ISSN:2231-0711
www.ijcset.net

A Proposed Modified Approach to Corruption Control for TCP in Mobile Ad-Hoc Networks

Poonam Tomar[#1], Shweta Yadav[*2]

[#]*Department of Information Technology, Mahakal Institute of Technology*
Ujjain, M.P. India

[1]poonam.bhumi@gmail.com
[3]shweta_yad@yahoo.co.in

***Abstract*— Among the two transport layer protocols, TCP is the reliable protocol that performs well in traditional networks where the main reason for packet loss is congestion. Wireless networks also suffer from some losses due to bit errors, hidden terminals etc. In response to all these losses TCP invokes congestion control algorithms. In case of packet losses due to corruption also TCP performs similarly. This is due to its inability to differentiate between congestion and Corruption. Here in this paper we will discuss all these issues and propose a mechanism that takes congestion and corruption conditions and works uniquely in each condition, so that unnecessary retransmissions are avoided. This will be done by varying the window size by calculating a new window.**

***Keywords*— Congestion, Corruption, TCP, Mobile Ad- Hoc network.**

I. INTRODUCTION

TCP is a reliable end to end protocol because it provides a reliable data packet delivery over the unreliable links. TCP provides the applications with reliable byte oriented delivery of data on the top of Internet Protocol (IP). When a TCP connection is established between sending and receiving processes, the sender writes bytes or characters into the connection and receiver reads them from the connection. Each byte of data is assigned a unique sequence number. The time that a packet takes to travel from source to destination and back from destination to source is known as RTT. On the basis of this RTT and its variation the retransmission timer is set up. Acknowledgements provide a reliable data delivery [1].

In Internet, one of the most important responsibilities of TCP is congestion control. TCP's congestion control mechanism works to protect the network against the incoming traffic that exceeds its capacity [2]. At the beginning the number of segments to be sent is based on the initial window which is of one or two segments initially [3].

TCP always acknowledges the completely received data when a new segment arrives. Whenever any out of order segment arrives the last acknowledgement is sent again. TCP uses a timeout mechanism that depends on the measured round trip time of the connection. If this retransmission timeout (RTO) expires without an acknowledgement from the receiver, TCP considers this a case of congestion and reduces the window size to one and sends the unacknowledged segment again. If until the next retransmission attempt no acknowledgement arrives, the timeout is doubled thus this timeout grows exponentially.

After the timeout a mechanism called slow start begins. In the slow start phase the transmission rate is increased exponentially. This is done to estimate the available network capacity which is provided by the congestion window. When an ACK is received the congestion window is increased by one segment, thus the sender transmit the number of ACK+1. This doubles the congestion window per RTT. Whenever the congestion window reaches the slow start threshold or a segment loss is detected, slow start ends and when this slow start threshold (ssthresh) is exceeded, the sender enters in the congestion avoidance phase. When the segment loss is detected , it is considered that there is congestion in network and the load on the network is decreased by setting the slow start threshold to half of the current flight size. In the congestion avoidance phase, the congestion window roughly increases by one segment per RTT.

The next phase is the Fast retransmit and Fast recovery phase. This was an enhancement to the TCP congestion control that previously only included slow start and congestion avoidance. This phase was to avoid waiting for a retransmit timeout every time a segment is lost. Previously only one Duplicate Acknowledgment (DUPACK) was sent as an indication of out of order segments, but since network can reorder or duplicate the packets only a single DUPACK is not sufficient to consider congestion. Therefore to consider the condition of congestion three duplicate acknowledgments are chosen. When three DUPACK's are received the fast retransmit algorithm is invoked. This time the DUPACK segment is considered to be lost and is retransmitted and the congestion window is halved. The purpose of fast recovery is to control the transmission of new data until a non duplicate acknowledgment is received.

II. CONGESTION & CORRUPTION: TCP'S INABILITY TO DISTINGUISH BETWEEN THEM

There are several resources in a network which are shared by various senders for communication & data transfer purpose but in such an environment where multiple senders compete for link bandwidth it becomes difficult to manage the data rate used by each sender to prevent network from overloading. Congestion occurs in a network

when number of packets sent in the network is greater than the capacity of the network [4]. Packets that arrive at a router and cannot be forwarded are dropped. Along with them are the retransmissions of those packets thus even more packets are sent in the network [5]. Thus this congestion problem can affect severely on network throughput and even collapse the network.

Wireless networks have high bit error rates due to multipath fading, interference, signal attenuation and fading, mobility of wireless devices etc that may lead to packet loss. So there is not a single reason of packet loss in wireless mobile networks. Thus congestion is one reason and packet loss due to above mentioned reasons is another reason. But TCP always reduces the transmission rate whenever any packet loss is detected, the reason whichever may be is not known.

III. PROPOSED SCHEME

In this paper we are proposing a new modified approach of TCP congestion and corruption control which is going to work differently in case of congestion and corruption. In wireless networks the lost packets are caused generally by high bit error rate [6] but not congestion. And TCP treats them similar. Several improved TCP congestion control schemes are presented like I-TCP [7], MTCP [8], Freeze TCP [9] etc.

The scheme that we are presenting here will distinguish between the lost packets due to congestion and lost packets due to corruption. This is done by making changes to the congestion avoidance algorithm of TCP presented as:

if (Receive ACKs || (Receive Explicit Corruption Loss Notification && Corruption Loss Rate Pe>Pemin))

{

if (cwnd > ssthresh)

cwnd = 1+1/cwnd

else

cwnd++;

m++;

if (Receive Explicit Corruption Loss Notification)

n--;

Pe=α* Pe + (1- α)* (n/m);

dewnd=cwnd* Pe;

cwnd=cwnd – dewnd;

$T_d = T_s - T_a$

If (T_d > 10%)

cwnd=cwnd/2

Fast Retansmission and Fast Recovery Algorithm

if (Congestion || Heavy Corruption)

{

if (Receive Same ACK 3 Times || Retransmission Timer Overtime) /* Congestion */

{

Ssthresh = max(flightsize/2 , 2*SMSS); // Flightsize are those data which have not acknowledged

if (Retransmission Timer Overtime)

{cwnd = 1; Exit and call slow-start;}

else /* Receive Same ACK 3 Time */

cwnd = ssthresh;

}

else if (Receive Explicit Corruption Loss Notification && Corruption Loss Rate Pe>Pemin)

m++;

if (Receive Explicit Loss Corruption Notification)

n--;

Pe= α * Pe + (1- α)* (n/m);

dewnd=cwnd* Pe;

cwnd=cwnd – dewnd;

else if (T_d > 10%)

$T_d = T_s - T_a$

If (T_d > 10%)

cwnd=cwnd/2

A. Packet Loss Rate due to Corruption

It is the rate of packet loss that occur due to corruption. And when the rate of packet loss increases we will have to decrease the packet sending rate to prevent unnecessary retransmissions. So a threshold is defined that describes a level which if exceeded causes the high corruption loss rate Pe[10]. And Pe is defined as Pe= m/n where m is the number of packets that a TCP sender sends and m packets are discarded because of bit errors caused by wireless link corruption. And since the corruption loss rate depends on bit error rate (BER) of wireless link and length of data frame, Pe= 1- (1-BER) ^{Length}. We will compare this corruption loss rate Pe with the lower limit Pemin, this minimum value depends on several factors like kind of application, length of frame, bit error rate of wireless link layer etc. A minimum value of Pe is chosen to be Pemin =0.4. If on comparing Pe is found higher than the lower limit Pemin, the packet sending rate will be decreased [11].

B. In case of Congestion

In case of normal TCP the size of the congestion window is kept normal. The algorithm works normally in case of congestion, congestion avoidance algorithm is invoked. And after the congestion avoidance phase Fast recovery and Fast retransmit are performed.

C. In case of Corruption

Upon receiving the Explicit Corruption loss notification and corruption loss rate exceeds the minimum value, the modified avoidance algorithm works. A new value of corruption loss rate is determined so that we can decrease the packet sending rate. The value by which we will decrease the current congestion window depends on the corruption loss rate and its old values. After determining this factor we will generate a new congestion window. After applying this mechanism we will determine the drop rate of the packets through the number of sent packets and packets acknowledged. If the drop rate calculated exceeds a certain value then it means that still there is need to decrease the congestion window and therefore we will half the new congestion window determined in previous step.

IV. CONCLUSIONS

The modified mechanism that is proposed in this paper will be implemented in a Mobile Ad-Hoc network in Qualnet 5.0. It considers that packet loss in a wireless network is not only due to congestion but also due to corruption. And instead of reacting in a similar way to both congestion and corruption, the algorithm will react differently in both cases. Further the calculation of drop rate is done that is based on total packets sent and total packets received.

Performance of this modified mechanism will be measured in different environments and our main work will be to compare it with the existing TCP variants and generate the results and hope it will come out to be a high potential in near future.

ACKNOWLEDGMENT

I am deeply thankful to my HOD Prof Shweta Yadav, Mahakal Institute of Technology for her guidance, support, stimulating suggestions and encouragement with the help of which in near future I will be able to complete my research work.

REFERENCES

[1] R. Brader, *Requirements for internet hosts- Communication layers. IETF KFC 1122,* October 1989.

[2] V. Jacobson. *Congestion Avoidance and control*. In proceedings of ACM SIGCOM'88, pages 314-329, August 1988.

[3] M. Allman. *A web Server's view of the transport layer.* ACM computer communication Review, 30(5), October 2000

[4] Behrouz A. Fourozen " *TCP/IP Protocol Suite*", 2006.

[5] Christian Lochert, Bjorn Scheuemann, Martin Mauve. *A survey on congestion control for mobile Ad-Hoc networks*, Wiley Wireless Communication and mobile computing. 7(5), pp. 655-676, June 2007.

[6] Bit Error Rate: *Fundamental Concepts and measurement issues, High frequency Electronics,* summit Technical Media/LLC, January 2003.

[7] Bakre A., Badrinath B, *I-TCP, Indirect TCP for mobile hosts*, In proceedings of the IEEE ICDCS'95, Vancouver, CA, 1995, pp. 136-143.

[8] Brown K., Singh S. *M-TCP: TCP for mobile cellular networks*. In ACM computer communication review, 19997, 27(5) : pp. 19-43.

[9] Goff T., Moronski J. *Freez TCP: A true end to end TCP enhancement mechanism for mobile environments*, In proceedings of IEEE INFOCOM, Tel-Aiv, Israel, 2000 , pp. 1537-1545.

[10] Xu Chang-Biao, Long Ke-Ping, Yang Shi-Zhang, *Corruption based TCP rate adjustment in wireless networks* In: Chinese Journal of Computers, 2002, 25(4): pp. 438-444 .

[11] Fu Lin, DengYiZhang, WenbinHu : *An Improved TCP congestion Control Mechanism based on Double Windows for Wireless Network,* ISWPC 2008.