IJRCS - International Journal of Research in Computer Science ISSN: 2349-3828

TCP VARIANTS IN MANET- A SURVEY

Jaya Shree S[1] | Gopika G[2]

[1](Dept. of CSE, Assistant Professor, SNS College of Technology, Tamil Nadu,India,s.jayashree092@gmail.com)

[2](Dept. of CSE, Assistant Professor, SNS College of Technology, Tamil Nadu, India)

***Abstract*—*Mobile Ad-Hoc networks are the collection of independent mobile nodes with no fixed infrastructure. The high mobility of nodes is responsible for the congestion of the network and this congestion may occur due to various other reasons. TCP's robust behaviour is mainly due to its reactive behaviour on congestion and retransmissions are done to ensure reliability. Although many reasons may be there TCP, which is a reliable End-to-End transport layer protocol finds it difficult to distinguish between the losses and considers the congestion losses to be due to queue overflow or queue delay. The non congestion losses like mobility, power depletion of nodes and link failure are not considered by the algorithms. Hence there must be proper algorithms to detect the non-congestion and congestion losses. In this paper the variants of TCP are discussed these variants provide a method to control congestion losses in the network whether it may be due to congestion or non-congestion. These mechanisms provide methods for re-transmission, congestion control and pattern to avoid congestion.*

Keywords*—*TCP; Non-congestion losses; TCP Variant

1. INTRODUCTION

Mobile ad-hoc network (MANET) is a collection of independent nodes which are connected to each other without any infrastructure or base station [1]. Here, each and every node acts as a router and forwards the packet to the neighbour node. Router is a device in a network which forwards the data packets from one network to the other network. Thus routers can be used to connect different networks. Typically in networks Routers are generally fixed at the gateway of a network which leads to the formation of an infrastructure. Since MANET does not have any fixed infrastructure, there is no special device like routers for forwarding the packets. Each and every node in the MANET itself acts as a router. Thus these nodes apart from sending and receiving its packets it should also forward the packets of other nodes They are applied in various fields such as military applications, disaster management areas, remote places [11] where construction of base station is not possible and in places which require quick establishment of networks for communication. MANETS are the type of wireless networks in which every device is connected by the wireless medium. Since the connection is wireless lot of challenges are faced by MANET. Major challenges include mobility of nodes, congestion, link failure, battery depletion, frequent route changes, etc. mobility of nodes causes various problems such as link route failures, battery- depletion and packet droppings [2]. Congestion is caused when the network has the packets higher than it can handle the packets. Congestion is mainly caused due to queue formation at the receiver side due to slow processing of packets by the receiver, less buffer size, link failure, route changes, mobility of nodes and transmission time out. Allocating and maintaining the network resources among various users is considered a major problem in networks. The bandwidth of links and the queues on the routers and switches are the main resources that are shared in the network. Packets in the queues wait for transmission and when the same link is contended by too many packets there occurs the queue overflow and the packets will be dropped. If this event occurs persistently, then the network is said to be congested. Since there is no central router or infrastructure in MANET, the mobile nodes acts as a router themselves hence the congestion control mechanisms are router centric or node centric.

A. Congestion control

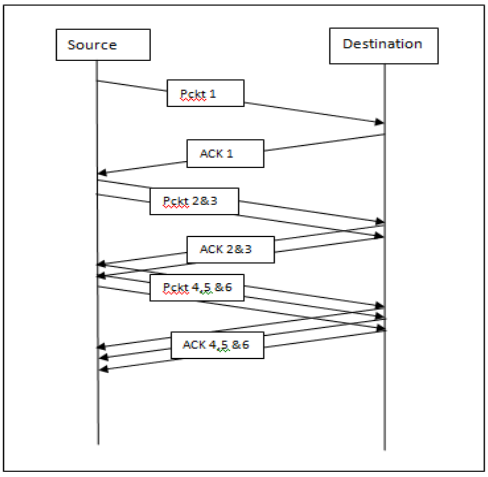

Fig:1 congestion control in TCP

TCP is reliable connection oriented protocol and the congestion avoidance scheme in TCP is very essential for the stability of the internet. TCP has the end-to-end delivery flow control and error control mechanisms. [3] TCP is originally designed for wired networks but when used in wireless networks it faces many problems one of the important issue is congestion. The TCP performance degrades when there is a multiple hop in the network and also due to the mobility and network partitioning. TCP does not treat route failure and congestion differently hence when a segment is lost, it considers it as a sign of congestion and invokes congestion control mechanism.

These congestion losses may be classified into two types congestion losses and non- congestion losses [7]. The congestion losses occur due to queue overflow or transmission time out. Non-congestion may occur due to mobility of nodes, link failure and battery depletion. But congestion is always considered to be caused due to congestion losses even though the cause is due to non-congestion loss. Hence these losses should be properly be differentiated to identify the cause of congestion.

In this paper, we are comparing the performances of different TCP variants namely TCP TAHOE, TCP RENO, TCP NEWRENO, TCP VEGAS. The robustness of TCP is due to its reactive behaviour in face of congestion and retransmission is done on the packet loss conditions. Each TCP variant exhibits different mechanisms on handling the congestion. These mechanisms are discussed here.

B. Additive Increase and Multiplicative Decrease of TCP:

[12] The Additive increase and multiplicative decrease is the congestion control algorithm in TCP. When there is a congestion in the network, AIMD combines linear growth of congestion window with an exponential reduction. Here, the transmission rate or window size is increased for usable bandwidth until there is a loss in the network. This policy increases the congestion window size by one as the maximum segment size every round trip time until there is a detection of loss in the network. When there is a loss, the window size is cut to half the size which is a multiplicative decrease.

2. ALGORITHMS FOR CONGESTION CONTROL IN MANET:

A. TCP Tahoe:

TAHOE forms the base for all the congestion control mechanisms in TCP. [9] i.e., all the mechanisms retain the basic principle of TAHOE in their transmission policy. Here, TCP implements this algorithm using the ACK which are the indication of packet taken by the wire or the node. The congestion window used here is to reflect the capacity of the network. TAHOE implements the algorithms called slow start algorithm and congestion avoidance. The slow start algorithm is used here because when the packet is induced into the network sudden burst of packets may result in network failure and the connection may not start over at all. In slow start algorithm, initially the size of congestion window is one. Upon the receiving of the ACK from the sender, the window increases by two. Thus, the window is increased exponentially. When packet loss occurs in network, TCP enters the congestion avoidance phase in which, the congestion window grows linearly from ‘n’ to ‘n+1’ only upon the receiving of n ACK. Additive increase and multiplicative decrease is used by congestion avoidance to avoid congestion [4]. The important drawback in TAHOE is that it takes a complete timeout to detect the packet loss and also the pipeline is completely emptied every time on the packet loss. This causes huge cost in high bandwidth delay product links.

B. TCP Reno

RENO retains the basic principles of TAHOE like the slow start and congestion avoidance. It also adds the principle of fast retransmission and fast recovery [2]. Unlike TAHOE it does not completely empty the whole pipe rather it checks for three duplicate acknowledgments when three DUPACKs are received, it takes this condition as the lost segment indication and retransmits the segments without waiting for a time out. And also it does not reduce the congestion window to 1 since it makes the pipe empty. This algorithm is known as fast retransmit. On the arrival of 3rd DUPACK, retransmission of lost segment takes place. The slow start threshold is set to half the current window size this saves the pipe from fully emptying and the flow is just reduced. The major problem faced by RENO is that it cannot detect the multiple packet losses.

C. TCP NEW RENO

A slight modification over RENO is NEW RENO in which NEW RENO can detect multiple packet losses [3]. The TCP stores the highest data sequence number of the packet which is sent when the 3rd DUPACK arrives which is termed ‘recover’. It also enters fast retransmit when multiple duplicate packets are received but does not exit from this mode unless it receives ACK higher than recover packet. When the ACK is lower than the recover packet, it starts retransmission. This ACK is known as partial ACK. Fast retransmission is done until the ACK with higher sequence number is received. The problem with NEW RENO is that it takes one RTT for detecting the packet loss. The segment loss is deduced only when we receive the first ACK.

D. TCP VEGAS

VEGAS use the new retransmission mechanism in which the RTT is measured for every segment which is based on the fine grained clock values. When an ACK or Duplicate ACK is received, VEGAS checks for the expiration of the time out period. For multiple losses, the congestion window is reduced only for the first fast retransmission. And for the congestion avoidance, it does not increase the window continually rather it observes the RTT of the segments that the connection has sent before. The congestion window is cut down when the observed RTTs become large which is the indication of congestion in the network. When the RTT become small VEGAS determines this as the sign of reliving from congestion and increases the congestion window. In the modified slow start phase, VEGAS differ from other algorithms. When a connection starts initially the bandwidth availability is not known and during the exponential increase of the window, it overshoots the bandwidth and thus congestion is induced. To overcome this, VEGAS exponentially increases its window only after every RTT and in between that it calculates the actual sending throughput to the expected this difference when exceeds the slow start and enters the congestion avoidance phase.

E. TCP SACK

SACK selective acknowledgement overcomes the problems faced by TCP Reno and NEW Reno [8]. Retaining the slow start and fast retransmission policy, it has the course-grained timeout to fall back, if the packet loss is not detected by the algorithm. In SACK, instead of acknowledging the packets in a cumulative way, it uses the selective acknowledgement. There is a block in each ACK which describes the acknowledged segments that are being acknowledged. Thus the sender knows about the segments that have been acknowledged and the segments which are all outstanding. During the fast recovery phase, a variable pipe is initialized which is an estimate of outstanding packets in the network and sets the CWND to half the current size. The variable pipe is reduced to 1 when ACK is received and when re-transmission is done, it increases by 1. It checks and sends the un-received segments when the pipe goes smaller than the CWD window. If no such segments are present, it sends the new packet. Thus many numbers of segments can be sent in one RTT. The major problem faced by SACK is that the implementation of selective acknowledgement is very difficult task.

REFERENCES

[1] Jitendra Padhye, Victor Firoiu, Donald F. Towsley, Fellow, IEEE, and James F. Kurose, Fellow, IEEE" Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation" IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 8, NO. 2, APRIL 2000.

[2] Dhananjay Bisen and Sanjeev Sharma" IMPROVE PERFORMANCE OF TCP NEW RENO OVER MOBILE AD-HOC NETWORK USING ABRA" International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 2, April 2011.

[3] 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.

[4] Veepin Kumar, Himanshu, Ajaydeep "A Comparative Analysis of TCP, Reno, New Reno, Vegas in MANET", International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 5 Issue 4 July 2015.

[5] Digvijaysinh B Kumpavat, Prof. Paras S Gosai, Prof. Vyomal N Pandya "Comparison of TCP congestion control mechanisms Tahoe, Newreno and Vegas", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 5, Issue 6 (Mar. - Apr. 2013), PP 74-79.

[6] Geetika Maheshwari, Mahesh Gour, Umesh Kumar Chourasia "A Survey on Congestion Control in MANET" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 998-1001.

[7] Bhagyashree S kayarkar, V.S.Deshmukh "A Survey of Congestion Control in Proactive Source Routing Protocol in Mobile Ad Hoc Networks" COMPUSOFT, An international journal of advanced computer technology, 3 (12), December-2014 (Volume-III, Issue-XII).

[8] Geetika Maheshwari et al, "A Survey on Congestion Control in MANET" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 998-1001.

[9] V.Jacobson. "Congestion Avoidance and Control". SIGCOMM Symposium no Cummunication Architecture and protocols.

[10] L.S.Brakmo, L.L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet", IEEE Journal on Selected Areas in Communication, vol. 13[1995],(1465-1490).

[11] 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.

[12] V. Thilagavathe Cross layer based congestion control Technique for reliable and energy aware Routing in MANET- Ph.D thesis.