PACKET LOSS CLASSIFICATION FOR IMPROVING TCP PERFORMANCE IN WIRELESS NETWORKS

[1]Balasundaram Ananthakrishnan, [2]Mrs. S.P.Valli
[1]Post Graduate Student, [2]Assistant Professor (S.G)
B.S.Abdur Rahman University
Email: [1]balasundaram2682@gmail.com,[2]vallisp@bsauniv.ac.in

Abstract— TCP has become one of the most preferred communication protocols that is used across various networks. But a major drawback observed in TCP is its inability to classify and distinguish congestion losses from non-congestion losses.

Initially, TCP has been designed only for wired networks. But when TCP was used for wireless networks, the significance for packet loss classification became even more prominent. This paper discusses about a technique for classifying wireless packet losses using Explicit Congestion Notification (ECN) and Random Early Detection (RED). The performance improvement in TCP in terms of throughput has been studied and simulated using Qualnet 5.0 simulator.

Index Terms— **TCP Loss Classification, ECN, RED, Wireless Networks.**

I. INTRODUCTION

Transmission Control Protocol (TCP) has been used mostly for transporting data packets that require guaranteed delivery. This is the most commonly used communication protocol and has significantly been used over internet communication. During the course of packet message transmission, the packet may encounter different kinds of losses as shown in Fig 1. In wireless networks, the loss may be due to a link failure, signal loss, interference between two adjacent nodes leading to a packet loss or the loss may even be congestion oriented.

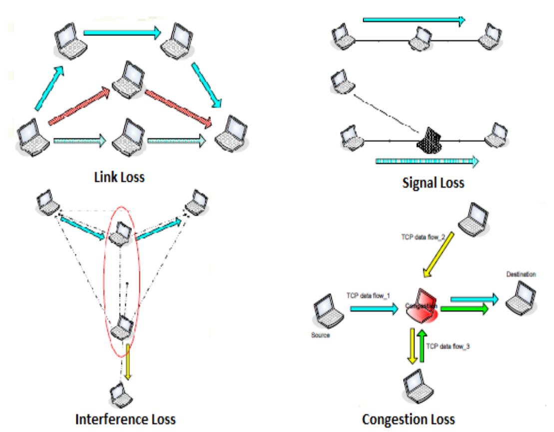

Fig 1: Types of Losses

But TCP does not have a defined mechanism to distinguish congestion losses from other kinds of losses. TCP by default assumes all losses as congestion oriented. As a result, the congestion window size gets adjusted every time a loss occurs, irrespective of whether it is a congestion related loss or not. When the congestion window size is reduced, the transmission rate is also reduced proportionately. As a result, the performance of TCP degrades considerably and yields a reduced throughput. Especially, in the case of wireless networks, the problem is compounded due to the distributed nature of the network and mobility of modes. Hence the need to classify congestion and non-congestion loss is more significant in wireless and adhoc networks. The objective of this paper is to discuss and propose a technique to distinguish congestion loss from other types of losses.

II. RELATED WORK

Loss differentiation algorithms can be categorized into three categories: (i) implicit or end-to-end differentiation and (ii) explicit loss differentiation algorithms. Unlike the implicit ones, explicit algorithms use agents that are deployed on the network's intermediate nodes. End-to-end or implicit solutions could involve the sender side only or both the sender and receiver sides (e.g. the 3 Duplicate Acknowledgements sent by the receiver to notify the sender of a packet loss).

A. *Implicit Differentiation Algorithm:*

These are end-to-end schemes [1],[2],[3] that modify the TCP congestion control algorithm to distinguish the losses because of the congestion in the network from the other random losses over wireless paths. They can be deployed easily via simple modification to the TCP congestion control at the sender or receiver side

TCP Westwood is an example of implicit loss classification algorithms. TCP Westwood is a sender-side modification of TCP New-Reno that estimates the connection bandwidth based on the rate of the received ACKs. TCP Westwood uses the estimated bandwidth to adjust and set its CWND and Slow-Start threshold parameters. This is in contrast to traditional TCP congestion control implementation, where both congestion window size and Slow-Start threshold are halved whenever a data packet loss is detected within the connection. This bandwidth estimation algorithm enhances the performance of TCP, in front of random, sporadic data packet losses (wireless channel related errors). In the link failure case, both TCP New-Reno and TCP Westwood recognize the packet loss with RTO expiration. Thus, both react the same way by backing off for a while and entering Slow-Start phase. In the link failure case, the average goodput of TCP Westwood is less than that of TCP New-Reno. This is due to the lost ACKs.

B. *Explicit Differentiation Algorithm:*

In an explicit loss notification approach, the receivers/network routers mark the acknowledgments with an appropriate notification of distinguishing the channel errors from the congestion losses. Then the senders respond to the notification. The explicit loss notification approaches require modifications to network infrastructure, the receiver and the senders.

.

III. METHODOLOGIES USED

A. *Explicit Congestion Notification (ECN)*

Explicit Congestion Notification uses unused bits in both the IP and TCP headers. At the Internet layer (for IP), a sending host must be able to indicate that it is capable of performing ECN and a router must be able to indicate that it is experiencing congestion when forwarding a packet. At the Transport layer, TCP peers must indicate to each other that they are ECN-capable. A receiving peer must be able to inform the sending peer that it has received a packet from a router experiencing congestion. The sending peer must be able to inform the receiving peer that it has received the congestion indicator from the receiving peer and has reduced its transmission rate.

ECN uses two bits in the TCP header that were previously defined as reserved. The two new flags defined for ECN support are the following:

- **ECE:** The ECN-Echo (ECE) flag is used to indicate that a TCP peer is ECN-capable during the TCP 3-way handshake and to indicate that a TCP segment was received on the connection with the ECN field in the IP header set to 11.
- **CWR**: The Congestion Window Reduced (CWR) flag is set by the sending host to indicate that it received a TCP segment with the ECE flag set. The congestion window is an internal variable maintained by TCP to manage the size of the send window.

Fig 2 shows the position of the above two bits in TCP Header.

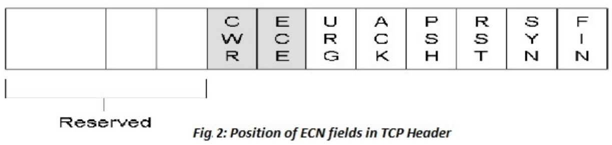

Fig 2: Position of ECN fields in TCP Header

The two bits have the following values:

- 00 - The sending host does not support ECN.
- 01 or 10 - The sending host supports ECN.
- 11 - Congestion has been experienced by a router.

An ECN-capable host sends TCP segments for an ECN-enabled TCP connection with the ECN field in the IP header set to either 10 or 01. An ECN-capable router that is experiencing congestion sets the ECN field in the IP header to 11. When a receiving TCP peer sends an acknowledgement, it sets the ECE flag in the TCP header and continues setting the ECE flag in subsequent acknowledgements. When the sending host receives the ACK with the ECE flag set, it behaves as though a packet was dropped by reducing its send window and running the slow start and congestion avoidance algorithms. For the next segment, the sender sets the CWR flag. Upon receipt of the new segment with the CWR flag set, the receiver stops setting the ECE flag in subsequent ACKs. Fig 3 shows an example of a TCP connection between ECN-capable TCP peers that experiences congestion by an ECN-capable router. TCP Peer A is sending data to TCP Peer B. TCP Peer A sends Segments 1 through 5. Segment 2 is forwarded by an ECN-capable router that is experiencing congestion, which sets the ECN field in the IP header to 11. When TCP Peer B receives this segment, it sends ACKs with the ECE flag set. When TCP Peer A receives the first ACK with the ECE flag set, it lowers its transmission rate and sends its next segment (Segment 6) with the CWR flag set. Upon receipt of the Segment 6 with the CWR flag set, TCP Peer B sends subsequent ACKs with the ECE flag cleared.

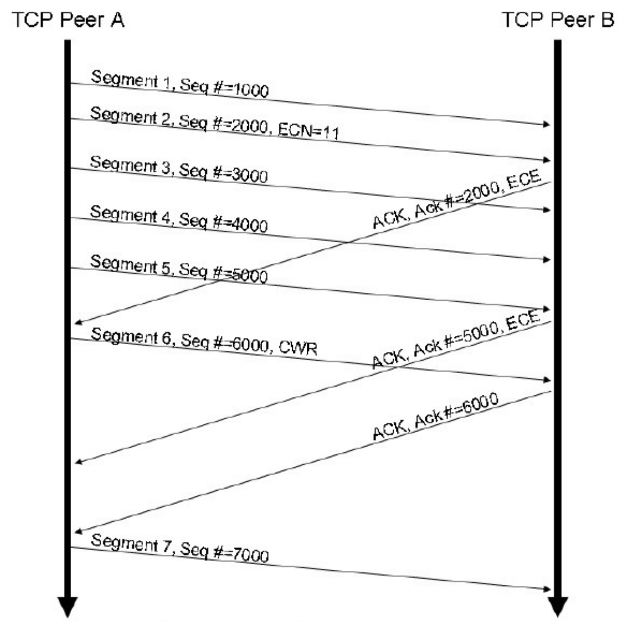

Fig 3: Example for ECN communication

B. Random Early Detection (RED)

The node congestion is often caused by buffer overflow queue, so the node queue management becomes the key for suppression of network congestion. The management of queue on the node is called Active Queue Management (AQM) mechanism. Random Early Detection (RED) was first proposed AQM mechanism and is also promoted by Internet Engineering Task Force (IETF). RED is most well known AQM algorithm. It averages queue length by using an exponential weighted moving average (EWMA) and calculate drop probability by applying a linear mapping function. The basic principles that RED uses is to monitor the average queue size for each output queue and use randomization to choose to notify that congestion has occurred. RED is primarily used to identify and notify both transient and longer-lived congestion. The primary benefits offered by RED include:

- Provide both congestion recovery and congestion avoidance.
- Avoid global synchronization and biases against bursty traffic.
- Maintain an upper bound of average queue size.
- Work with TCP and non-TCP transport-layer protocol.

IV. PROPOSED APPROACH

A. Methodology

A modification to the basic ECN principle is suggested in the way how the congestion loss is classified. A loss is classified as congestion related based on the below conditions:

- The number of packets that experienced congestion (EC) should be greater than 0
- The buffer queue size has not exceeded maximum capacity (MaxCapacity)
- The number of packets lost (NLP) as indicated by the received acknowledgement should be greater than zero and the current RTT should be greater than the sum of average RTT and the deviation observed in RTT.

All other cases are considered as non-congestion losses. Also, the principle has a linear complexity and yields optimized throughput results when simulated using the below simulation model.

B. Simulation Model

Fig 4 shows the diagrammatic representation of the wireless network scenario that is employed to perform this study.

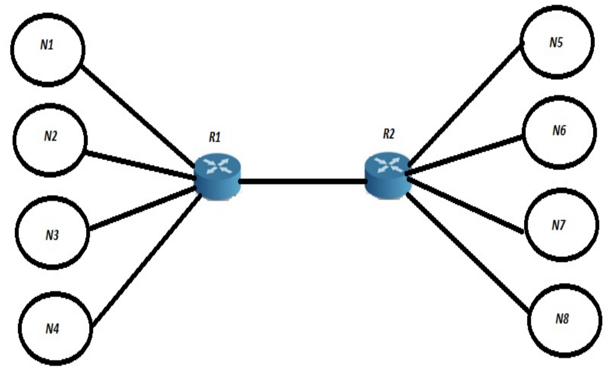

Fig 4: Network Scenario

As observed in the above figure, there are 8 nodes interconnected to form a wireless network. Nodes 1 to 4 are the sender nodes while, the nodes 5 to 8 are the receiver nodes. The bottleneck link is formed between the nodes R1 and R2. The nodes R1 and R2 act as wireless routers and use the concepts of Explicit Congestion Notification (ECN) for Congestion notification and Random Early Detection (RED) for Active queue Management. The wireless network uses the standard protocol 802.11 for communication. The maximum segment size is set to 512 bytes and the simulation time is set to 100 seconds.

The configuration parameters used while simulating the network are listed below:

Parameter	Value
Routing Protocol	AODV
TCP Variant	New Reno
Wireless Protocol	802.11
Simulation Time	100 seconds
Items to Send	1000
Item Size	512 bytes
Wireless Routers	5,6
Maximum Segment Size	512 bytes

The simulation is performed using Qualnet 5.0 simulator. Modifications have been made to the simulator setup such that the wireless routers support ECN and the active queue management is handled using RED.

C. Simulation Results and Observations

Nodes 1 to 4 are the source nodes while Nodes 8 to 11 form the destination nodes. Nodes 5 and 7 form the wireless routers. As part of this experiment, the throughput of wireless TCP was observed. The performance study included creating scenarios where only congestion losses were encountered and scenarios where both congestion and non congestion losses occurred. The results were extrapolated and graphically represented as shown in the Fig 5.

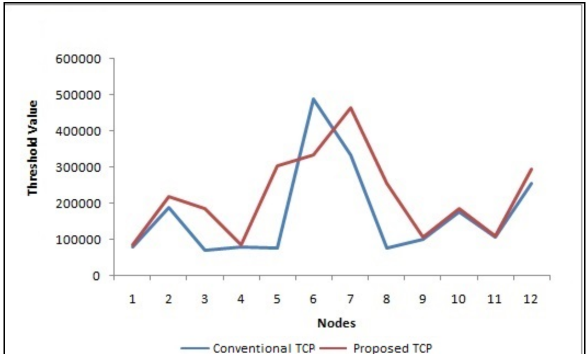

Fig 5: Throughput Comparison

It can be seen that the threshold values across all nodes have increased considerably for the proposed TCP due to reduced retransmissions and decreased window adjustments. Table1 lists the values corresponding to the graph shown in Fig 6.

Table 1: Throughput Comparison (Bits/sec)		
Node	**Conventional TCP**	**Proposed TCP**
1	79804	84302
2	187469	218128
3	69804	184976
4	79847	83983
5	77049	303422
6	488617	333810
5	335068	463187
6	76869	252984
7	99241	104436
8	176900	184607
9	107398	107173
10	255441	292914

Another interesting observation was that the peak queue size has decreased considerably across the nodes for the proposed TCP as observed in Fig 7. Since the peak queue size has decreased, it indicates that not much packets are waiting in the buffer queue for each nodes and also the wait time for packet transmission will subsequently get reduced.

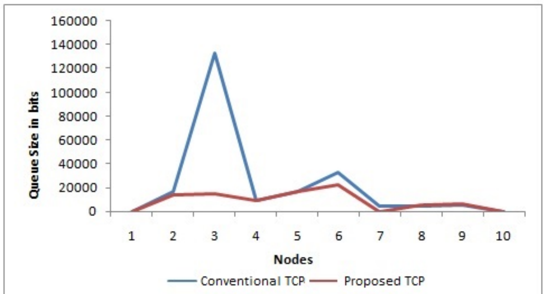

Fig 6: Peak Queue Size Comparison

Table2 lists the values corresponding to the graph shown in Fig 6. Based on the throughput observation and queue size, it can be observed that the performance of TCP in wireless network has improved significantly with the loss classification change. The performance can be further enhanced by making changes to tweak the way in which other congestion related parameters can be optimized.

Table2: Peak Queue Size		
Node	**Conventional TCP**	**Proposed TCP**
1	102	89
2	16808	14016
3	133336	14556
4	9980	9428
5	17332	16604
6	33248	22268
7	5092	124
8	5092	5172
9	5724	6276
10	124	164

Fig 7 represents the Qualnet 5.0 screen print for throughput comparison.

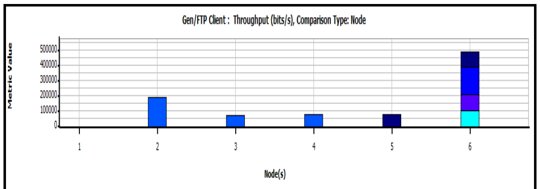

Fig 7: Qualnet Screenshot for Threshold Comparison

Fig 8 represents the Qualnet 5.0 screenshot for peak queue size analysis. The Qualnet analyzer provides the user to represent the output in various graphical representations.

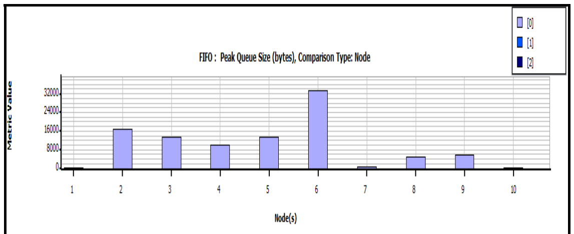

Fig 8: Peak Queue Size Screenshot

V. CONCLUSION AND FUTURE WORK

As discussed above, this paper provides an efficient way to distinguish Congestion and Non-congestion losses in wireless networks. Also, the network scenario has been simulated in Qualnet 5.0 simulator and the TCP throughput and queue size has been observed. The results indicate that the throughput has improved significantly and the queue size has decreased considerably for the proposed TCP model.

Future work can be in the direction where other parameters which influence congestion control such as RTT, RTO, buffer queue threshold etc., can be optimized and the changes can be included along with the current changes to achieve greater improvement in TCP performance.

REFERENCES

[1] Satoshi Utsumi , Salahuddin Muhammad and Salim Zabir, "A new high-performance TCP friendly congestion control over wireless networks" JOURNAL OF NETWORK AND COMPUTER APPLICATIONS 41 369-378, 2014 .

[2] A. G. T. Henderson, S. Floyd and Y. Nishida, “The newreno modification to TCP’s fast recovery algorithm,” IETF RFC 6582, Apr. 2012.

[3] Prasanthi. S, Sang-Hwa Chung, Changwoo Ahn, “An Enhanced TCP Mechanism for Detecting and Differentiating the Loss of Retransmissions over Wireless Networks”, International Conference on Advanced Information Networking and Applications, 2011.

[4] Nadim and Carey, “An Analytical Throughput Model for TCP NewReno”, IEEE/ACM Transactions on Networking, Vol.18, April 2010.

[5] Christos Antonopoulos and Stavros Koubias, “Congestion control framework for Ad-Hoc wireless networks”, Wireless Personal Communications, 52(4), 753-775 , 2010.

[6] Pham Thanh Giang and Pham Minh Vi, “Cross layer design to enhance TCP performance in Multi-hop Ad hoc Networks”, The International Conference on Advanced Technologies for Communications (ATC'13),2013.

[7] Wassim Ramadan, Eugen Dedu and Julien Bourgeois” EcnLD, ECN Loss Differentiation to optimize the performance of transport protocols on wireless networks” IEEE 2009.

[8] Liw Jia Seng, Mohd Noor Derahman and Azizol Abdullah “Loss Discrimination Algorithm for Wired/Wireless Networks” Journal of Computer Science 7 (12): 1798-1804, ISSN 1549-3636, 2011.

[9] Mohammad Amin Kheirandish Fard Sasan Karamizadeh and Mohammad Aflaki “Packet Loss Differentiation of TCP over Mobile Ad Hoc Network Using Queue Usage Estimation”IEEE 2011.

[10] Dhiman Barman Ibrahim Matta “Effectiveness of Loss Labeling in Improving TCP Performance in Wired/Wireless Networks”, Proceedings of the 10 th IEEE International Conference on Network Protocols (ICNP’02), IEEE 2002.