strategy is fast retransmit which uses duplicate ACKs (typically 3) to trigger the retransmission of a missing packet, well before the retransmission timer expires. After retransmitting the missing segment fast recovery algorithm takes over.

3. TCP IN WIRELESS AD HOC NETWORKS

The performance of TCP degrades in wireless networks because the assumptions of TCP are with respect to wired networks and they may not hold the same in a wireless environment. In wired networks, buffer overflows are the main reason for most packet loss. When a packet loss event occurs TCP slows down the sending rate by adjusting its congestion window. Unfortunately, wireless networks suffer from other types of losses that are not related to congestion also. Some of the factors that have the most significant impact on the TCP performance and that lead to the performance degradation of the TCP have been analyzed in detail.

3.1 Mobility

In a mobile ad hoc network as mobile nodes moves out of the transmission range of the other nodes, route failure and link breakage occur leading to packet losses. However TCP cannot distinguish the cause of packet loss and assumes it as congestion and thereby congestion control is invoked. After a route failure if the route discovery process takes longer time than the RTO, the TCP sender will invoke congestion control after timeout. The throughput already reduced due to packet losses further decreases.

3.2 Multipath routing

Due to mobility the network topology frequently changes in ad hoc networks. The topology changes lead to route breakages and hence the route to a particular destination may need to be recomputed very often. This introduces some delay due to route recomputation and packets may not arrive in order. The TCP receiver misinterprets such out-of-order packet arrivals as congestion and thus generates duplicate ACKs. This causes the sender to invoke congestion control algorithms like fast retransmission.

3.3 Congestion Window

TCP considers the congestion window as a measure of the rate of transmission that is acceptable to the network. However in an ad hoc wireless network, congestion control is invoked for various reasons such as route failure, link breakage, network partitioning and so on. The congestion window size computed for the old route may be too large for the newly found route, resulting in network congestion if the sender still transmits at the full rate allowed by the old congestion window size.

3.4 Energy efficiency

Energy resources are limited in an ad hoc environment than in a wired network. Hence any scheme devised for such a network must be energy efficient.

3.5 Channel error

TCP data packets from the sender or acknowledgments from the receiver may be corrupted by bit errors, leading to the loss of TCP data packets or acknowledgements. This leads to misinterpretation as packet loss due to congestion and retransmission is done. In the case of an acknowledgement lost the TCP sender waits till the RTO to invoke congestion control procedures.

3.6 Medium contention and collision

There are three key problems in wireless environments the hidden terminal problem, the exposed terminal problem, and unfairness. These problems could be still worse in multi-hop ad hoc networks than in wireless because ad hoc networks are characterized by multi-hop connectivity.

To summarize some of the main factors that contribute to performance degradation of TCP in wireless networks are

- TCP is unable to differentiate losses due to route failure and losses due to congestion.
- Hidden and exposed terminal problems.
- Frequent Route failures caused by node mobility.
- Unpredictable channel errors.

- TCP unfairness

4. RELATED WORK

In this section we present various methods proposed in literature to enhance the performance of TCP in Ad hoc networks. Improvement and optimization of TCP performance have drawn significant attention in the areas of TCP.

TCP-ELFN [3] suggested an explicit link failure notification from the network to distinguish congestion and routing changes in a wireless environment. ATCP [4] monitor the change of the end-to-end network conditions to distinguish congestion from other networking event. All of these mechanisms are based on the idea of freezing states of TCP and keeping large congestion windows without decreasing the transmission rate at the occurrence of routing changes.

A number of TCP-related cross layer enhancement techniques have been proposed [5] [6]. The scheme proposed in [5] dynamically adapts the MAC layer retry limit to reduce performance degradations due to spurious triggering of TCP congestion control. The Eifel detection algorithm [7] allows a TCP sender to detect a posteriori whether it has entered into the loss recovery phase unnecessarily. This algorithm tries to nullify the effects of spurious timeouts. Tsaoussidis [8] et al. proposed an end to end proposal called TCP probing. In this scheme when a data segment is delayed or lost, the sender enters into a probe cycle instead of retransmitting and reducing the congestion window size. In case of persistent error, TCP decreases its congestion window and slow start threshold.

5. SIMULATION RESULTS AND DISCUSSION

The parameter settings for the simulation experiments are illustrated in Table 1. The simulation were done using Qualnet simulator [9].

Table 1. Simulation Parameters

Transmission Range	250 m
Routing Protocol	AODV
MAC protocol	IEEE 802.11
TCP Traffic type	FTP Generic
Packet Size	512 bytes
TCP Variant	TCP Lite

The performance metrics that are analyzed for evaluation include throughput, number of retransmissions, fast retransmissions and the time when the first packet is received. After extensive simulations the inferences from the simulations depict the main reasons for TCP performance degradation.

In this paper the interaction between TCP and IEEE 802.11 MAC protocols through simulations has been studied and the simulation results indicate that TCP does not perform well with 802.11 MAC in multi-hop wireless networks. Using the experimental setup described earlier, the simulation was performed and the throughput was calculated between the end nodes of the chain in a chain topology. Fig 1 depicts the node interference in chain topology.

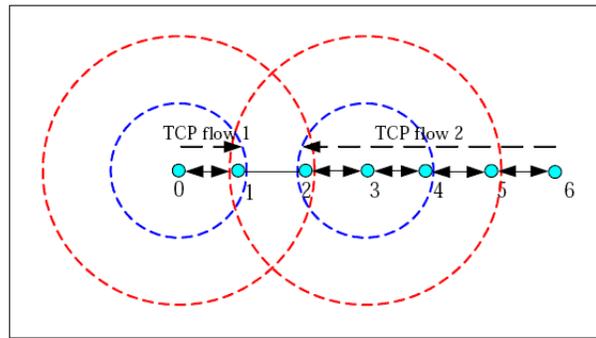


Fig. 1. Node Interference in Chain Topology

The outer lines indicate the interference range and the inner lines indicate the transmission range. The measured throughput is shown as a function of number of hops. Table 2 shows the time at which the first packet is received.

Table 2. Number of Hops and Time when First packet Received

Number of hops	First Packet received at (sec)
1	1.13
2	1.18
3	1.3
4	1.3
5	1.4
6	4.3

As seen in Table 2 the time taken for the first packet to reach the destination is stable until a specific number of hops and increases as the number of hops increase after a certain value. This is because the node is in the transmission range of the sender and hence the first packet reaches early. However when the destination node moves farther away from the source node, interference may occur in the transmission range causing the deviation in the time of first packet received. Also if the nodes are in the colliding transmission from the destination and the source an increase in the time of the first packet reaching may be reflected.

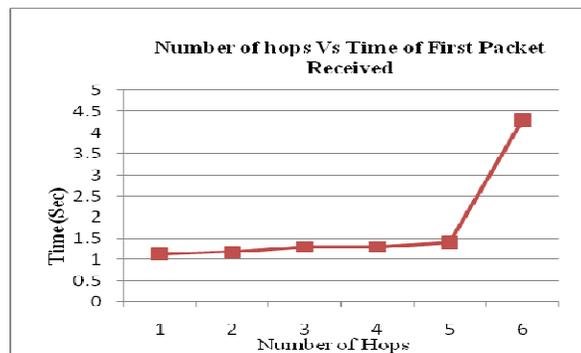


Fig. 2. Number of Hops Vs Time first packet received

Simulation results shown in Fig 3, plots the throughput of TCP connections with different number of nodes which changes the number of hops between source and destination.

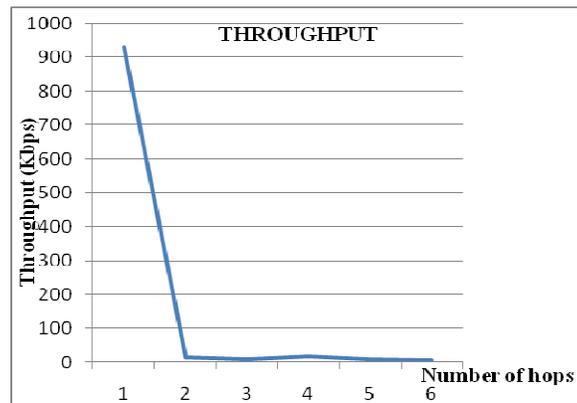


Fig. 3. Throughput Vs Number of hops

The results show that when the number of hops is less no link is used simultaneously due to the hidden terminal problem. When the number of hops grows there are links that are used simultaneously because of channel spatial reuse. This is the reason for the rapid decrease in the TCP throughput when the number of nodes increases from 1. However the throughput stabilizes when the number of hops becomes large. Simulations were done by varying the number of hops and number of flows. Table 3 shows the number of retransmissions that occur when the numbers of flows are varied.

Table 3. Number of flows and retransmissions

Number of Flows	Retransmissions
1	65
2	88
3	119
4	130
5	154
6	176

The number of retransmissions increases with the number of flows as shown in Fig. 4. The reason is due to excessive buffering at each node and congestions that arise due to this. The increase in the number of retransmissions may also be due to several reasons such as mobility of the nodes because of which route failure occurs and thereby the performance is degraded.

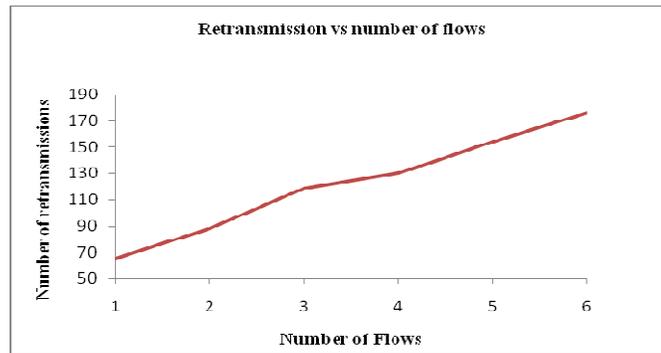


Fig. 4. Number of Flows Vs Retransmissions

The following table, Table 4 shows the number of data items sent using FTP/Generic and the number of retransmissions. The graph in Fig 5 shows that as the payload increases the number of retransmissions increase. It is to be noted from Fig 5 that number of retransmissions is more than the number of fast retransmissions. Data packets are retransmitted when packets are lost. Retransmissions occur when the retransmission timer expires. However Fast Retransmissions occur when more than three duplicate acknowledgments are received by the sender.

Table 4. Payload and Number of Retransmissions

Number of data items sent using FTP Generic	Number of Retransmissions
1000	110
2000	187
3000	273
4000	355
5000	416
6000	521
7000	606

This indicates that there is no congestion in the network however out of order delivery of packets exists. In wireless environments packets may be lost due to many reasons but TCP interprets every loss as loss due to congestion and invokes congestion control procedures. Simulations also show that the number of fast retransmissions also increase as the payload size is increased but more number of retransmissions due to RTO expiry occurs compared to the number of fast retransmissions.

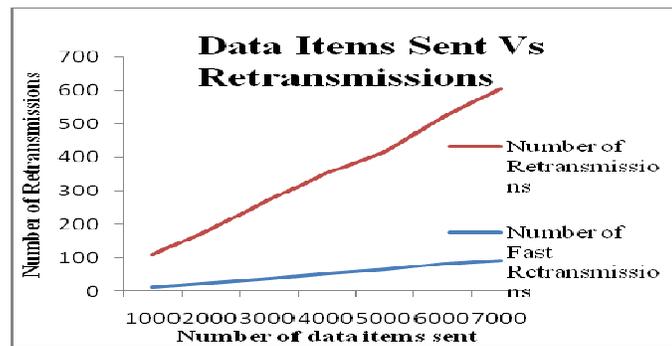


Fig. 5. Data packets sent Vs Number of Retransmissions.

6. CONCLUSION

This paper has examined the performance of TCP in mobile ad hoc networks based on few parameters. An investigation of discrepancy of the TCP performance over a stable path was provided. This study takes a step to analyze TCP performance in ad hoc wireless networks. The analysis is based on a chain topology containing N hops. Simulation results shown depicts that when the number of hops increases from one, the TCP throughput decreases first, and then stabilizes when the number of hops increases. The main reasons for degradation of performance of Transmission control protocol in mobile ad hoc network is imposed by the dynamic topology, multi-hopping, channel contention and high number of transmission errors. TCP also have the limitation to differentiate between transmission and congestion losses. Most of the time congestion control is activated without any need. TCP recovery process is also very slow. However this paper focuses on analysis based on few parameters that contribute to throughput. Future research prospects include the reasons and differentiation for the causes of packet loss. The same study can also be extended to other topologies such as cross, grid and random. It may be of particular interest to explore to what degree hidden terminal effects, or other MAC layer incompatibilities are responsible for the degradation in TCP throughput, compared to the degradation caused by route breakages.

REFERENCES

- [1] Postal, "Transmission Control Protocol," Request for comments, RFC 793, Protocol Specification, DARPA Internet Program, September 1981.
- [2] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," Request for Comments, RFC 2581, Network Working Group, Internet Engineering Task Force, April 1999.
- [3] S.Mascolo, C.Casetti, M.Gerla, S.S.Lee and M, Sanadid, TCP Westwood: Congestion control with faster recovery. Technical Report 200017,UCLA CSD 2000
- [4] J. Liu and S. Singh, "ATCP: TCP for Mobile ad hoc Networks," IEEE Journal in Selected Areas in Communication (JSAC), Vol 19, No.7, pp. 1300-15, July 2001.
- [5] S.Lohier, Y. G. Doudane and G. Pujolle, "Mac -Layer adaptation to improve TCP performance in 802.11e wireless networks "proc,IEEE International Conference on Wireless and Mobile Computing, Networking and Communication, 19-21 June 2006.
- [6] S.Fowler, M.Eberhard and K.Blow, "Implementing an adaptive TCP fairness while exploiting 802.11e over wireless networks, "International Journal of Pervasive Computing and Communications (IJGCC), Vol.5.no.3,pp 272-294,2009
- [7] R.Ludwig and M. Meyer."The Eifel detection algorithm for TCP ".RFC 3522, April 2003
- [8] Tsaoussidis, V. Badr, H. "TCP-probing: towards an error control schema with energy and throughput performance gains", The 8th IEEE Conference on Network Protocols", ICNP 2000, Osaka, Japan, November 2000.
- [9] Qualnet Simulator 5.0.