

# Performance Analysis of TCP in Wireless Network

Amit T Kukreja  
K.J.Somaiya College of  
Engineering Vidyavihar  
Mumbai

Manish M Potey  
K.J.Somaiya College of  
Engineering  
Vidyavihar, Mumbai

Prasanna J Shete  
K.J.Somaiya College of  
Engineering  
Vidyavihar, Mumbai

## ABSTRACT

In this paper, the comparison of performance of TCP over the wired cum wireless network using only ECN and only RED is done and then combine effect of both is observed with different version of TCP like Tahoe and Reno. The comparison is done on the basis of parameter throughput v/s number of nodes.

## General Terms

Transmission Control Protocol (TCP), QualNet(5.0)

## Keywords

Explicit Congestion Notification (ECN), Random early detection (RED)

## 1. INTRODUCTION

TCP is one of core protocols of the Internet protocol suite. It has in built mechanism to behave network friendly. Within a connection TCP can guarantees such as in order delivery or reliable data transmission using retransmission techniques. TCP adapt the transmission rate of the packets to the available bandwidth. The number of packets that have not yet received is bounded by a parameter called a congestion window. If it encounters packet loss, it assumes network internal congestion and slow down the transmission rate. There are two phases to the window adjustment algorithm .The connection begins in slow start phase and current congestion window is doubled each roundtrip time until it reaches the slow start threshold. A TCP packet is considered lost if three repeated ACKs for same packets arrive at the source or if ACKs for transmitted packet does not arrive within specified time out period.

TCP has been optimized for wired networks. Any packet loss is considered to be result of congestion and congestion window is reduced dramatically as a precaution. Wireless links are known to experience sporadic and usually temporary losses due to fading, shadowing, handoff ,and other radio effects, that cannot be considered congestion. TCP uses a number of mechanisms to achieve high performance and avoid 'congestion collapse', where network performance can fall by several orders of magnitude. These mechanisms control the rate of data entering the network, keeping the data flow below a rate that would trigger collapse. Coupled with timers, TCP senders and receivers can alter the behavior of flow of data. This is more generally referred as congestion control and /or network congestion avoidance. With increasing the

number of nodes on sender and receiver side increases the congestion which in turn reduces the average throughput.

In the next section TCP variants such as TCP Tahoe and TCP Reno are explained in detail. In the section 3 congestion avoidance mechanism such as RED and ECN are explained. In section 4 Implementation details are shown and results are discussed.

## 2. TCP VARIANTS

### 2.1 TCP Tahoe

TCP is based on principle of 'conservation of packets', i.e. if the connection is running at the available bandwidth capacity then a packet is not injected into the network unless a packet is taken out as well. TCP implements this principle by using the acknowledgements to clock outgoing packets because an acknowledgment means that a packet was taken off the wire by the receiver

Tahoe [1] suggests that whenever a TCP connection starts or restarts after a packet loss it should go through a procedure called 'slow-start'. The reason for this procedure is that an initial burst might overwhelm the network and the connection might never get started. Slow starts suggest that the sender set the congestion window to 1 and then for each ACK received it increase the cwnd by 1. So in the first round trip time (RTT) it send 1 packet, in the second RTT it send 2 and in third it send 4 packets. Thus it increases exponentially until a packet loss which is a sign of congestion is detected. When congestion occurs, it decreases the sending rate and reduce congestion window to one and start over again. The important thing is that Tahoe detects packet losses by timeouts. In usual implementations, repeated interrupts are expensive so coarse grain time-outs which occasionally checks for time outs. Thus it might be some time before one notice a packet loss and then re-transmit that packet.

#### 2.1.1 Congestion Avoidance:

Tahoe uses 'Additive Increase Multiplicative Decrease' for congestion avoidance. A packet loss is considered as a sign of congestion and Tahoe sets the threshold value as half of current window. It then set cwnd to one and starts slow start until it reaches the threshold value. After that it increments linearly until it encounters a packet loss. Thus it increase it window slowly as it approaches the bandwidth capacity.

### 2.1.2 Problems:

Tahoe takes a complete timeout interval to detect a packet loss and it takes even longer because of coarse grain timeout. It sends cumulative ACKs for that it follows 'go back n' approach. Every time a packet lost it waits for timeout and pipeline is emptied.

## 2.2 TCP Reno:

It retains the basic principle of Tahoe such as slow start and coarse grain re transmit timer. It detects lost packets earlier and pipeline is not emptied every time a packet is lost.

In case of Reno the duplicate ACK is received if next segment in sequence has been delayed in the network and segments reached there out of order or that packet is lost. The three duplicate ACK are considered as sign of lost segment, so it goes for retransmission of that segment without waiting for timeout. This is known as "Fast Retransmit". After a packet loss, it does not reduce cwnd but it set the value of slow start threshold to half the current window size and congestion window to same value. For each duplicate ACK received increase cwnd by one. If cwnd is now greater than amount of data in the pipe then transmit a new segment else wait.

### 2.2.1 Problems:

It performs well over TCP when packet losses are small. In case of multiple packet loss in a single window TCP RENO's performance is almost same as TCP Tahoe The information about 2<sup>nd</sup> packet loss will come after ACK for retransmitted first segment reaches the sender after one RTT. It is possible that cwnd is reduced twice for packet loss which occurs in single window.

## 3. AUGMENTATION WORK

### 3.1 Random Early Detection (RED)

RED [2] is a congestion avoidance mechanism implemented in routers that work on the basis of active queue management. In case of traditional queue management algorithms when buffer is full then packets are dropped, whereas in RED router signals incipient congestion to TCP by dropping packets probably before queue runs out of buffer space. The drop probability is dependent on average queue size to avoid any bias against the heavy traffic. A RED router randomly drops arriving packets with a probability that belong to a particular flow that is proportional to flow share of bandwidth. If sender is using more bandwidth, it gets more of its packet dropped.

RED algorithm consist of two main parts: estimation of average queue size using a simple exponential weighted moving average queue length computation (wq) and decision of whether or not to drop an incoming packet. Two parameters  $min_{th}$  and  $max_{th}$  represent thresholds set by RED.  $min_{th}$  specifies the average queue size below which no packets are dropped, while  $max_{th}$  specifies the average queue size above which all packets are dropped. As average queue size varies from  $min_{th}$  to  $max_{th}$ , packets will be dropped with a probability  $p_a$  that varies linearly from 0 to  $max_p$  where  $p_a$  is a function of average queue size. As the average queue length varies between  $min_{th}$  and  $max_{th}$ ,  $p_a$  increase linearly towards a configured maximum drop probability,  $max_p$ . Dropping packets in this way ensures when subset of the source TCP packets get dropped and

they invoke congestion avoidance algorithms that will ease congestion at the router.

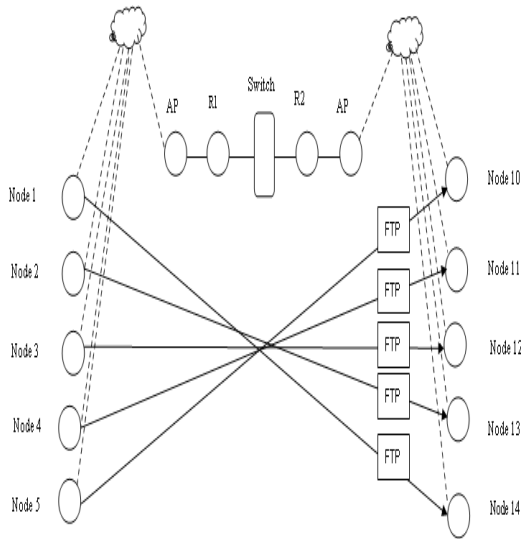
## 3.2 Explicit Congestion Notification (ECN)

ECN[3][4][5] provides a light-weight mechanism for routers to send direct indication of congestion to source. When avg is between  $min_{th}$  and  $max_{th}$ , ECN marks instead of dropping, an incoming packet probabilistically. The marking probability in ECN varies as RED. A connection receiving congestion notification in the form of the ECN marking cuts its congestion window in half just as if it had detected a packet loss. When avg is above or equal to  $max_{th}$ , ECN also drops deterministically all incoming packets since ECN marks packets before congestion actually occurs. Upon receipt of a congestion marked packet, the TCP receiver informs the sender about incipient congestion which in turn trigger the congestion avoidance algorithm at the sender. ECN requires support from both the router as well as end hosts, i.e. end hosts stack needs to be modified. Packets from the flows that are not ECN capable will continue to be dropped by RED.

There are two main changes that need to be made to add ECN to TCP to an end system and one extension to a router running RED. It is necessary that router identifies that a packet is ECN capable. It uses two bits in IP header. The ECN Capable Transport (ECT) bit is set by sender end system if both the systems are ECN capable. Packets encountering congestion are marked by router using the Congestion Experienced (CE) bit. On their way to the receiver end system with a probability proportional to the average queue size following the procedure used in RED routers. For packets from ECN capable hosts, the router marks the packets rather than dropping them. For changes at the router TCP host side reserved field is designated as ECN-Echo (ECE) flag and Congestion Window Reduced (CWR) flag. These two bits are used for initializing phase in which the sender and receiver negotiate the capability and desire to use ECN, as well as for subsequent actions to be taken in case there is congestion experienced in the network during the established state.

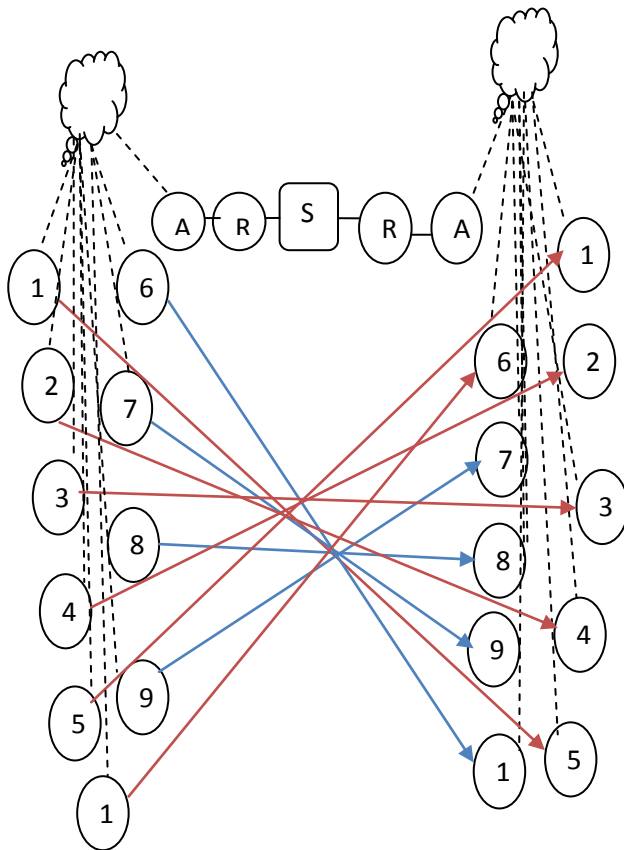
## 4. IMPLEMENTATION OF RED AND ECN

The figure 1 consist of 14 nodes of which 10 nodes are wireless, 2 nodes are wired cum wireless which act as base station (access point), remaining 2 wired nodes which will act as router and they are connected to each other through switch. With TCP Tahoe first the RED is applied and observed the performance. Then on the same network with TCP Tahoe ECN is applied the and observed the output. Further combined ECN as well RED applied simultaneously with TCP Tahoe and observed the output. Similarly, the above procedure is repeated for TCP RENO.



**Figure 1: 5 Nodes output**

The above procedure is repeated with increasing the number of nodes i.e. by taking 10 nodes transmitting simultaneously to 10 destination nodes as shown in fig. 2.



**Figure2: 10 Nodes Output**

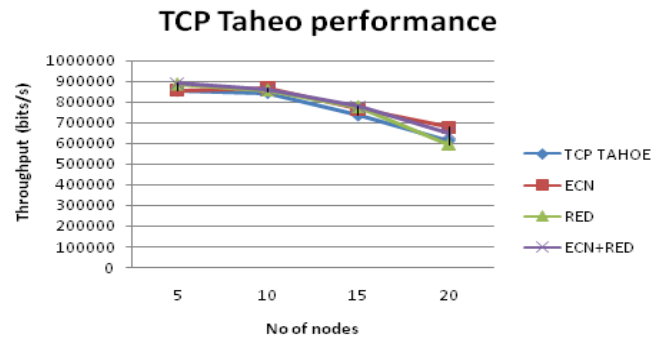
Then the number of nodes is increased to 15 and 20 to both source and destination and congestion avoidance mechanism such as RED is applied and in then another mechanism ECN is applied and combined effect of both ECN+RED is observed with both variants of TCP i.e. TCP Tahoe and TCP Reno.

### 5. RESULT AND ANALYSIS OF THE NETWORK:

**Table1: Throughput of TCP Tahoe**

Throughput ↓ No of nodes	TCP Tahoe	ECN	RED	ECN+RED
5	855932	854390	885958	890106
10	845173	866202	857759	861465
15	736344	762805	776803	777097
20	613083	677501	594143	651225

The above table shows the performance of TCP Tahoe in terms of throughput (bps) V/s no of nodes with different congestion avoidance mechanism like ECN and RED.



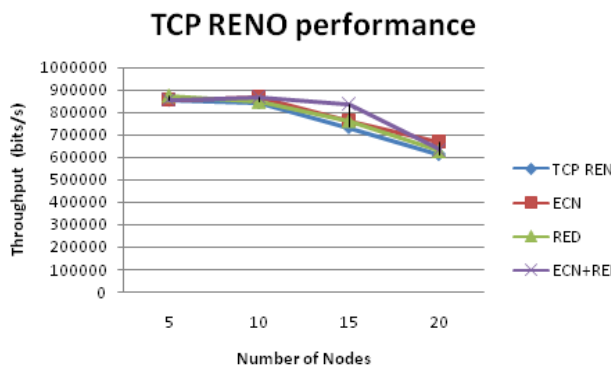
**Figure 3: TCP Tahoe Performance**

The throughput of the network decrease with increasing number of nodes because of congestion in the network. Also with increasing the number of nodes the performance of network with ECN +RED is better than that of applying only ECN or only RED as it provides the advantages of both the mechanisms. RED performance is better than ECN when number of nodes is less in the given network as shown in the figure

**Table 2: TCP Reno Performance**

Throughput ↓ No of nodes	TCP Reno	ECN	RED	ECN+RED
5	855932	854390	872456	854036
10	845173	866202	846117	865703
15	733737	762805	761912	836127
20	612433	669045	629278	633114

The above table shows the performance of TCP Reno in terms of throughput V/s no of nodes with different congestion avoidance mechanism like ECN and RED



**Figure 4: TCP Reno Performance**

Above graph shows that TCP Reno performance in terms of throughput is better with applying ECN+RED with increasing number of nodes as it provides advantages of both. Also as number of nodes increases the congestion in the network increases which reduces the throughput. ECN provides notification to sender to slow down the sending rate of the packets.

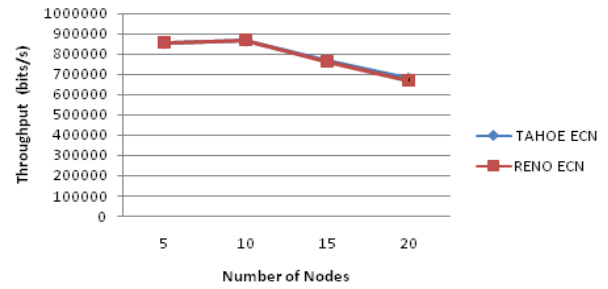
## 5.2 Comparing The Performance Of Both TCP With ECN, RED And Combined.

**Table 3: Different TCP with ECN**

Throughputs ↓ No of nodes	Tahoe + ECN	Reno + ECN
5	854390	854390
10	866202	866202
15	762805	762805
20	677501	669045

The above table shows the performance of different TCP in terms of throughput(bits/s) with no of nodes by applying ECN.

**Performance of TCP's with ECN**



**Figure 5: TCP performance with ECN**

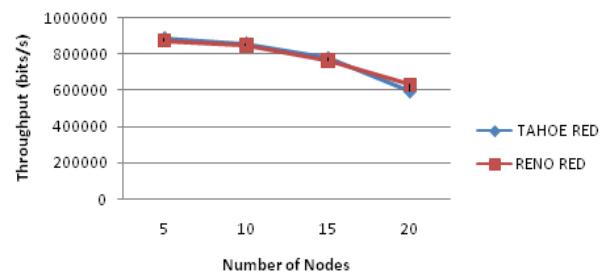
The above graph shows that with ECN the throughput of both the TCP i.e. Tahoe and Reno is almost same. ECN provides notification to sender to slow down the sending rate when congestion occurs. so loss of packets will be reduced.

**Table4: Different TCP with RED**

Throughputs ↓ No of nodes	Tahoe + RED	Reno + RED
5	885958	872456
10	857759	846117
15	776803	761912
20	594143	629278

Here comparison of TCP Reno and Tahoe with RED is done.

**Performance of different TCP with RED**



**Figure6: TCP performance with RED**

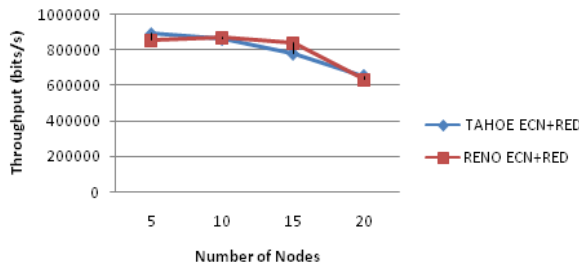
The above graph shows that with RED, TCP RENO performance is better as compared to that of Tahoe with increasing number of nodes. TCP Reno detects the lost packets earlier.

**Table5: Different TCP with ECN+RED**

Throughput ↓ ↘ ↗ ↑ No of nodes	Tahoe with ECN+RED	Reno with ECN+RED
5	890106	854036
10	861465	865703
15	777097	836127
20	651225	633114

The above table shows combine effect of both the mechanism i.e. ECN and RED with TCP variants.

**Performance of different TCP with ECN+RED**



**Figure 5: TCP performance with ECN+RED**

The throughput of TCP Reno is better than that of TCP Tahoe when we apply both ECN and RED simultaneously. TCP Reno detects the lost packets earlier and ECN provides notification of congestion to the sender to slow down the sending rate.

## 6. CONCLUSION

The throughput of the network decreases with increase in the number of nodes as congestion occurs. The performance of network is improved with congestion notification mechanism such as ECN and congestion avoidance such as RED. The combine effect shows overall improvement in the performance. TCP Reno detects lost packets earlier and hence shows better performance than TCP Tahoe.

## 7. ACKNOWLEDGMENTS

Our thanks to Head of IT department, teaching and non teaching staff of IT department for allowing us to access the QualNet software.

## 8. REFERENCES

- [1] V. Jacobson “Congestion Avoidance and Control ” SIGCOMM Symposium on communication Architecture and Protocols.
- [2] S. Floyd, V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance” IEEE /ACM Transactions on Networking, Vol., No 4, August 1993.
- [3] S.Floyd., “TCP and Explicit Congestion Notification”, ACM Computer Communication Review, vol. 24, no. 5, pp. 10-23, October 1994.
- [4] S. Floyd and K.Fall, “Promoting the Use of End-to-End Congestion Control in the Internet”, IEEE/ACM Transactions on Networking, vol. 7, no. 4, pp. 458-472, 1999.
- [5] S. Kunniyur . and R. Srikant , “ End-to-end congestion control schemes: utility functions, random losses and ECN marks” in proceedings of IEEE infocom Vol.3 March 2000.
- [6] Robert R.Chodorek. “A Simulation of ECN-Capable multicast Multimedia Delivery in NS-2 Environment.” Department of Telecommunications. The AGH University of Technology al.Mickiewicza 30 30-059 Kraków Poland
- [7] M. Alnuem, J. Mellor, R. Fretwell “Performance evaluation of ECN-based TCP error discriminator over wireless networks”. Mobile Computing and Networks Research Group.School of Informatics.University of Bradford,Bradford, UK.
- [8] Ning Yang . “Congestion-Aware Cross-Layer Design for Wireless Ad Hoc Networks”.
- [9] Bagal,P.,Kalyanaraman, S., and Packer, B.,“Comparative study of RED, ECN and TCP Rate Control”, Technical Report, March 1999.
- [10] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links.” IEEE/ACM Transactions on Networking 5(6), pp. 756– 769, 1997
- [11] Braden, R, “Recommendations on Queue Management and Congestion Avoidance in the Internet”, RFC2309, April 1998