International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 10, Issue 12 (December 2014), PP.67-71

A Survey of Different Approaches for Differentiating Bit Error and Congestion Error

Hina Asari[1], Rajendra Patel[2], Hitesh Ghadiya[3]

Abstract:- TCP provides reliable wireless communication. The packet loss occurs in wireless network during the data transmission and these losses are always classified as congestion losses. While Packet is also lost due to random bit error. But traditional TCP always consider as packet is lost due to congestion and reduce it congestion window. Thus, TCP gives poor performance in wireless link. Many TCP variants have been proposed for congestion control but they cannot distinguish error either due to congestion or due to bit error thus it reduces congestion window every time but when there is a bit error then no need to reduce the transmission rate. In this survey the general approaches taken for differentiating congestion or bit error has been discussed.

Keywords:- Congestion Control, Random Bit error, TCP Variants, Throughput, Congestion Window

I. INTRODUCTION

Many Internet applications and their protocols, e.g., HTTP and FTP, use TCP as their transport layer. Transmission control protocol (TCP) is reliable and connection oriented protocol. The network is not perfect and a small percentage of packets are lost during transmission, which is either due to network error or congestion error. TCP ensures reliability by starting a timer whenever it sends a segment. If it doesn't receive an acknowledgement from the receiver within the 'time-out' interval then it retransmits the segment. The congestion control algorithms of TCP are very essential for the reliability of data transmission as well as stability of the Internet.

Now a day's internet user are gradually increases so that internet congestion is much possible, and is one of the key issues in network. Congestion occurs when the number of received packets at node is more than its output capacity. TCP Tahoe is the first TCP variant which includes the first congestion control algorithm. Jacobson and Karle's developed this congestion control algorithm in 1986. Then after, many enhancements and modifications are introduced on Tahoe, and leads to design and development of new TCP variants with different congestion window algorithms. RFC 793 standardized the first TCP version with its basic configuration based on a scheme of window-based flow control. TCP Tahoe performs the second generation of TCP versions, which includes two techniques, congestion avoidance and fast transmission. Reno represents the third version of the first developed series, and it's standardized in RFC 2011, where the congestion control mechanisms are further extensive by fast recovery algorithm. Consequently, researchers have focused much in improving the performance of TCP - Reno and Vegas.

The main reason for TCP degradation is its incapability to determine the reason for three duplicate acknowledgement and retransmission timeout. Retransmission timeout caused by non congestion event is the main problem.

II. TRANSMISSION CONTROL PROTOCOL (TCP)

TCP [1, 2, 3] is a reliable, connection-oriented, end-to-end, error free in-order protocol. TCP connection is a virtual circuit between two computers, conceptually very much like a telephone connection but with reliable data transmission between them. A sending host divides the data stream into segments. Each segment is labelled with an explicit sequence number to guarantee ordering and reliability. When a host receives in sequence the segments, it sends a cumulative acknowledgment (ACK) in return, notifying the sender that all of the data preceding that segment's sequence number has been received. If an out-of sequence segment is received, the receiver sends an acknowledgement indicating the sequence number of the segment that was expected. If data is not acknowledged for a period of time, the sender will timeout and retransmit the unacknowledged segments.

In wireless network has some transmission errors are present that is generated by noise, interference, distortion or bit synchronization errors, but all TCP variants like TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas cannot able to distinguish both the error. But all these variants of the TCP and original TCP are still unable to sense the cause of packet loss. Hence all loss is treated as congestion loss, not consider as bit error and hence reduce window size and data flow. So that, congestion control algorithms perform poorly in wireless environment.

The TCP performs better in wire network but in wireless network; degrade the performance of the TCP. Therefore, first is need to distinguishing congestion loss from random loss in wireless link. Second is to reaction according to error. If the packet is lost due to congestion then retransmit that lost packet and set cwnd to half of the current cwnd. If the packet lost due to random error then its need to retransmit that packet but no need to reduce the cwnd.

III. RELATED WORK

There are several ways to differentiate congestion error and bit error or transmission error to improve the performance of TCP over wireless networks.

The first approach is based on ECN (Explicit Congestion Notification) referred from [14] with active queue management is used to control congestion in wired networks. And also suggests that ECN can be used to distinguish congestion loss from wireless loss by diagnosing a loss.

TCP NRT [4] uses modified ECN mechanism to differentiate congestion retransmission time out from non- congestion RTO. ECN is an extension to the random early detection (RED) algorithm for dropping packets. In modified ECN, the router is configured with 2 parameters: minimum threshold (min^{th}) and maximum threshold (max^{th}). When a packet arrives at the router, ECN calculates the average queue length (AQL) and if it is below min^{th}, the router will not mark the packets. If the AQL exceeds max^{th} then TCP router marks the packet. As a result there is no need to mark and reduce the size of cwnd when the AQL is between min^{th} and max^{th}. Whenever sender detects RTO, the sender checks whether the last received ACK is RTO is marked or not, If it is marked then the sender considers that there is congestion in the network and sender immediately retransmits the lost packet and reduces the current cwnd. Otherwise the sender assumes that the RTO is due to some other non congestion either random loss or packet reordering. If the detected RTO is due to random packet loss, the sender retransmits the lost segment without changing current cwnd.

In [6] , Explicit Congestion Notification (ECN) for congestion control mechanism and propose a solution to distinguish wireless and congestion losses. ECN avoids congestion losses by making use of early congestion warnings. By avoiding congestion losses it gives performance benefits. In this way it avoids the end-to-end retransmissions, as well as timeouts caused by multiple losses in a window.

Current ECN uses two bits in the IP header [14]. This scheme is not efficient, because it uses two bits indicate three states: not ECT (ECN Capable Transport); ECT and congested; and ECT and not congested. Since two bits can represent four states, a more efficient scheme is possible. Based on that proposed new scheme has been presented in [14], which also uses three levels of congestion with ECN signals to detect congestion losses. The ECN field consists of two bits and the ECN-Echo field consists of two bits. The ECN field is set by the source and changed by the intermediate routers; the ECN-Echo field is set by the receiver. The proposed bit patterns are summarized in Table 1.

With these bit patterns, network routers can indicate three levels of congestion with ECN signals: no congestion, mild congestion and severe congestion. The receiver can then faithfully reflect the ECN signals in the ECN-Echo field. Use of this feedback is left to the sender TCP.

Table 1 Proposed ECN and ECN Echo Field

ECN	ECN Field Meaning	ECN Echo	ECN-Echo Field Meaning
00	Not ECN capable	00	Reserved for other use
01	ECN capable and no congestion	01	Echo of no congestion
10	ECN capable and mild congestion	10	Echo of mild congestion
11	ECN capable and severe congestion	11	Echo of severe congestion

The routers set congestion signals based on two thresholds, one for mild congestion and another for severe congestion. A router marks the packet according to its current congestion status, but if the packet already carries a higher level mark, it will not change the mark.

In [8], **TCP_Reno** cooperates with the router configured with explicit congestion notification (ECN), it is capable of distinguishing the wireless packet losses from the congestion packet losses, and reacting accordingly. This is done by observing the Congestion Experienced (CE) bit in the duplicate ACKs received if the TCP is ECN-capable. When the packet loss occurred due to bit error then the sender estimates the packet loss rate. Suppose queue length has exceeded the minimum threshold value of the average queue length. If this occurs and the network is ECN-capable, it will set the CE bit in the TCP header of the packet with some probability. The receiver also sends ACK to the sender with its CE bit set. If the sender receives three duplicate ACK with its CE bit set then the sender can assume that a packet loss has occurred due to congestion in the network and it can start its congestion avoidance algorithm to take care of it. When the buffer actually overflows and a packet is dropped then the receiver sends duplicate ACK and the sender can conclude that there is congestion in the network. Suppose the queue length of the buffer is below the minimum threshold value, then CE bit in the TCP header will not be set by the node and as a result the ACKs coming from the receiver will also not have their CE bit set

In Second approach is discuss in [5] and [9] , TCP NCE [5] differentiates congestion from non congestion losses by computing the queue length of the bottleneck link. If the queue length is greater than a threshold value then, congestion is reported. Otherwise the packet loss is because of some non congestion event. The formula to compute queue length (Ql) is,

$$Ql = B\,(RTT_{now} - RTT_{min})$$

Where, RTT_{now} is the current round-trip time when the sender receives an ACK.
RTT_{min} is the observed minimum RTT by the TCP sender
B is the bandwidth of the link.

Here, RTT is measure using the Time stamp option field[13]. For detecting the non-congestion events at the time of receiving the three dupacks, than sender check the current queue length which is greater than a threshold value. If Ql is greater than a threshold value, the TCP sender confirms that the dupacks is due to network congestion and router uses drop-tail queuing policy and modified router queue management scheme is as shown in table 2.

Table 2 Router Queue Management Scheme

Buffer Size	Load	Status
less than 30%	Light	not congested
less than 90% and greater than 30%	medium	not congested (easy to become congested after sometime)
greater than 90%	heavy	under congestion (easily overflow)

When the sender receives three dupacks and the current queue length is less than the threshold value, then the sender assumes that these dupacks are the sign of non-congestion events.

In [9] TCP CERL (Congestion Control Enhancement for Random Loss), TCP CERL modifies sender-side of TCP. It is utilize the RTT measurements made throughout the duration of the connection to estimate the queue length of the link and after that estimates the congestion status. By distinguishing random losses from congestion losses based on a dynamically set threshold value, TCP CERL successfully distinguish performance degradation issue of TCP that is random loss. Unlike other TCP variants, TCP CERL doesn’t reduce the congestion window and slow start threshold when random loss is detected. Assume that TCP connection that performs first-in-first-out drop tail queuing. When packet is received by router then it is first measure queue length. Queue length l can be calculated using the following equation:

$$l = (RTT - T)\,B$$

Where, T to be the smallest RTT observed by the TCP sender and l is updated with the most recent RTT measurement every time a new RTT measurement is received. To distinguish random losses from congestive losses, the queue length l measured by above equation is used to estimate the congestion status of the link. Specifically, set a dynamic queue length threshold N:

$$N = A \times l_{max}$$

Where, l_{max} = largest value of l observed by the sender
A = constant between 0 and 1

If l < N, when a segment loss is detected by three duplicate acknowledgments, It assume the loss has been random rather than congestive. Otherwise, CERL will assume the loss has been caused by congestion.

Third approach is **TCP K-Reno** [7], when there is less congestion in the network but the probability of random bit error is high, some segments will be corrupted and rejected by the receiver. If the next segments in sequence do not suffer from bit error then they will reach the receiver and be regarded as out of order segments then receiver generat duplicate acknowledgements. The continuous flow of acknowledgements from the receiver will prevent timeouts from being occurred at the sender. Therefore, the number of 3-dupacks experienced by the sender will be much higher than the number of timeouts. This phenomenon can be used during a timeout or a 3-dupack event to decide whether there is real congestion in the network, or the segment loss occurred due to random bit error. It keeps count of the number of timeouts and the number of 3-dupacks. Whenever the sender experiences a timeout or 3-dupack event, TCP K-Reno computes the ratio of the number of timeouts to the number of 3-dupacks. If the ratio is very small (in between 0.01 to 0.2) that means this event has been caused by a bit error event, not by the congestion. If the ratio is high (e.g. greater than 0.5) then the event is more likely the result of segment drops at intermediate routers due to congestion.

In Fourth approach [10], it use 1 bit flag of TCP Header for strong indication of congestion or bit error. If F = 0 then congestion and F = 1 then bit error. This algorithm states if any segment loss due to any reasons during communication, receiver has been received out of order sequence numbers. When receiver receives out of order sequence number, it will start observing timing of next successive segments and continuous retransmits dupack. If next successive segments suffer from delay, then receiver assume that congestion occurs in the medium and segment loss due to congestion. If next successive segments does not suffer from delay and received continuously without delay, then receiver assume that congestion not occur in medium. So set FLAG = 0 in the 3 rd dupack (duplicate acknowledgment) otherwise set FLAG = 1.

If F = 0 in 3 rd dupack, Sender assumes that packet loss due to congestion and set cwnd to half & enter into fast retransmit phase of TCP. If F = 1 in 3 rd dupack, Sender assumes that packet loss due to bit error and not reduce cwnd & enter into congestion avoidance phase of TCP.

From study this different approaches we can classified as shown in below table 3.

Table 3 Different Detection Approach

Approach	Detection approach	Different TCP Approaches	Modification	Indication of error
First	ECN	TCP NRT [4]	Using ECN capable router	If Packet is marked then it is congestion otherwise random bit error
		Modified TCP [6]	ECN is modified with Three level ECN	
		TCP_Reno [8]	Using ECN capable router	
Second	Measure Queue length(l)	TCP NCE [5]	Define threshold value based on buffer size (90%)	If buffer size is grater than 90% then received 3 dupack is due to congestion
		TCP CERL [9]	Set dynamic queue length threshold (N)	If l < N then random bit error Otherwise congestion
Third	Based on no. of timeout and no. of 3 dupack	TCP K-Reno [7]	Count the number of timeouts and the number of 3-dupacks	Ratio of no. Of timeout to no. Of 3 dupack is small then 3 dupack is due to bit error
Fourth	Observing timing of next successive	Modified TCPW [10]	Measure delay of subsequent packet and set on reserved bit in header	Successive segments suffer from delay then set FLAG = 0 in 3rd dupack Otherwise it set FLAG = 1 in 3rd dupack

IV. CONCLUSIONS

Wireless networks are not always perfect due to that packets are lost and these losses are always classified as congestion loss by sender, causing reduced throughput. While Packet losses are mainly due to congestion and random bit error which is vary negligible. Traditional TCP does not differentiate losses either congestion or bit error. If packet loss is due to bit error even though it reduce congestion window and reducing its performance. So unfortunately it reduces congestion window in both cases and degrade the performance. So, there are several different approaches for differentiating congestion and bit error. ECN is mostly used for detection to differentiate congestion and non congestion error.

REFERENCES

[1] W. Boulevard, and A. Way, "Transmission Control Protocol", RFC 793, September 1981
[2] V. Jacobson and M. J. Karels, "Congestion avoidance and control", In ACM Computer Communication Review; Proceedings of the Sigcomm'88 Symposium, volume 18, pages 314–329, Stanford, CA, USA, August 1988.
[3] Hanaa A. Torkey, Gamal M. Attiya and I. Z. Morsi, "Performance Evaluation of End-to-End Congestion Control Protocols", Menoufia journal of Electronic Engineering Research (MJEER), Vol. 18, no. 2, pp. 99- 118, July 2008.
[4] Prasanthi Sreekumari and Meejeong Lee, "TCP NRT: a new TCP algorithm for differentiating non-congestion retransmission timeouts over multihop wireless networks", EURASIP Journal on Wireless Communications and Networking, 2013
[5] Prasanthi Sreekumari and Sang-Hwa Chung, **"**TCP NCE: A unified solution for non-congestion events to improve the performance of TCP over wireless networks **",** *EURASIP Journal on Wireless Communications and Networking*, 2011
[6] Mohammad Usman Ali Khan, Shahid Khan, Yasir Ali Rasheed, "Optimization of tcp over wireless networks", International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 11 No: 05, October 2011
[7] Ahmed Khurshid, Md. Humayun Kabir, Md. Anindya Tahsin Prodhan, "An Improve d TCP Congestion Control Algorithm for Wireless Networks", IEEE ,2011
[8] Hu Han, "Performance Improvement of TCP_Reno Based on Monitoring the Wireless Packet Loss Rate", IEEE 3rd International Conference on Communication Software and Networks (ICCSN), 2011
[9] Hosam El-Ocla, "TCP CERL: congestion control enhancement over wireless networks", Conference on wireless network Springer, 2010
[10] B.Chellaprabha, J.Priyadharsini, "RARELD for Loss Differentiation and Reduction in Wireless Network", International Journal of Scientific and Research Publications, Volume 3, Issue 6, ISSN 2250-3153, June 2013
[11] Qian Wang, Dongfeng Yuan, Senior, "An Improved TCP Congestion Control Mechanism with Adaptive Congestion Window", in International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Vol.4, July-2010
[12] RFC: 2001, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", January 1997
[13] RFC: 1323, "TCP Extension for High Performance", 1992
[14] RFC: 2481, "A Proposal to add Explicit Congestion Notification (ECN) to IP",1999