Indian Journal of Science and Technology, *Vol 8(16), DOI: 10.17485/ijst/2015/v8i16/62902, July 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Analysis on TCP Packet Reordering Problem in Mobile Ad-Hoc Network

J. Govindarajan[1*], N. Vibhurani[1] and G. Kousalya[2]

[1]Department of CSE, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore – 641112, Tamil Nadu, India; j_govindarajan@cb.amrita.edu, n.viburani@gmail.com
[2]Coimbatore Institute of Technology, Coimbatore – 641014, Tamil Nadu, India; kousir@gmail.com

Abstract

In Ad-hoc network, Packet reorder is effect of many causes like congestion loss, packet loss due to corruption, multi-path routing and node's mobility. Transmission Control Protocol (TCP) performs poorly over wireless links which misinterprets the reordered packets as lost packets due to congestion. This has motivated researchers on developing TCP variants towards packet reordering problem for enhancing TCP performance. Some researchers have found the opportunity to use existing TCP variants like SACK, TCP-Westwood to address the reordering problem. TCP-DCR (TCP-Delayed Congestion Response) and RR-TCP (Reordering Robust TCP) are the TCP variants to acknowledge the packet reordering at the end level. In the existing investigations, the protocols were tested in a simulation environment with one or two causes (congestion or error) at a time over a simple topology. In our experiments, the simulation study of TCP variants has been done with all reordering causes simultaneously, over a large ad-hoc environment including mobility and multi-path routing. A performance analysis had been done using reordering metric (percentage of reordering) which accounts for earlier and lately arriving reordered packets and estimated using throughput and goodput metrics. From our analysis, it is observed that the Selective acknowledgement option of SACK-TCP and the delayed congestion response of DCR help to achieve better performance than other variants in presence of low reordering due to high congestion and low error. However, these variants were failed to meet up their performance in case of high reordering (reordering greater than 50%) due to high congestion loss and high channel error losses. This observation has given the research opportunity to incorporate a discrimination algorithm in the existing reordering solutions to classify the packet error from packet reordering.

Keywords: Analysis, End-to-End Performance MANET, Packet Reordering, TCP

1. Introduction

Transmission Control Protocol (TCP)[1] is a connection oriented protocol, for the development of internet applications which is in usage for several decades, currently has many challenging problems like protecting the users' data and their services from attackers[22]. Similarly, MANET also has many challenging problems like designing efficient routing protocols and providing QoS[18,23] over the dynamic network with limited power nodes. The deployment of TCP based applications which are designed for wired network over wireless and ad-hoc network has given a new research direction to the design of customized protocol. As compared to wired environment, wireless channel provides lossy and noisy transmission due to attenuation, environment interferences and noise. In addition to propagation effects of wireless medium, the frequent path change due to mobility of Mobile Ad-hoc Network (MANET) nodes affects the transmission. These non-congestion events lead to random packet losses in the wireless environment. In TCP, packet loss is detected either by duplicate ACK or timeout. The loss is concluded

**Author for correspondence*

as the indication of network overload (congestion), hence it reduces its sending rate. Since the losses are mapped into congestion, even for non-congestion events TCP reacts by reducing its sending rate. The inability of TCP to distinguish the packet loss due to non-congestion events from congestion events leads to spurious retransmissions and unnecessary reduction in the sending rate.

The congestion and non-congestion losses constitute the dominant cause for packets to arrive in an out of order sequence by packet retransmission. In addition, Multi-path routing and behavior of underlying protocol also contribute to arrival of packets in out of order sequence. For example, LLRTX[4,5] Link-Layer Retransmissions mechanism retransmits the packets upon loss detection. This retransmission causes the packets to arrive later which leads to sequence discontinuity. In common, reordering of packets simply means the earlier sent packets (or segments) by sender arrives at the destination later and vice versa. In[13] Packet reordering is defined as "network behavior where the relative order of some packets in the same flow is altered, when these packets are elated in the network". This packet reordering problem degrades the performance of TCP by unnecessary reduction in TCP's sent rate.

Many TCP variants are recommended by researchers towards the packet reordering problem for improving the performance of TCP. In[2,3] authors have stated that the variants endeavor against packet reordering either by postponing the congestion response over time period or by revoking past congestion response.

The rest of the paper is structured as follows. Section 2 describes the motivation behind the detailed study. Section 3 details the causes and impacts of packet reordering on TCP over MANET. In Section 4, the existing solutions are summarized. Section 5 details the simulation environment and a performance study of the protocols under investigation. Section 6 concludes and discusses some possible directions for the improvement of TCP.

2. Motivation for Packet Reordering Study over MANET

Based on the following motivations, we have done our study on packet reordering over MANET.

- In the existing analysis on packet reordering[3], general causes for packet reordering and its effects are discussed.

Since the characteristics of MANET are different from other types of networks like wired network, wireless infrastructure based network, an in depth study on packet reordering specific to MANET is required.

- Even though many causes are mentioned in3, only route fluttering (out of many causes) is considered for the analysis. Actually route fluttering is an effect of many causes like frequent change in link cost. They simulated the route fluttering effect by changing the path delay.

For in depth analysis, simulation study should be performed with all possible causes for reordering.

- In[13], the comparisons of TCP variants over packet reordering by varying the Link layer retransmission limit are done over multi-hop network and wireless infrastructure network. And the simulations were done by varying the packet error rates over wireless networks. A simple wireless network with one base station and two end users was simulated. Also a chain topology with five nodes without base station was also simulated.

For in depth analysis, a larger size ad-hoc network should be considered.

- In[3,13], the analysis was performed only with throughput and goodput. These metrics help only to measure the achieved performance. High throughput or high goodput may be achieved in the absence of reordering.

Hence, in depth analysis should be based on the level of reordering that is analyzing the behavior of TCP variants for the various levels of reordering.

The main aim of this work is to analyze the behavior of TCP variants over large size Mobile Ad-hoc Network with all possible packet reordering causes using a metric to measure the level of reordering.

3. Packet Reordering over MANET

3.1 Causes of Packet Reordering

Packet reordering phenomena mainly affects the performance of Transport layer protocols. The receipt of out of sequence packets occurs due to congestion and non-congestion events. In[3] the general causes for packet reordering were listed. Among the general causes, we have listed the events that cause the packets to reorder at the end level over an MANET environment. Figure 1 depicts the various packet reordering causes.

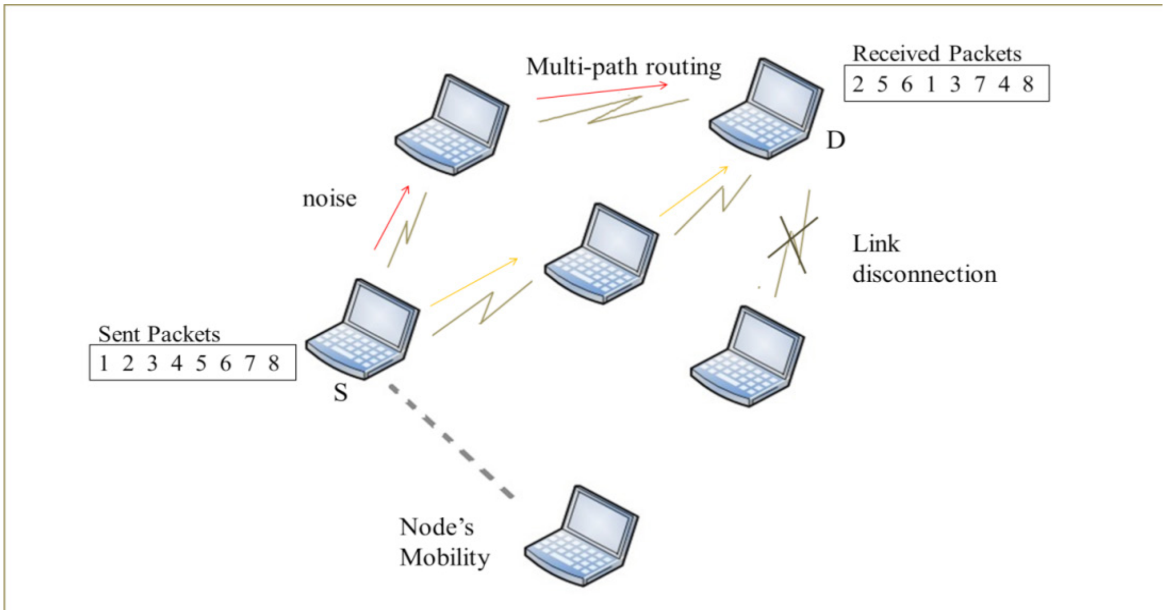


Figure 1. Depiction of various packet reordering causes.

3.1.1 Packet Retransmission due to Congestion Loss

The self-clocking principle of TCP says that the increase in receiving rate of ACK will yield higher sending rate. After some number of cycles, the network may reach its congestion level. This network congestion is a common event in the packet switching network. To recover the lost packets, senders follow the retransmission procedure. Since the receiver is receiving latest packets earlier and earlier packets later, the received packets are in an out of sequence at the receiver side.

3.1.2 Packet Retransmission due to Channel Error

Unlike wired, wireless links are more error-prone. A packet transmitted over wireless is lost due to channel errors and hence it will not reach the destination in time. There retransmission comes into effect and causes further packets to reorder at the end-to-end level (receiver).

3.1.3 Route Fluttering due to Mobility

MANET nodes have the flexibility to change their positions, without any manual procedure for physical disconnection. This mobility results the path instability, where the available path to a destination may utter. The frequent changes in available routes may cause the packets to reach the destination in a sequence disorder.

3.1.4 Packet-Level Multi-path Routing

For tackling the congestion in the network, routing the packets in multiple paths is endorsed. This Multi-path10 routing balances the traffic load over wired/wireless links than single path routing and enhances the performance of Transport protocols. This will increase the bandwidth utilization and capacity over links. This packet level multi-path routing allows packets to be forwarded over multiple paths simultaneously of same traffic for achieving load balance in the network. These packets arrive at the destination later or earlier depending upon their dissimilar path delays. These packets arrive in an out of sequence at the destination.

4. Impacts of Packet Reordering in the TCP End-To-End Performance

Packet reordering is a cumulative effect of many causes which are discussed in previous section. Unlike packet loss due to congestion or bit corruption, the reordered packets will reach the destination successfully and hence loss recovery and CWND (Congestion Window) reduction are unnecessary. However, receiving the packets in out of order will have more impacts on the performance of TCP. These impacts over MANET are discussed below.

4.1 Spurious Segment Retransmission

The reordered packets trigger the receiver to send duplicate ACK by misinterpreting the reordered packets as lost packets. The source initiates the congestion response by receiving more than three duplicate ACKs and the inferred loss packets are retransmitted. This spurious retransmission generates additional yet unnecessary traffic load which leads to further decrease in the sending rate thereby performance of TCP slows down.

4.2 Unnecessary Reduction in TCP Sending Rate

Multipath routing was designed for dynamic network like Ad-Hoc network to utilize all possible paths to reach the destination. Node's mobility causes frequent path change. Both Multi-path routing and Node's mobility yield persistent reordering i.e. probability for packet reordering is high. This persistent reordering triggers the congestion response by receiving more duplicate ACKs or timeout at the source. The frequent reduction in CWND leads to low sending rate. Hence the utilization of available bandwidth will be very low.

4.3 Underestimation of RTT and RTO

Karn's algorithm[14] says that the measured RTT samples of retransmitted packets should not be considered for RTO estimation. The principle helps the sender to avoid RTT calculation using false RTT samples in the case of reordering and early timeout. However, in the case of persistent reordering, all measured RTT samples will be discarded and hence sender fails to reflect RTT variation in the RTO estimation.

5. Related Solutions and it's Investigations

To improve TCP performance, researchers have contributed, by proposing a modification to the basic TCP congestion control procedure. In[3], authors specified few TCP Variants and they classified the variants based on its nature. In[11,13] authors have suggested that bandwidth estimation procedure of TCP-W and Selective Acknowledgement procedure of TCP-SACK may help the sender to maintain its CWND in presence of reordering which reported as loss. These algorithms are summarized in this section.

5.1 Eifel Algorithm

This algorithm[6] inserts the Timestamp option in the header of the segment for detecting spurious retransmission. When the sender determines the last retransmission as spurious it revokes congestion response by restoring the congestion window. This algorithm fails if the transmitted packet and the retransmitted packet get reordered.

5.2 RR-TCP

Reordering Robust TCP (RR-TCP)[7,8], by Zhang which reduces dupthresh (duplicate threshold) dynamically depending on fast false retransmit ratio (FA ratio) and send additional one ACK clocked CWND data. This improves the connection throughput whereas RR-TCP maintains the reordering histogram for every reordered segment, which is an additional overhead.

5.3 Leung-Ma Algorithm

Like RR-TCP, Leung-Ma Algorithm makes use of dupthresh to avoid spurious retransmission but includes the mean deviation length of reordering segments. A mechanism[9] for reducing the dupthresh while retransmission timer expires is also included. It helps in improving connection throughput and works similar to RR-TCP but incurs less overhead.

5.4 TCP-DCR

TCP-Delayed Congestion Response (TCP-DCR)[11] by Bhandarkar and Reddy focuses on delaying the congestion response over a time period. This can be done by timer based delay (waiting over one RTT) or by ACK based delay. Delaying the congestion response avoids spurious retransmission meanwhile receiver sends the cumulative ACK for the received packets. TCP-DCR sends one new segment while receiving duplicate ACK and postponing response helps for maintaining larger congestion window. This improves the connection throughput and helps in avoiding spurious retransmissions.

5.5 TCP Westwood (TCPW)

TCP-Westwood[14], a TCP variant decouples the congestion loss and random packet loss over the lossy wireless links. Upon the receipt of ACK, TCPW sender estimates the available bandwidth, and then adjusts the congestion window for further transmission. Using this estimation, upon congestion loss, it selects the ss thresh at the time of congestion response, without blindly halving down congestion window. Duplicate ACK is also included in the bandwidth estimation. This bandwidth estimation helps by being friendly when compared to other variants over wireless links.

5.6 SACK-TCP

SACK-TCP (Selective Acknowledgement[12] is a TCP variant which helps in reporting the lost packets during multiple losses in single congestion window). This SACK option avoids spurious retransmission of successfully arrived packets and enhances TCP performance3 suggests that the temporal approach like TCP-DCR helps in reducing the spurious retransmissions due to reordering which keeps the congestion window as large and sustaining a higher connection throughput. Even though algorithms like Eifel, Leung-Ma and RR-TCP attempt in enhancing TCP performance, they failed to resolve the potential loss due to packet reordering. Hence theses algorithms are excluded from our analysis. In this paper we compare TCP-DCR with other variants (TCPW, SACK-TCP, and TCP-New Reno) in presence of persistent reordering by simulating all reordering causes instantaneously over wireless Mobile Ad-hoc Network.

6. Performance Analyses

6.1 Simulation Design

NS2 Simulator (NS 2.35) has been chosen to conduct the simulation experiments. The following parameters (Table 1.) are configured to simulate the dynamics of a real ad-hoc network.

Since our focus is on analyzing the performance of packet reordering in worst case scenario, a random set of nodes, which are 400m apart have been configured as sources and destinations at runtime. Simulations were conducted with varying number of flows, and variable queue size of mobile nodes to invoke the congestion loss inside the network. The experiments with varying channel error rate (Bit Error Rate - BER) help to simulate the packet loss due to bit corruption. Mobility of the nodes and multi-path routing will invoke the route fluctuations. Gauss-Markov mobility model is a temporal dependency model[16] which closely maps the model into real world i.e. the future location and velocity of a mobile node are correlated with its past location and velocity[17]. To incorporate this mobility model with our simulation environment, we have used BonnMotion-1.7 tool[19], a predominant Java software tool used for generating and analyzing the mobility models and for multi-path routing, MP-OLSR (Multi-Path Optimized Link State Routing) protocol has been used. MP-OLSR[18] is a fault tolerant and load balancing multipath routing protocol, used for simulating multiple paths routing simultaneously. In all simulations, FTP is configured as traffic source with the maximum packet size of 1500 bytes. The topology which was considered for our experiments is depicted in Figure 2. The simulations are grouped into cases as detailed in Table 2.

Table 1. Simulation Parameters

Simulation area	800m x 800m
Simulation time	1000 sec
Number of mobile nodes	50
Number of TCP Flows	5 , 10, 15 ,20
Mobility Model	Gauss-Markov Mobility Model
Queue size	20,100
Error Model	Uniform Error Model
Error rate (BER)	0, 0.0001 , 0.001
Routing Protocol	MP-OLSR (Multi-path Optimized Link State Routing)
Transport level protocols	TCP-DCR/TCPW/SACK-TCP/TCP-New Reno

Table 2. Simulation Cases

Cases	Queue size (No. of Packets)	Channel Error Rate (BER)	Mobility Model	Multipath Routing
1. Low Congestion and No error	100	0	Gauss – Markov Mobility Model	MP-OLSR Multipath – Optimized Link State Routing
2. High Congestion and No error	20	0		
3. Low Congestion and Low error	100	0.0001		
4. Low Congestion and High error	100	0.001		
5. High Congestion and Low error	20	0.0001		
6. High Congestion and High error	20	0.001		

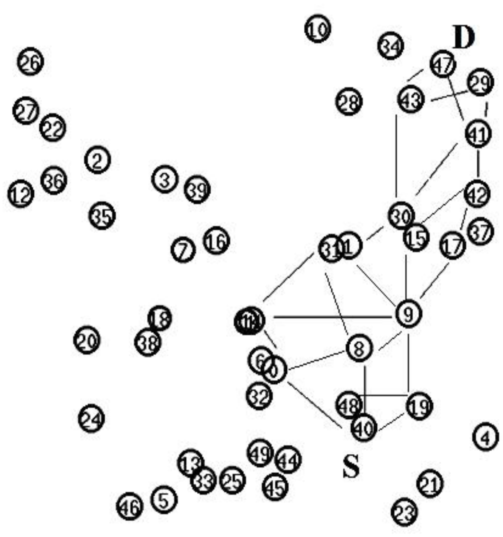


Figure 2. A Topology with 50 nodes.

(The multiple paths between one source-destination pair are highlighted).

6.2 Methodology for Analysis

Since we have designed our experiments to map the real world scenario with simulation, the flows are distributed by the congestion loss, error loss, path unavailability and failure to access the channel. Hence we followed a methodology (discussed below) for our analysis.

For the purpose of classification of flows, the reorder ing ratio and MAC level Channel access probability are used. The flows which are not finding any path during the simulation time are excluded from the analysis and the remaining flows are classified based on the measured reordering ratio. Some flows are affected by high congestion; high channel error rate and multi-path, hence achieve low end-to-end performance. Some of them will achieve high end to end performance due to less congestion, low channel error rate and low multipath packet reordering. The high congestion, high channel error rate and the transmission of packets through multiple paths will contribute to severe packet reordering in the network. After consolidating the reordering ratio of all the flows for all possible scenarios, we found that the maximum reordering ratio is 60% and low reordering ratio is 0%. Therefore, the flows which are having reordering ratio between 0% and 20% are grouped into low reordering level, the flows which are having reordering ratio between 21% and 40% are grouped into medium reordering level and the flows which are having reordering ratio between 41% and 60% are grouped into high reordering level. The flows which are closer to the higher boundary of a level are moved to next higher level. For example, a flow with 39% of reordering ratio is moved to high reordering level.

6.2.1 Packet-Reordering Ratio

In presence of reordering, it is desirable to affirm the degree of reordered packets, with respect to other packets. The packet reordering ratio metric is defined[20] as,

$$\text{Reorder Ratio} = \frac{\text{Count of packets with Type-p-reordered} = \text{True}}{L} \quad (1)$$

Where, L is the number of arrived packets at the destination and “count of packets with Type-p-reordered = True” represents the number of reordered packets. This is can be expressed as percentage by multiplying 100 to it. Duplicate packets are excluded from calculation.

The variable Type-p-reordered is a flag to designate the reordered packets. The RFC 4737[20] has given a procedure to designate the incoming packets using the variable “Type-p-reordered”[21] recommend incorporating both lateness and earliness packets for packet reordering ratio calculation. As per the recommendation of RFC 5236[21] the procedure has been modified as follows (modifications is highlighted),

If s = NextExp then /* s is in-order */

NextExp = s + 1;

Type-P-Reordered = False;

Else if s < NextExp or s > NextExp /* s is in out-of-order (lateness and earliness)

Type-P-Reordered = True.

6.2.2 Channel Access Probability

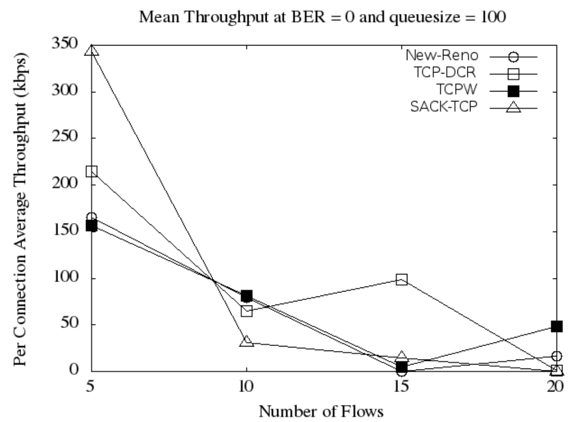
While routing the packets towards the destination, the abrupt path disconnection of mobile nodes leads to packet drop. Frequent path disconnection lowers the protocol performance. The flows with high packet drop rate due to route unavailability are eliminated from performance analysis.

$$\text{ChannelAccess Probability} = \frac{\text{Count of packets dropped due to route failure and MAC level failure}}{\text{Count of packets sent by source}} \quad (2)$$

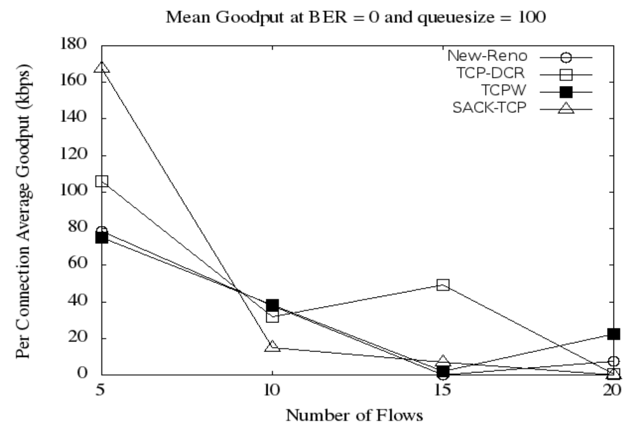
6.3 Simulation Results and Analysis

The simulation observations are summarized below.

Case 1: Experiments with low congestion loss and no error (queue size = 100, BER = 0, Gauss-Markov Mobility Model and Multi-path routing using MP- OLSR).

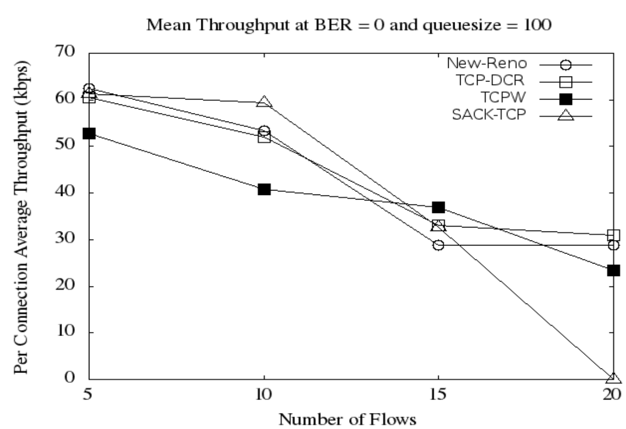


(a)

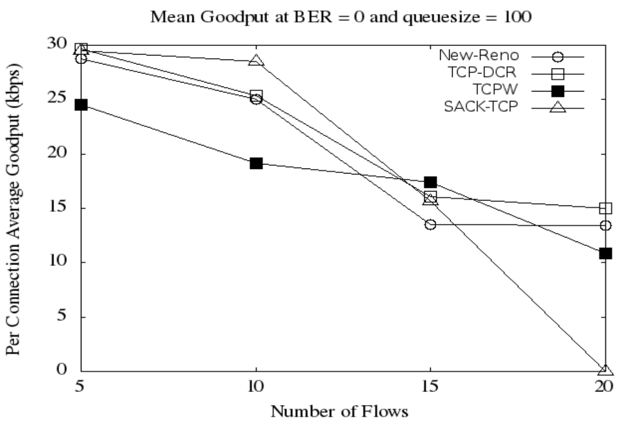


(b)

Figure 3. Performance of the connections with medium reordering with BER = 0 and Queue size = 100 packets. (a) Mean Throughput. (b) Mean Goodput.



(a)



(b)

Figure 4. Performance of the connections withhigh reordering at BER = 0 and Queue size = 100 packets. (a) Mean Throughput. (b) Mean Goodput.

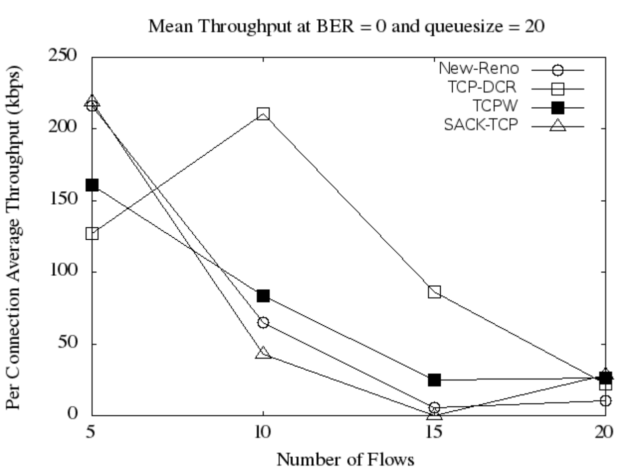
By setting queue size = 100 and BER = 0, the loss probability due to congestion and error becomes very less. In this experiment, we observed congestion probability of 0.5%, error loss probability of 0% out of total loss probability of 20%. In the remaining loss of 19.5 %, loss due to mobility and multi-path routing is 10%, which may contribute to reordering. Other 9.5% losses are due to the events like collision and exceeding retry count at MAC level.

Performance metrics of the connections with medium reordering and high reordering are plotted in Figure 3 and Figure 4. At medium reordering (maximum of 33%), TCP-DCR and SACK-TCP outperform than others. The high reordering threshold of TCP-DCR (i.e. number of duplicate ACKs > 3) and non-contiguous block reporting mechanism of SACK-TCP protect the sender to maintain the sending rate at same level or with less reduction in CWND. However, in high reordering (maximum of 55%), the late arrival of out-of-order packets triggers frequent timeout events in TCP-DCR and repeated unnecessary retransmissions in SACK. Hence both schemes fail to achieve the desired end-to-end performance.

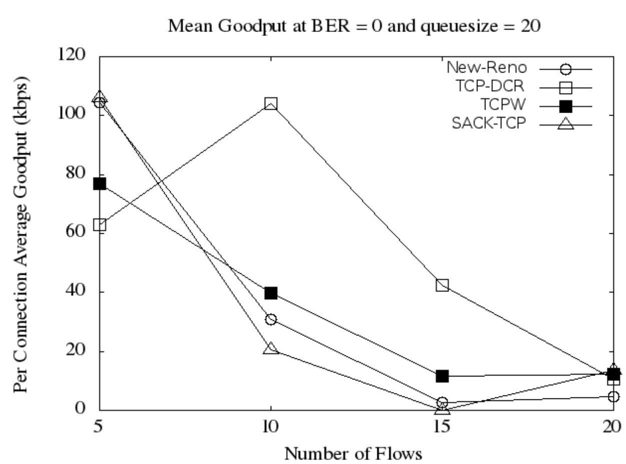
A small variation of 10kbps among the solutions is also observed. While increasing the number of flows as large as 20, all TCP variants failed to meet up their performance due to over network traffic (bottleneck link) at nodes and due to frequent path disconnection and route fluctuations.

Case 2: Experiments with high congestion loss and no error (queue size = 20, BER = 0, Gauss-Markov Mobility Model, Multi-path routing using MP-OLSR).

In this experiment, by setting the queue size of nodes as 20, the congestion loss probability increases to 10% out of total loss probability 20%. Even in the presence of congestion, the TCP-DCR performs well than others in medium reordering (reordering ratio maximum of 38%) shown in Figure 5 (a). The delayed congestion avoidance invocation procedure, the fast recovery procedure and the

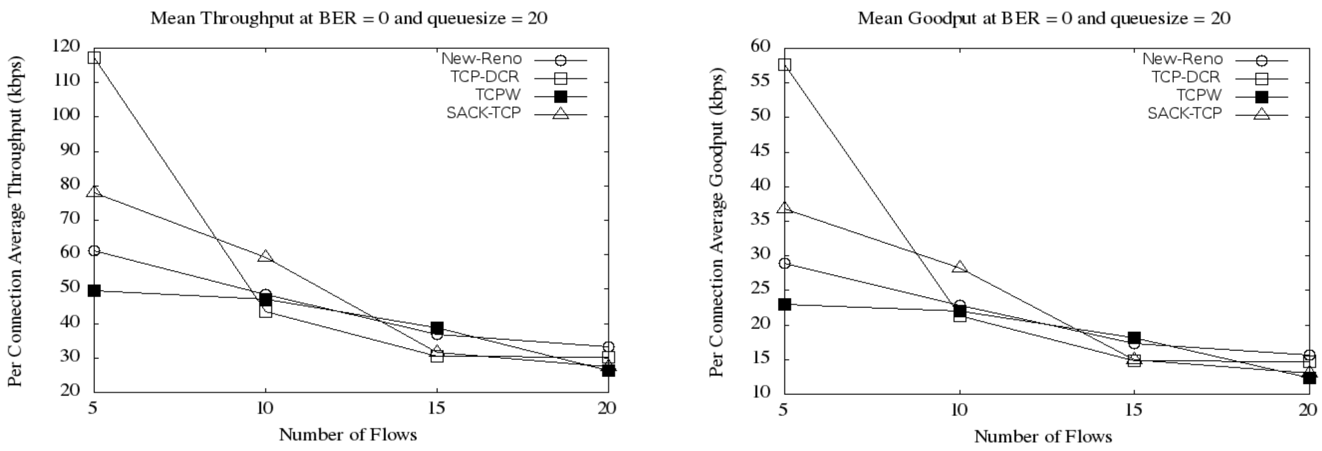


(a)



(b)

Figure 5. Performance of the connections with medium reordering at BER = 0 and Queue size = 20 packets. (a) Mean Throughput. (b) Mean Goodput.



(a) (b)

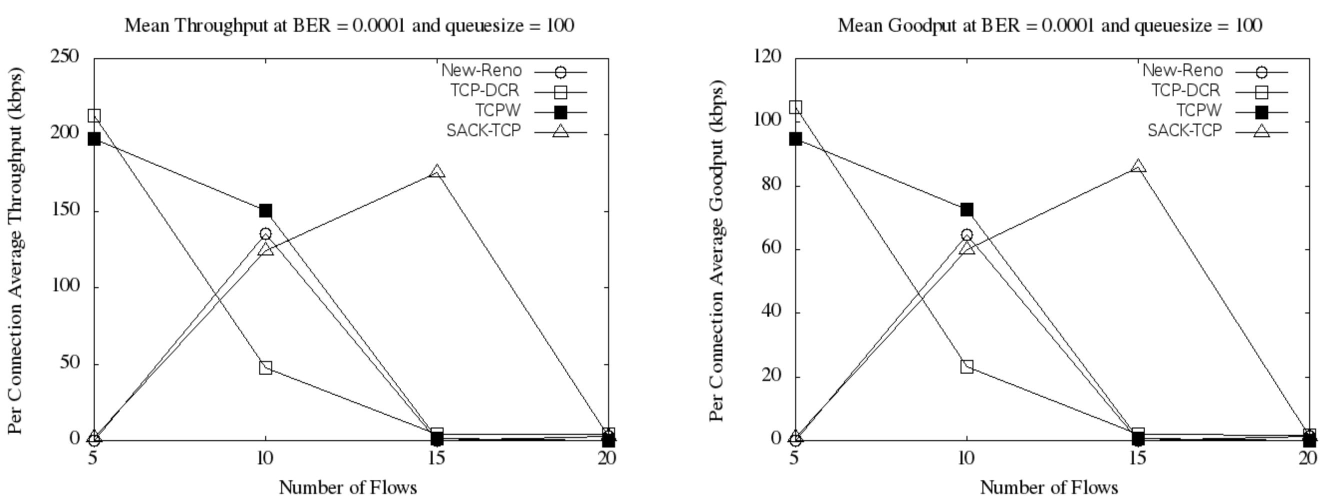
Figure 6. Performance of the connections with high reordering at BER = 0 and Queue size = 20 packets. (a) Mean Throughput. (b) Mean Goodput.

multiple decrease procedure of basic TCP in the case of high congestion provide an adaptive nature to the TCP-DCR. Since SACK is maintaining sender in congestion avoidance, it acts as aggressive mechanism in the high congestion.

Even though TCP-DCR provides a high end-to-end performance in high congestion and medium reordering, it achieves low performance like other TCP variants in high reordering (reordering ratio maximum of 55%).

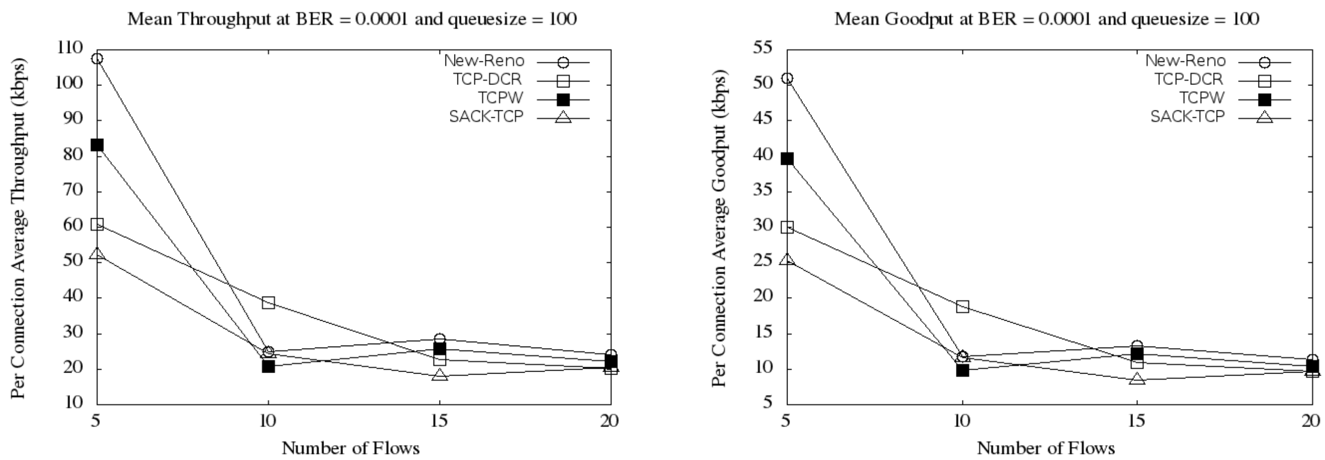
Case 3: Experiments with low channel error loss and low error (queue size = 100, BER = 0.0001, Gauss-Markov Mobility Model, Multi-path routing using MP-OLSR).

Over wireless medium, channel errors results in frequent random packet losses in the network. In presence of low channel error loss, most of connections get affected by reordering. Hence, the medium reordering, the ratio is increased to 45% (compared to 33% in Case1). In high reordering, the ratio remains same.



(a) (b)

Figure 7. Performance of the connections with medium reordering at BER = 0.0001 and Queue size = 100 packets. (a) Mean Throughput. (b) Mean Goodput.



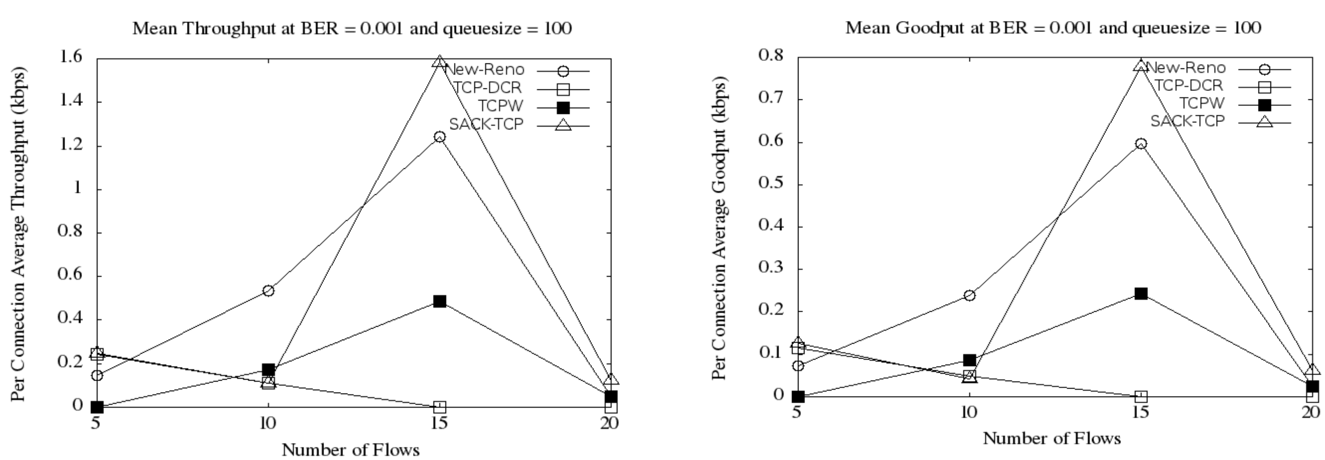
(a) (b)

Figure 8. Performance of the connections with high reordering at BER = 0.0001 and Queue size = 100 packets. (a) Mean Throughput. (b) Mean Goodput.

In medium reordering (Figure 7.), SACK-TCP yields high performance compared to other variants and sustains up to number of flows equals to 15. When the number of flows equals to 20, all variants are yielding very low performance. In contrast, TCP-DCR and TCP-W yields very high performance only in less number of flows. In high reordering (Figure 8.), all TCP variants behave similar and yields high performance loss.

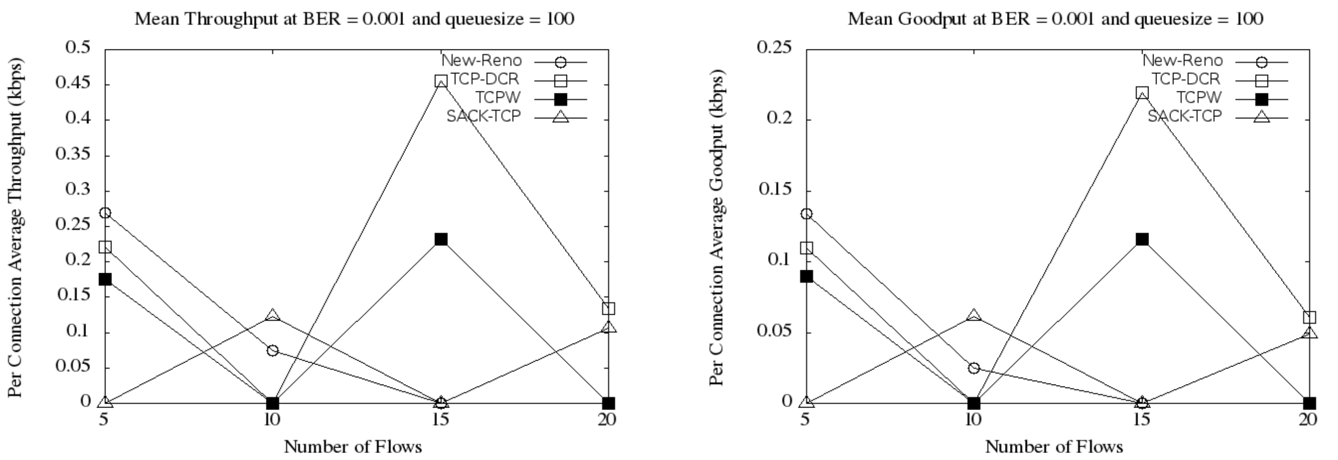
Case 4: Experiments with low congestion and high channel error loss (queue size = 100, Channel error rate = 0.001, Gauss-Markov Mobility Model, Multi-path routing using MP-OLSR).

In the presence of high channel error loss, in case of medium reordering the ratio maximum of 33% (in case 2) has been increased to 48%. Figure 9 shows that in medium reordering, compared to other variants SACK-TCP has higher achieved throughput and goodput. However, numerically it yields negligible performance i.e. the variation between SACK-TCP and other variants are very less. Similarly, Figure 10 shows that DCR has higher



(a) (b)

Figure 9. Performance of the connections with medium reordering at BER = 0.001 and Queue size = 100 packets. (a) Mean Throughput. (b) Mean Goodput.



(a) (b)

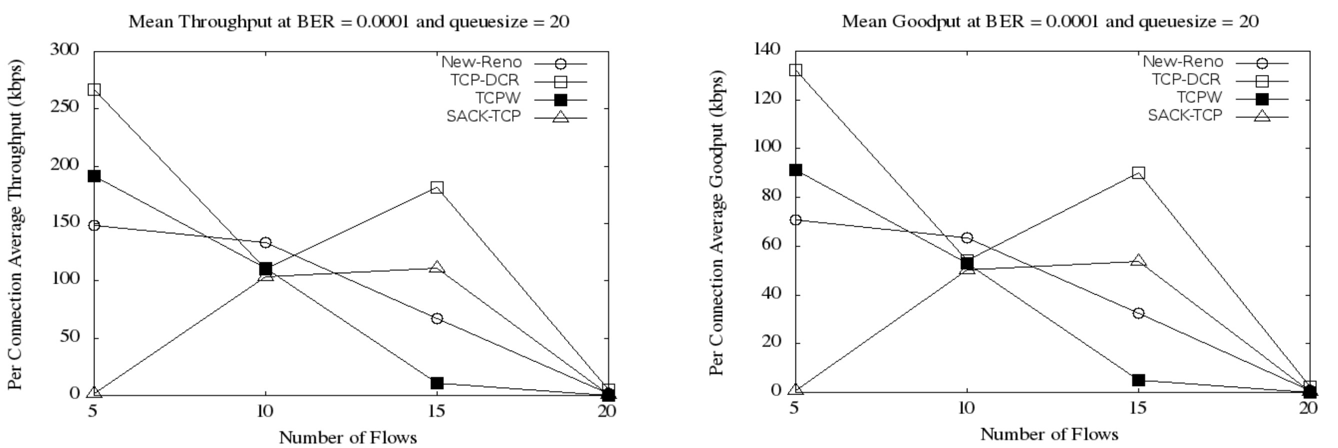
Figure 10. Performance of the connections with high reordering at BER = 0.001 and Queue size = 100 packets. (a) Mean Throughput. (b) Mean Goodput.

throughput and, goodput and they yield negligible performance i.e. the variation between TCP-DCR and other variants are very less.

The high transmission error leads to heavy packet reordering and engender more duplicate packets by spurious retransmission frequently. Frequent congestion response leads to performance degradation in all variants.

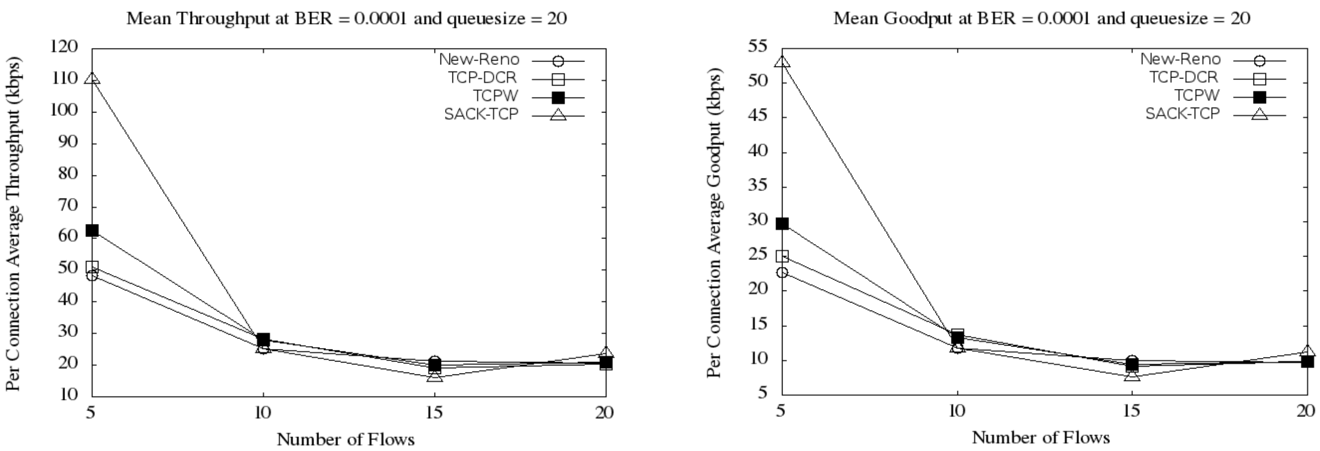
Case 5: Experiments with high congestion and low channel error loss (queue size = 20, Channel error rate = 0.0001, Gauss-Markov Mobility Model, Multi-path routing using MP-OLSR).

When comparing all variants in the presence of low error rate (BER = 0.0001) and high congestion, in the case of medium reordering, TCP-DCR and SACK-TCP perform better than TCP-New-Reno and TCPW. The packet loss is the effect of high congestion and low error and as a result, it invokes congestion action unnecessarily. Hence in high reordering, all TCP variants (TCP-SACK, TCPW, and TCP-New Reno) yield poor performance.



(a) (b)

Figure 11. Performance of the connections with medium reordering at BER = 0.0001 and Queue size = 20 packets. (a) Mean Throughput. (b) Mean Goodput.



(a) (b)

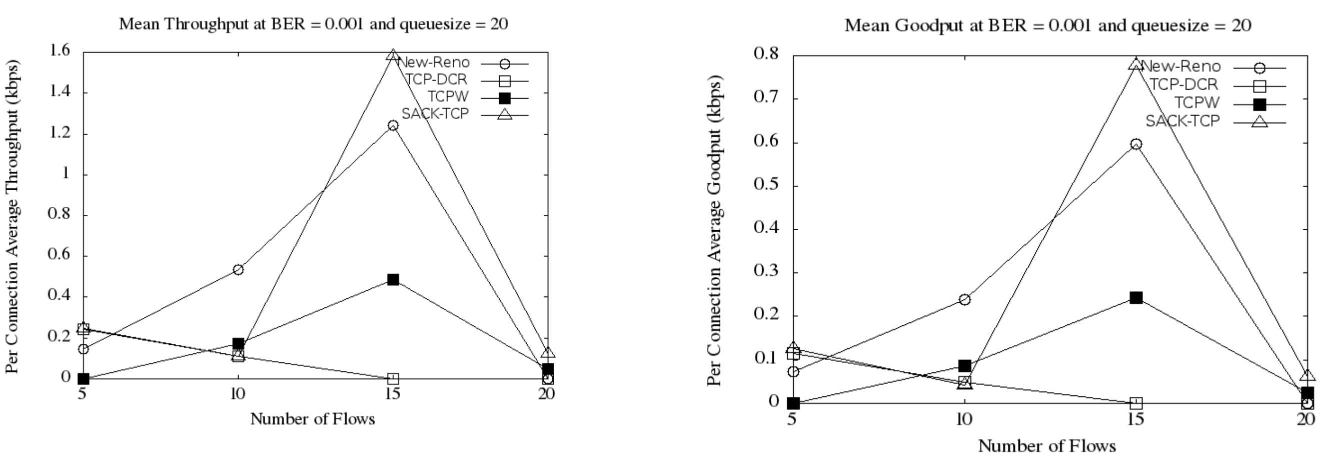
Figure 12. Performance of the connections with high reordering at BER = 0.0001 and Queue size = 20 packets. (a) Mean Throughput. (b) Mean Goodput.

Case 6: Experiments with high congestion and high channel error loss (queue size = 20, Channel error rate = 0.001, Gauss-Markov Mobility Model, Multi-path routing using MPOLSR).

In the presence of higher error rate and high congestion, most of connections faced the severe packet reordering i.e. most of the connections fall under high reordering level and a very few flows fall under medium reordering. At medium reordering, the SACK-TCP performs better than others and its performance variation with others is negligible. In high reordering, all TCP variants yield low throughput Figure 13 (a) and low goodput Figure 13 (b) within range 0 to 0.25kbps. In this case of severe reordering due to congestion, high channel error and multipath routing, all variants failed to meet up their performance.

Figure 14. Performance of the connections with high reordering at BER = 0.001 and Queue size = 20 packets.

From all our previous analysis, the variants are compared in high and medium reordering separately. Hence to visualize the performance of variants both in medium



(a) (b)

Figure 13. Performance of the connections with medium reordering at BER = 0.001 and Queue size = 20 packets. (a) Mean Throughput. (b) Mean Goodput.

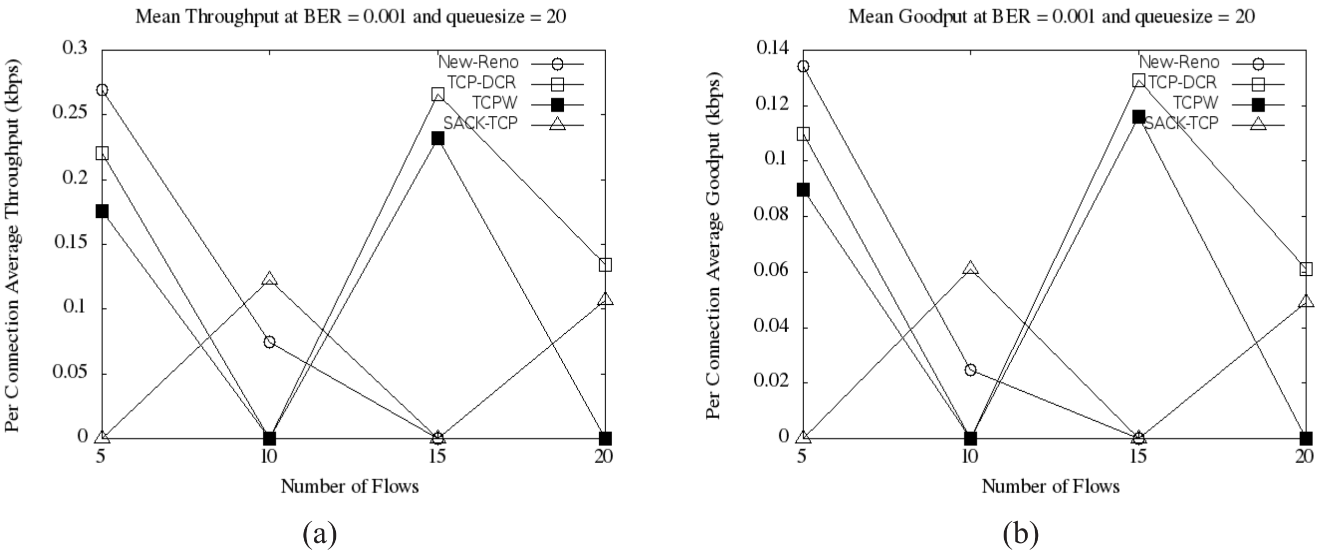


Figure 14. Performance of the connections with high reordering at BER = 0.001 and Queue size = 20 packets. (a) Mean Throughput. (b) Mean Goodput.

and high reordering level, the results of case 5 and case 6 are summarized as shown in Figure 15 and Figure 16. Since case 5 and case 6 are the worst case scenarios, other cases are excluded in this summary. Following are the few important observations from this summary.

- In high congestion and low channel error (case 5), the connections with medium reordering (20% to 39%) achieves high performance with the help of delay procedure in TCP-DCR and in high congestion and high channel error (case 6), the connections with medium reordering (20% to 39%) achieves high performance with the help of Selective Acknowledgement procedure in TCP-SACK.
- Both in case 5 and case 6, all variants fail to yield high performance to the connections with high reordering (40% to 60%).

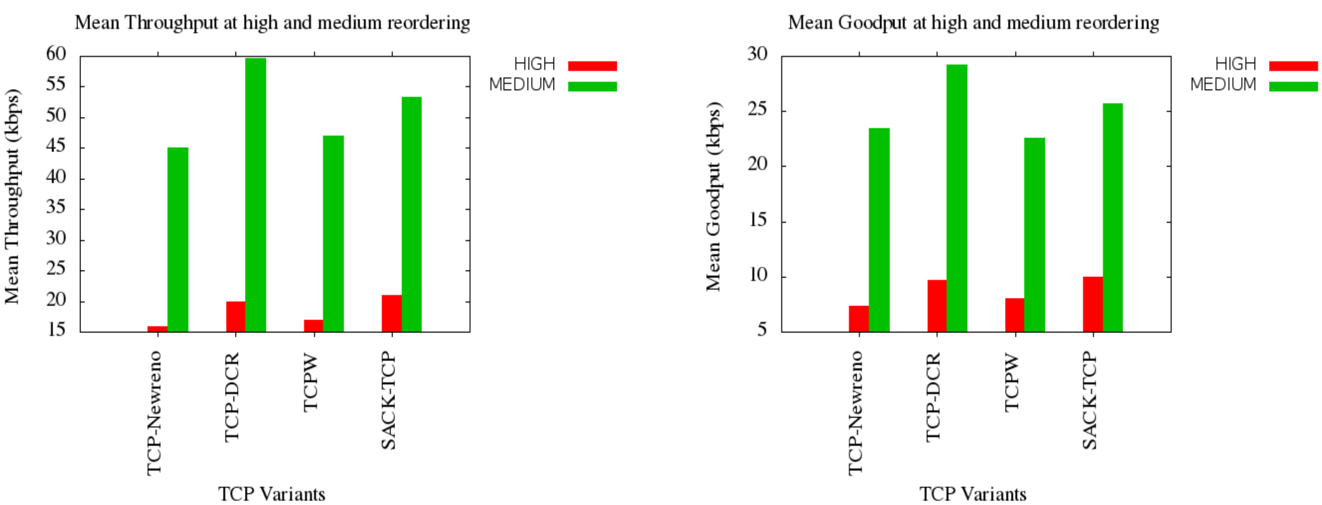
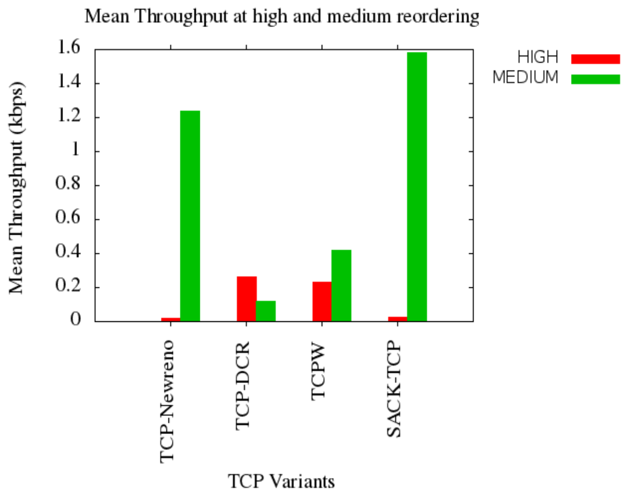


Figure 15. Performance of TCP variants in case 5.



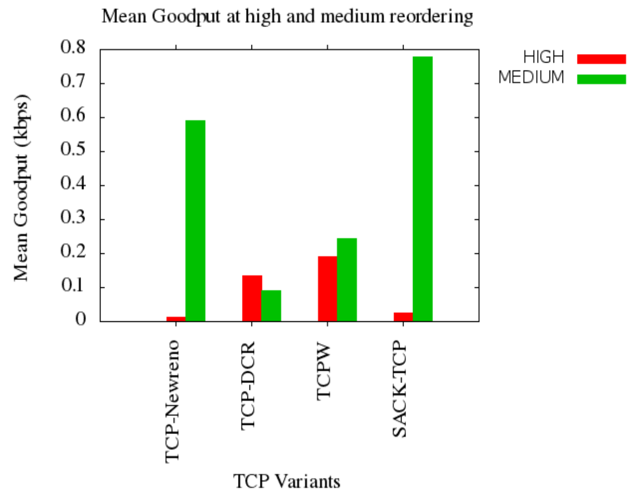


Figure 16. Performance of TCP variants in case 6.

7. Conclusion

Packet reordering occurs implicitly due to frequent retransmission of lost packets. The problem of receiving out of sequence packets is inevitable over an ad-hoc network. From our simulation we analysed the performance trade-off between the TCP variants (TCP-DCR, SACK-TCP, TCPW, and TCP-New Reno) over severe packet reordering. In high reordering, the variants perceive the packet reordering events as loss which invokes frequent unnecessary congestion action. This leads to performance degradation. In future TCP-DCR and SACK can be extending with discrimination procedure to avoid the false time out events.

8. References

1. Postel J. Transmission control protocol Request for Comments, RFC 793, Protocol Specification. DARPA Internet Program. 1981 Sep.
2. Mellian M, Meo M, Muscariello L, Rossi D. Passive analysis of TCP anomalies. Computer Networks. 2008 Oct; 52(14):2663–76.
3. Leung K-C, Li VOK, Yang D. An overview of packet reordering in Transmission Control Protocol (TCP): Problems, solutions, and challenges. IEEE Transactions on Parallel and Distributed Systems. 2007Apr; 18(4):522–35.
4. Balakrishnan H, Padmanabhan VN, Seshan S, Katz RH. A comparison of mechanism for improving TCP performance over wireless links. IEEE/ACM Transactions on Networking. 1997, Dec; 5(6):756–69.
5. Hu F, Sharma N. Enhancing wireless internet performance. IEEE Communications Surveys and Tutorials. 2002 Dec; 4(1):2–15.
6. Ludwig R, Katz RH. The Eifel Algorithm: Making TCP Robust against spurious retransmissions. ACM SIGCOMM Computer Comm Rev. 2000 Jan; 30(1):30–6.
7. Zhang M, Karp B, Floyd S, Peterson L. RR-TCP: A reordering-robust TCP with DSACK. Proc IEEE Int'l Conf Network Protocols (ICNP'03); 2003 Nov. p. 96–106.
8. Blanton E, Allman M. On making TCP more robust to packet reordering. ACM SIGCOMM Computer Comm Rev. 2002 Jan; 32(1):20–30.
9. Leung K-C, Ma C. Enhancing TCP performance to persistent packet reordering. Journal of Communications and Networks. 2005 Sep; 7(3):385–93.
10. Lee Y, Park I, Choi Y. Improving TCP performance in multipath packet forwarding networks. Journal Communications and Networks. 2002 Jun; 4(2):148–57.
11. Bhandarkar S, Reddy ALN. TCP-DCR: Making TCP robust to non-congestion events. Lecture Notes in Computer Science. 2004 May; 3042:712–24.
12. Fall K, Floyd S. Simulation-based comparisons of Tahoe, Reno, and SACK TCP. ACM SIGCOMM Computer Comm Rev. 1996 Jul; 26(3):5–21.
13. Yang D, Leung K-C, Li VOK. Simulation-based comparisons of solutions for TCP packet reordering in wireless networks. Proceedings of IEEE WCNC 2007; Kowloon, China. 2007 Mar. p. 3240–5.
14. Casetti C, Gerla M, Mascolo S, Sanadidi MY, Wang R. TCP Westwood: End-to-End congestion control for wired/wireless networks. Wireless Networks. 2002 Sep; 8(5):467–79.

15. Karn P, Partridge C. Improving round-trip time estimates in reliable transport protocols. ACM Transaction and Computer Systems. 1991 Nov; 9(4):364–73.
16. Camp T, Boleng J, Davies V. A survey of mobility models for ad hoc network research. Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad-Hoc Networking. Research, Trends and Applications. 2002; 2(5):483–502.
17. Ariyakhajorn J, Wannawilai P, Sathitwiriyawong C. A comparative study of random waypoint and Gauss-Markov mobility models in the performance evaluation of MANET, Communications and Information Technologies. 2006 International Symposium on Communications and Information Technologies (ISCIT'06); Bangkok. 2006. p. 894–9.
18. Yi J, Cizeron E, Hamma S, Parrein B. Simulation and performance analysis of MP-OLSR for mobile ad hoc networks. Proceedings of IEEE Wireless Communications and Networking Conference (WCNC); Las Vegas, NV. 2008. p. 2235–40.
19. Bonnmotion Scenario generator. 2014; Available from: http://sys.cs.uos.de/bonnmotion
20. Morton A, Ciavattone L, Ramachandran G, Shalunov S, Perser J. Packet Reordering Metrics (RFC 4737). 2006 Nov.
21. Jayasumana A, Piratla N, Banka T, Bare A, Whitner R. Improved Packet Reordering Metrics (RFC 5236). 2008 Jun.
22. Anbar M, Ramadass S, Manickam S, Al-Wardi A. Connection failure message-based approach for detecting sequential and random TCP scanning. Indian Journal of Science and Technology. 2014, May; 7(5):628–36.
23. Akhtar MAK, Sahoo G. Behavior based high performance protocol for MANET, Indian Journal of Science and Technology. 2013 Oct; 6(10):5342–50.