A review of TCP Reno, TCP Vegas and TFRC for mobile ad hoc networks

Gurmeet Singh[#1], Jaswinder Singh[*2]

[#] *Computer engineering & University college of engineering, Punjabi university, Patiala , India*

[1] *gurmeetsekhon73@gmail.com*

[*] *Assistant professor, Computer engineering & University college of engineering , Punjabi university, Patiala,*

[2] *jaswindersinghmtech@gmail.com*

***ABSTRACT*:** The most common transport protocol in a network is the Transmission Control Protocol. There are different variants of TCP like TCP, Tahoe, Reno, New Reno, Vegas, STCP and so on.TCP in its present form is not well suited for mobile adhoc networks (MANETs) where packet loss due to broken routes can hinder the working of TCP's congestion control mechanisms. In this paper we studied 2 types of Congestion Control techniques. One is Windows based Congestion Control and seocnd is Rate based Congestion Control .TCP Reno, TCP Vegas are windows based Congestion Control variants and TFRC works on Rate based flow congestion control.

1 .INTRODUCTION

Ad-hoc networks are self-organizing wireless networks, in which all end nodes act as routers. This network improves the efficiency, range of fixed or mobile internet access and enables totally with new applications. A Mobile Ad hoc Networks (MANET) consists of a set of mobile hosts within communication range and exchange the data among themselves without using any preexisting infrastructure. MANET nodes are typically distinguished by their limited power, processing and memory resources as well as high degree of mobility. In such networks, the wireless mobile nodes may dynamically enter the network and leave the network. Due to the limited transmission range of wireless network nodes, multiple hops are usually needed for a node to exchange information with any other node in the network. [1]

TCP is basically used for implemented congestion control these days. TCP do not use resource reservation on network for sending packet. It reacts only if any event occurs in network. TCP is a reliable protocol and provide reliable services for end to end entities. TCP congestion control mechanism is well suited for connection oriented network but for wireless networks it is not well suited. For congestion control in mobile Ad-Hoc networks different variants of TCP are used which we will discuss in next section.[2]

2. TCP PERFORMANCE IN MANET

TCP ensures reliable end-to-end message transmission over wired networks, but a number of researches have showed that TCP performance can be substantially degraded in MANET. The following are the different types of constraints influencing TCP performance in MANET.

2.1. Route Failure

In MANET, the major reason for the route failure is the mobility of the node and in case of route failure reestablishment of the route is instantly needed. A new route establishment may experience longer duration than the RTO of the sender. As a result, the congestion control mechanism will unnecessary invoke by the TCP sender.

2.2. Path Asymmetry Impact

Within MANET, the network topology is changed very frequently and arbitrarily that leads to the creation of an asymmetric path. Since TCP is highly dependent on time responsive feedback information, this path formation negatively influences the TCP performance. The packet is lost when the sender starts transmitting data in a burst when a number of ACKs are received together .Path asymmetry can be grouped into different forms such as loss rate asymmetry, bandwidth asymmetry and route asymmetry.

2.3. Network Partitioning

When a node departs from the network, a network partition takes place which results in an isolation of some parts of a mobile ad-hoc network. The fragmented portions are defined as partitions. Network partitioning is mainly caused due to the mobility or energy-constrained (limited battery power) operation of nodes and TCP considers network partitioning as one of the most imperative challenges in MANET. All transmitting packets are found to be dropped by the network if the source and the destination of a TCP connection lie in different parts of the network .In consequence of that, TCP sender will instantly invoke the congestion control algorithm.[7]

3. TCP Reno

TCP Reno is one of the TCP variant which is used for congestion control in mobile ad-hoc network. TCP Reno is based on the principle of TAHOE but it also use fast recovery so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost. It has four main parts.

a) Slow start

b) Congestion avoidance

c) Fast retransmit

d) Fast recovery

In TCP Reno slow start threshold value is use to determine whether slow start is used or congestion avoidance is used.

a) Slow Start

The slow start mechanism adds a new parameter that controls the rate at which packets are sent, congestion window denoted by cwnd.

1. Start with cwnd=1 (slow start).
2. On each successive acknowledgement value of cwnd is increased by 1.
   cwnd ⟵ cwnd+1
3. There is exponential growth of cwnd
   Each RTT: cwnd ⟵ 2*cwnd
4. Enter Congestion Avoidance when
   cwnd >= ssthresh

b) Congestion Avoidance

1. Starts when cwnd >= ssthresh
2. On each successful acknowledgement
   cwnd ⟵ cwnd+1/cwnd
3. There is a linear growth of cwnd
   Each RTT:
   cwnd ⟵ cwnd+1

c) Fast Retransmit

In Fast Retransmit mechanism when 3 duplicate acknowledgements are received, it is understood that there is packet loss. Hence even before the actual packet loss is detected, the packet is retransmitted.

1. Waiting for a timeout is too long
2. Retransmit data immediately after 3 duplicate acknowledgments without waiting for timeout
   Timeout :
   ssthresh ⟵ cwnd/2
   cwnd=1
3. Adjust threshold value
   Flightsize = min(cwnd, cwnd)
   ssthresh ⟵ max(flightsize/2,2)
4. Enter slow start cwnd=1

d) Fast Recovery

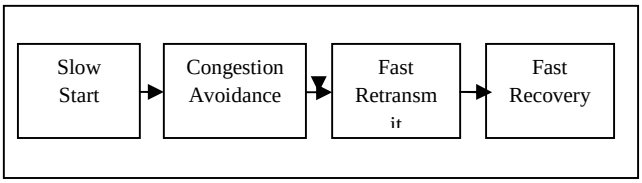
The motivation of fast recovery is to prevent the pipe from emptying after retransmission. The main idea behind fast recovery is each duplicate ACK represents a packet having left the pipe (successfully received).

1. Enter Fast recovery after 3 duplicate ACK.
2. Set ssthresh ⟵ max(flightsize/2,2)
3. Retransmit loss packet
4. Set cwnd ⟵ ssthresh+ndup(window inflation)
5. Wait till W = min(awnd, cwnd) is large enough; transmit new packet(s).
6. On non-dup ACK (1 RTT later), set
   cwnd ← ssthresh (window deflation)
   Enter CA

There are some flaws in TCP Reno.

1. Reno work very well in tcp when packet loss is small but when packet loss is high or there is multiple pack loss TCP Reno is not suited for that.
2. In tcp reno window size is small so 3 duplicate packets can't be received for fast retransmission so we can retransmit packet only when timeout is expire. because of this TCP Reno cannot detect multiple packet loss effectively. [3]

To remove these flaws we use another variant of TCP called TCP Vegas.



4. TCP Vegas

It is proposed by Brakmo and Peterson in 1995.it is a Congestion control algorithm which uses RTT time to measure the network situation. It basically Compare the expected efficiency and actual efficiency to decide whether increasing or decreasing the cwnd value.

Vegas use following three modified mechanism to increase delivery throughput and decrease packet loss.[4]

a. Modified Slow-Start Mechanism
b. New Congestion Avoidance Mechanism
c. New Retransmission Mechanism

a) Modified Slow Start Mechanism

- Limited Slow-Start Mechanism
- To be able to detect and avoid congestion during slow-start, Vegas reduce the increasing rate of cwnd.
- Cwnd is allowed exponential growth only every other RTT. (doubles the size of cwnd every 2 RTT time while there are no losses).
- In between, the cwnd stayed fixed so a valid comparison of the expected and actual rate can be made.
- When the actual rate falls below the expected rate by a certain amount - the γ threshold - Vegas changes from slow-start mode to linear increase/decrease mode.
- By limiting the maximum increase in the cwnd in a round-trip time, Limited Slow-Start can reduce the number of drops during slow-start, and improve the performance of TCP connections with large congestion windows.
- When Vegas detect there is queuing in network , the queue length exceed γ,and the actual rate falls below the expected rate, Vegas will change its state from slow-start mode to congestion avoidance mode, and set cwnd to 7/8 of current value.

b) Modified Congestion Avoidance

- TCP Vegas has another New Congestion Avoidance Mechanism to control the size of cwnd by observing the variation of RTT.
- When the sender receives an ACK, Vegas calculate the difference between the expect send rate and the actual send rate.
- Vegas defines two thresholds, α and β, and
  Diff = Expected send rate – Actual send rate
- Expected send rate= Window size / Base RTT
- Actual send rate= (the bytes transmitted between the time that the segment is sent and its ACK is received) / (the Segment's RTT)
- Base RTT is the minimum of all measured RTT, that is updated if the Diff < 0.

- When Diff < α, Vegas increase the cwnd linearly during next RTT, and when Diff > β, Vegas decrease the cwnd linearly during the next RTT.
- Vegas leaves the cwnd unchanged when α < Diff < β. The default value of (α, β) is (1,3).

c) New Retransmission Mechanism

Vegas improved the Fast retransmit mechanism in order to detect packet loss earlier and retransmit immediately.

- When a duplicate-ACK is received, Vegas checks if the RTT time, which is the difference of the current time and the timestamp recorded for the relevant segment, is greater than the timeout value. If it is, Vegas retransmit the segment immediately without 3 duplicate-ACKs
- When a non-duplicate ACK is received, if it is the first second one after a transmission, Vegas again checks if the time interval since the segment was sent is larger than the time out value. If it is, retransmit the segment.
- Vegas only decrease the cwnd if the retransmitted segment was previously sent after the last decrease.[5]

d) TFRC(TCP Friendly Rate Control)

TFRC is a congestion control mechanism designed for unicast flows operating in an Internet environment and competing with TCP traffic. TFRC is designed for applications that use a fixed packet size, and vary their sending rate in packets per second in response to congestion. TFRC should only be used when the application has a requirement for smooth throughput such as telephony or streaming media where a relatively smooth sending rate is of importance. TFRC is a receiver-based mechanism, with the calculation of the congestion control information (i.e., the loss event rate) in the data receiver rather in the data sender.

A new transport protocol was proposed by the Internet Engineering Task Force (IETF), in response to the problems of TCP , which is named as TCP-Friendly Rate Control protocol. To be friendly to TCP flows and at the same time maintain the smoothness of the changing rate to avoid severe performance degradation are the main objectives of TFRC.

1. Initial slow start:
   Initial rate：4 packets/RTT
   The rate Double every RTT
2. Linear increase / decrease
   No window size

In this Sending rate is regulated by Markov Model

3. The model is based on TCP Reno[6]

Table 1
Comparison Of TCP Reno and TCP Vegas

Parameters	TCP Reno	TCP Vegas
RTT measurement	Coarse grained timer is used	Fine-grained clock values are used
Retransmission decision	When receiving 3 duplicate ACK	When receiving a duplicate ACK check whether timeout is expire and if so then retransmit
Congestion window decrease	Decrease more than once during one RTT	Reduced only for first fast retransmission
Congestion detection	1.it is reactive protocol 2. Uses the loss of segments as a signal that there is congestion in the network.	1. it is proactive protocol 2. it tries to detect incipient congestion by comparing the measured throughput to its notion of expected throughput
Congestion window increase	doubles congestion window size every RTT	doubles congestion window size every other RTT (valid comparison of the expected and the actual rates)

6. CONCLUSION

TCP Reno is one of the TCP variant used for congestion control in mobile ad-hoc networks. but it is well suited only if there is single packet loss in a network. When there is multiple packet loss in a network then it is not working properly. TCP Vegas which is another variant of TCP is used for congestion control when there is multiple packet loss in network. TFRC is used for applications which need smooth output like internet telephony and multimedia streaming. It is implemented for those applications which use fixed packet size.

7. REFERENCES

[1] Saher S. Manaseer, "On backoff mechanisms for wireless mobile ad hoc networks," in Ph. D thesis, *The Faculty of Information and Mathematical Sciences at University of Glasgow, Scotland*, 2009, PP. 1-156.

[2] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva.. A performance comparison of multi-hop wireless ad hoc network routing protocols. In

[3]http://www.arl.wustl.edu/~jst/cse/770/talks/georg10-14-04.pdf

[4] M.Jehan Dr.G.Radhamani T.Kalakumari "VEGAS: Better Performance Than Other TCP Congestion Control Algorithms on MANETs" *International Journal of Computer Networks (IJCN)* Volume (3), Issue (2) , june 2011

[5] S. R. Biradar, Subir Kumar Sarkar , Puttama dappa C "A Comparison of the TCP Variants Performance over different Routing Protocols on Mobile Ad Hoc Networks" *(IJCSE) International Journal on Computer Science and Engineering* Vol. 02, No. 02, 2010, 340-344

[6] Floyd, S., M. Handley, J. Padhye and J. Widmer , 2008. TCP Friendly Rate Control (TFRC): Protocol specification. University College London

[7] Md Nazmul Islam Khan, Rashed Ahmed and Md. Tariq Aziz "A SURVEY OF TCP RENO, NEW RENO AND SACK OVER MOBILE AD-HOC NETWORK" *International Journal of Distributed and Parallel Systems (IJDPS)* Vol.3, No.1, January 2012