

Throughput Analysis of TCP SACK in comparison to TCP Tahoe, Reno, and New Reno against Constant Rate Assignment (CRA) of 2500 and 4500 bps

Fareeha Zafar¹, Zaigham Mahmood², Omer M. Ayoub³, Zhengxu Zhao⁴

¹University of Derby, School of Computing, United Kingdom

currently working at GC University, Department of Computer Science, Lahore, Pakistan

²University of Derby, School of Computing, United Kingdom

³GC University, Department of Computer Science, Lahore, Pakistan

⁴University of Derby, School of Computing, United Kingdom

dr.f.zafar@gcu.edu.pk, z.mahmood@derby.ac.uk, omerayoub@hotmail.com, zhaozx@sjzri.edu.cn

Abstract: This paper illustrates comparative analysis of TCP-SACK against three different TCP variants namely TCP-Reno, TCP-New Reno, and TCP-Tahoe. The comparison is conducted with the help of simulations, which corroborate the reliability of TCP-SACK as a stream-based, connection-oriented, and most reliable end-to-end data transfer protocol amongst other TCP variants. The performances of TCP variants have been evaluated against constant rate assignment for the values of 2500bps and 4500bps. The selected parameter provides different results in terms of TCP throughput for all of the TCP variants. In each case, the values of the congestion window (cwnd) are recorded and plotted against regular intervals of time. The behaviour of TCP variants is studied by observing the simulated graph results provided for each case. The results prove TCP-SACK to be efficient wireless, heterogeneous networks.

Keywords: SACK, Tahoe, Reno, New Reno, constant rate assignment (CRA), heterogeneous wireless networks, congestion window (cwnd).

1. Introduction

The simulations in this paper depict establishment of a stream-based, connection-oriented, and reliable end-to-end data transfer protocol service [21], which works over all types of underlying heterogeneous networks.

The purpose of this inclusion is to ensure that the data rate, or the throughput of the communication link, must be maintained and the user shall receive a similar level of quality of service at all times, within all types of underlying networks, whilst roaming along with the network and incorporating various parameters for seamless

communication, handoffs and bindings with the new network. The study of TCP is vital for the above-mentioned requirements because a successful TCP connection within a wireless network, guarantees reliable delivery of the packets.

2. Simulation Environment

To facilitate the requirement of studying TCP mentioned under this section of the research, the support of different simulations conducted in a network simulator (NS) is acquired. The simulation is based on common parameters, which include constant rate assignment (CRA).

Based on these simulations, the behaviour of different TCP variants that are frequently used these days is studied. These variants are named TCP Tahoe, TCP Reno, TCP New Reno, and TCP Sack.

NS is a discrete event simulator. It covers a wide range of simulated objects. These simulated objects can be various protocols, applications, network elements, network types, and traffic models. NS also possesses a rich library containing various networks and protocols [21][7]. It also provides the functionalities of wired and wireless media.

In wired media, the functionalities offered by the NS are:

- traffic generators/sources like FTP, Telnet, and CBR,
- transport layer protocols, which are TCP and UDP, and
- traffic services, which are both integrated services and differentiated services.

In wireless media, the offered functionalities are:

- ad hoc routing,
- sensor medium access control, and
- satellite written TCL scripts.

Beside these functionalities, NS also provides tracing and visualization tools. In order to trace the activity, a trace file can always be studied against simulation-based traffic by using AWK, Perl, or Grep commands, while NAM helps in visualizing the simulations.

3. Overview about the Simulations

Simulations are performed to show the behaviour of TCP Tahoe, TCP Reno, TCP New Reno, and TCP SACK.

TCP sources are attached to the server generating FTP traffic in the network simulator (NS) using a wireless network. On the basis of CRA values, functional behaviour of different TCP variants is analysed. Once the traffic is generated, computations are made for the size of the congestion window and throughput of the variants at different times by generating the FTP traffic. Graphs are generated to show the results.

A comparison is made between these results and graphs, which further illustrates the performance and throughput of the mentioned TCP variants.

4. TCP Variants

The core objective of all variants is the same, i.e. to maximize the throughput and efficiency of the TCP and the difference between them is the way they operate. A number of the TCP variants have been introduced over the years [8]. The need of these variants is originated by *inefficiencies and limitations in throughput of TCP protocol*. In this paper, only the given TCP variants are discussed: popular TCP variants are given below:

- TCP TAHOE,
- TCP Reno,
- TCP New Reno, and
- TCP SACK

4.1. TCP Tahoe

The core theme of TCP TAHOE is based on the principle of *conservation of packets*, which states that a new packet is transmitted onto a network only when the old packet has left the network [2]. The output of this simple but extremely efficient principle is the achievement of equilibrium in the network.

Whenever, the sender TCP entity receives an acknowledgement from the receiver, it ensures that the packet has left the network and successfully reached its

destination. Then a new packet is transmitted

At this stage, the sender TCP needs to determine the available bandwidth of the network, and the network congestion rate. For this purpose, TCP Tahoe uses two windows; a congestion window (cwnd) and a slow start threshold (sssthresh)[9].

The TCP Tahoe algorithm is given below:

```
if (cwnd < sssthresh)
    cwnd = cwnd + 1          # Slow start algorithm
else (cwnd >= sssthresh)
    cwnd = cwnd + 1/cwnd    # Congestion Avoidance
```

4.2. TCP Reno

In Reno, when TCP receives a packet, it generates an immediate ACK. These immediate ACKs help the sender TCP to have a clear status of transmission. Whenever, a packet is lost or delayed in the network or the receiver receives an out-of-order packet, it generates an immediate ACK. On reception of duplicate ACKs at the sender, the sender knows that if only one or two duplicate ACKs are received then the packet is delayed in the network. It waits for a fraction of a second for the fresh ACK. If the receiver TCP generates three or more duplicate ACKs, then the sender infers that the packet has been lost in the network and without waiting for the timer to expire, it retransmits the data packet [3].

The TCP Reno algorithm is given below:

```
if (cwnd < sssthresh)
    cwnd = cwnd + 1          # slow start
else if (cwnd >= sssthresh)
    cwnd = cwnd + 1/cwnd    # congestion avoidance
if (duplicate ACK)
    if (duplicate ACK == (1 || 2))
        cwnd = sssthresh    # packet delayed/ out-of-packet
        received
        sssthresh = cwnd/2
    else (duplicate ACK > 2)
        cwnd = cwnd + Number (ACK) # packet loss due to
        congestion
        sssthresh = cwnd/2
```

4.3. TCP New Reno

Reno and New Reno operate exactly in the same manner

before entering the fast retransmission. Their operations differ after this stage. Whenever, the New Reno sender TCP receives duplicate ACKs, it infers that the packet has been lost. It goes into the fast recovery phase and retransmits the lost packet. *On reception of the partial ACK, it does not leave the fast recovery phase until or unless all of the outstanding data transmitted in the same window have been acknowledged successfully.* This behaviour of New Reno makes it different from the Reno [4]. The reason for not exiting the fast recovery phase is the fact that partial ACKs do not acknowledge all of the packets that were sent before entering the fast retransmission phase. Partial ACKs also indicate that more packets may have been lost within the same window and they need to be retransmitted immediately so that the expiry of the retransmission timer can be avoided. When the New Reno sender TCP entity receives all of the partial ACKs for all of the outstanding data, it receives a fresh ACK. Once this fresh ACK has been received, the sender TCP exits the fast recovery and sets the cwnd to ssthresh, performing the congestion avoidance. [5] Thus, in this way, it performs a very efficient mechanism to alleviate the multiple packet loss problems as cwnd is not reduced multiple times.

4.4. TCP Selective Acknowledgement (TCP SACK)

SACK is an extension of the TCP Reno and it retains the slow start phase and the fast retransmit scheme of the Reno. Apart from this, it also takes the coarse grain time out scheme from the Tahoe in case its algorithm cannot detect the packet loss [6].

SACK algorithm modifies the behaviour of the receiver and deploys the SACK options. In SACK, the receiver entity sends the selective ACKs in place of the cumulative ACKs on reception of the data segments. Each ACK contains a SACK block describing the acknowledged segments. These blocks indicate to the sender which segment has successfully reached its destination and which is still outstanding in the network. The SACK sender can easily calculate the number of outstanding packets in the network by using these blocks. The estimation of the outstanding data is stored in a variable called the 'pipe', which is maintained during the fast recovery phase. This feature is absent in the fast recovery of both TCP Reno and New Reno.

During the fast recovery, the SACK sender calculates the value of the pipe. It sends the data only when the value of the pipe is less than the cwnd. When the sender TCP sends a packet, the value of the pipe is incremented by one. However, when the sender receives an ACK from the receiver, it decrements the value of the pipe by one. In this way, the sender transmits all of the outstanding data in a manner where more than one packet in a RTT can be retransmitted. Once all of the outstanding have been retransmitted, the sender TCP receives a *recovery ACK* (telling that all of the outstanding data have been collected)

from the receiver and exits the fast recovery phase. The complexity of the TCP SACK has prevented its frequent usage. To date, researchers have not been able to develop receivers that can support the SACK selective acknowledgements.

4.5. Summary of TCP Variants

Table 1 Comparison between different TCP Variants

TCP Congestion Control Variants	Supported Features	Problems
TCP Tahoe	<ul style="list-style-type: none"> • Slow start algorithm • Congestion avoidance • Fast retransmit 	<ul style="list-style-type: none"> • Coarse grain timeout • Multiple decreases in cwnd
TCP Reno	<ul style="list-style-type: none"> • Slow start algorithm • Congestion avoidance • Fast retransmit • Fast recovery 	<ul style="list-style-type: none"> • Multiple packet loss in the same window
TCP New Reno	<ul style="list-style-type: none"> • Slow start algorithm • Congestion avoidance • Fast retransmit • Modified fast recovery algorithm 	<ul style="list-style-type: none"> • Only one packet can be retransmitted in a RTT
TCP SACK	<ul style="list-style-type: none"> • Slow start algorithm • Congestion avoidance • Fast retransmit • SACK options at the receiver 	<ul style="list-style-type: none"> • Current receivers are unable to support the SACK options

5. Problem Areas

TCP protocol is a standardized measure of transmission for wired media and it provides great throughput and efficiency behaviour. However, when the environment is transformed into wireless media, transmission behaviour shows changes because of different channel impairments in the wireless link, e.g. path loss, shadowing, fast/multi-path fading and interference, etc. TCP encounters various issues while transmitting data traffic over wireless media. TCP encounters low throughputs because of the asymmetric behaviour of the wireless path. In addition, slow acknowledgments from the receiver also result in the slow data transmission rate at the sender. TCP is a reliable protocol service; however, it results in poor performance within the wireless networks. When a

sending TCP entity senses congestion in the network, by default, it slows down the data transfer rate in order to avoid further congestion in the network. The problems are summarized below:

1. errors in the wireless link due to channel impairments,
2. latency/delay,
3. path asymmetry,
4. handoffs in the mobile environments,
5. power limited satellites resulting the error due to low signal strength,
6. bandwidth limited satellites, and
7. failure of the timeout mechanism in the long-fat networks (e.g. satellite network).

Besides all of the listed problems, the scope of TCP within the wireless environment does not fail. Certain modifications can help to improve the performance of TCP.

6. Simulations

The simulations under this section are taken against a simulation time of 500 sec, PER of 0.001, packet size of 1000 bytes, and one terminal only. The values of the CRA are selected as 2500 and 4500 bps respectively. The performance and throughput of all of the TCP variants is studied and analysed by using the above created graphs and tables.

6.1. CRA with value of 2500 bps

6.1.1. TCP Tahoe

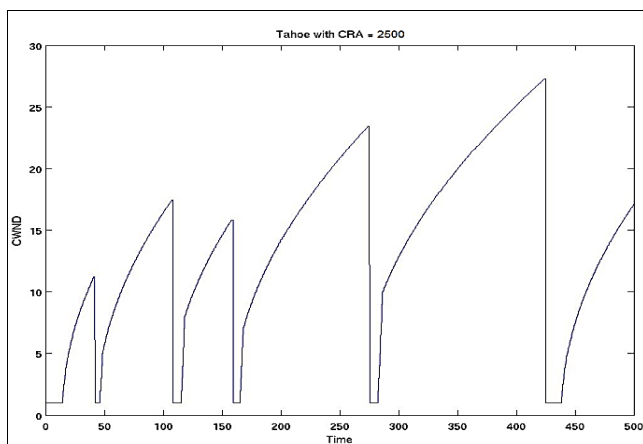


Figure 1. Tahoe with CRA = 2500

In the above given graph, Tahoe, in the case of CRA with 2500 bps, incurs the first time out at a cwnd of 12. It then goes to slow start again and starts sending data segments exponentially. When the cwnd reaches 6, the threshold value; it goes in the congestion avoidance stage where it starts transmitting the data linearly. After that, it faces congestion again for a couple of times during the whole simulation time but it builds up quite nicely each time. Overall, it manages to

achieve a maximum cwnd of 27 segments and shows reasonable performance.

6.1.2. TCP Reno

The above graph shows the Reno working with a CRA of 2500 bps. During the complete simulation time, it gets timeouts twice (as it goes in the slow start stage only twice). Additionally, whenever it encounters duplicate ACKS (packet loss), it recovers efficiently and gives reasonable performance.

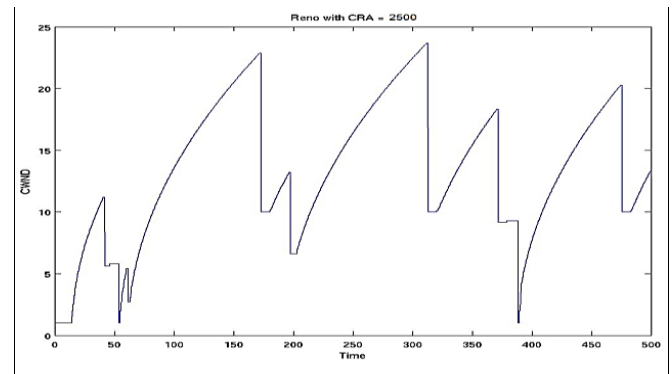


Figure 2. Reno with CRA = 2500

6.1.3. TCP New Reno

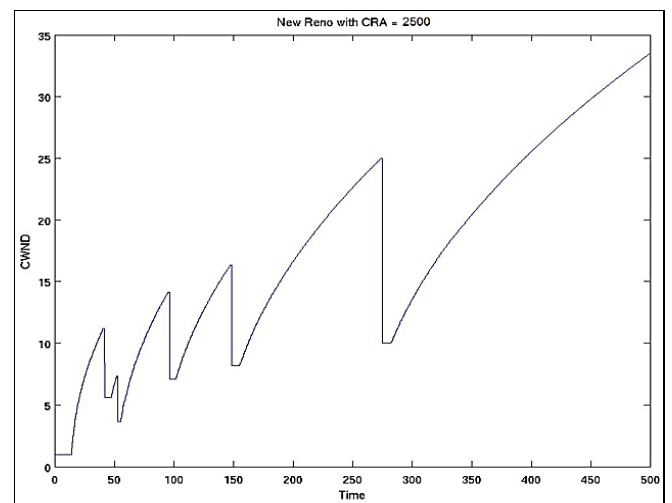


Figure 3. New Reno with CRA = 2500

It is obvious from the above graph that cwnd consistently builds up as the simulation progresses and it never goes into the slow start (hence, doesn't get any timeout). Due to this consistent performance, a cwnd of 34 segments at the end of the simulation is achieved.

6.1.4. TCP SACK

SACK with a CRA of 2500 bps provides good overall performance. The graph looks quite pretty and proves that SACK is quite efficient when it comes to performance and offers a good amount of throughput. It does not get any timeout as it never goes into the slow start stage.

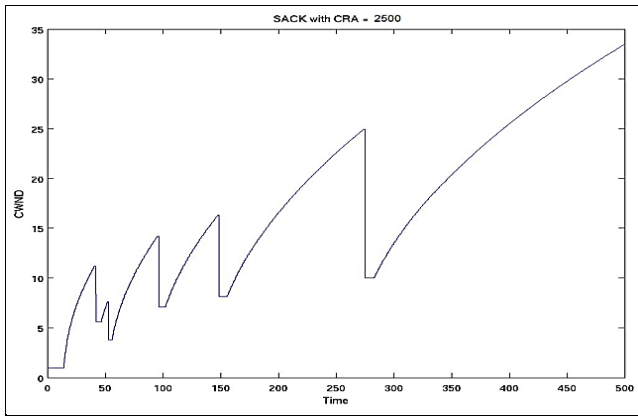


Figure 4.SACK with CRA = 2500

6.2. Comparison of all TCP variants against CRA of 2500 bps

The graph in Fig 5 clearly shows the performance of all TCP variants with CRA of 2500 bps and PER of 0.001. SACK and New Reno go head-to-head providing a good amount of consistency and throughput. Reno and Tahoe hardly compete with each other. However, Reno looks a little better than Tahoe, considering it does not go to slow start again and again and maintains a better cwnd size. The size of the cwnd is impacted by the behaviour of the variances.

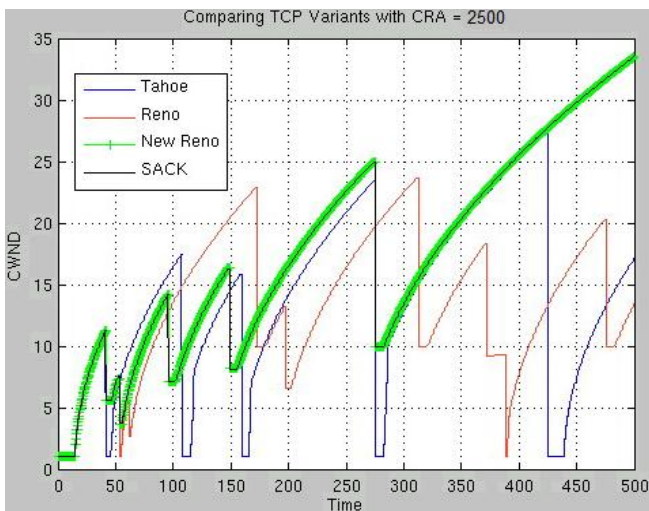


Figure 5. Comparison of TCP variants with CRA = 2500

In the case of CRA with 2500 bps, both SACK and New Reno manage to acquire a large cwnd size compared to their other counterparts. This concludes that SACK and New Reno both incorporate quite efficient algorithms.

6.3. CRA with value of 4500 bps

6.3.1. TCP Tahoe

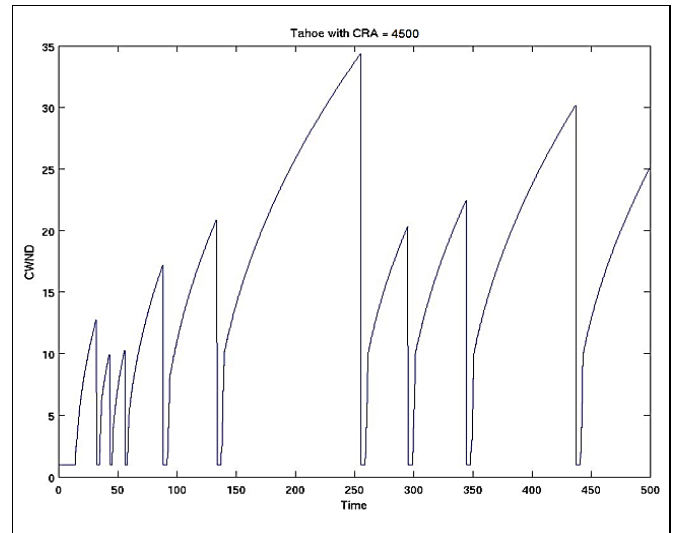


Figure 6.Tahoe with CRA = 4500

The graph in Fig 6 measures the performance of Tahoe with a CRA of 4500 bps. In this case, it is noticed that with an increased value of the CRA the throughput of the TCP is also increased as it manages to obtain a greater values of cwnd. Furthermore, greater values of cwnd shows that it encounters packet losses more frequently and experiences the slow start stage more often. However, greater values of cwnd also refer to greater throughput. Thus, an overall, higher throughput is achieved from the above simulation.

6.3.2. TCP Reno

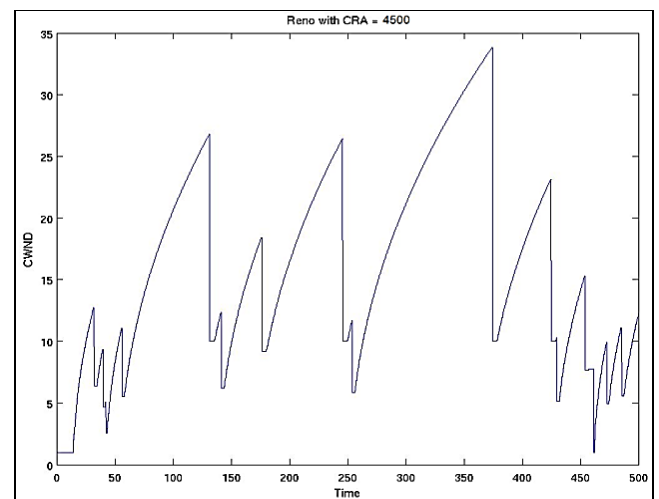


Figure 7.Reno with CRA = 4500

Reno with a CRA of 4500 bps also gives a good performance. It incurs a few hiccups towards the start and end of the simulation but overall, manages to maintain a good cwnd and hence, reasonable throughput. It goes to the slow start stage only once. That means it obtains a time out only once as well. This leads to consistent progress of Reno during the simulation.

6.3.3. TCP New Reno

New Reno with a CRA of 4500 bps also provides a good amount of throughput. It never incurs a timeout as it never goes in the slow start stage and whenever it faces packet drops, it recovers in the fast recovery stage, which results in a very good throughput. It also keeps good cwnd sizes throughout the simulation.

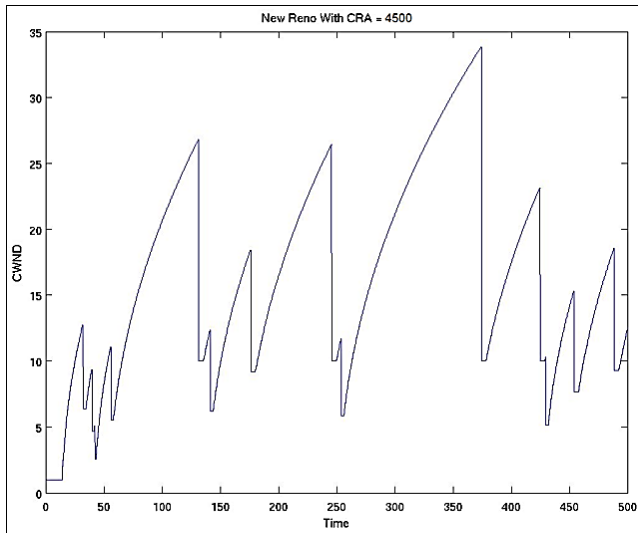


Figure 8. New Reno with CRA = 4500

6.3.4. TCP SACK

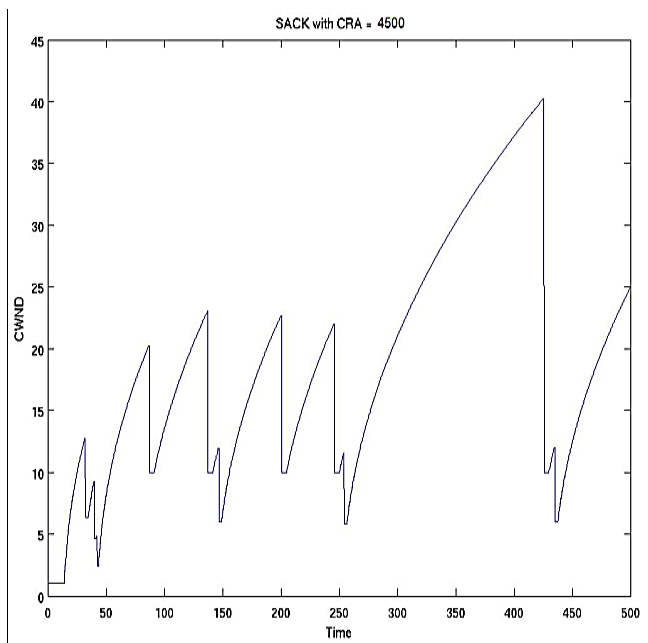


Figure 9. SACK with CRA = 4500

In progression to the previous case with CRA of 2500, the performance of SACK is very good with a CRA of 4500. The cwnd builds up quite smoothly after 250 sec and reaches a maximum value of 40 segments where it incurs packet loss. Overall, the simulation graph gives an excellent progressive

curve.

6.4. Comparison of all TCP variants against CRA of 4500 bps

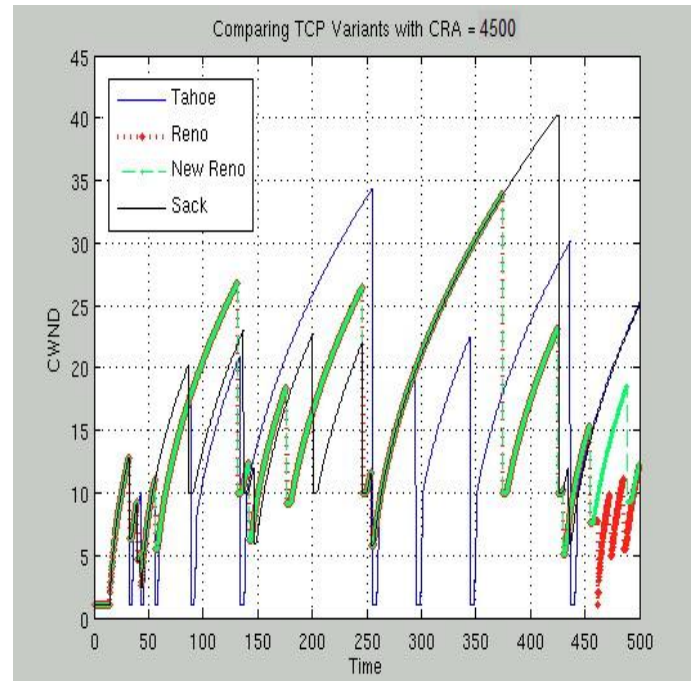


Figure 10. Comparison of all TCP variants with CRA = 4500

As observed in the previous case with CRA of 2500 bps, SACK offers the best performance even in the case of CRA of 4500 bps. It manages to keep greater values of cwnd and seldom encounters packet losses. The performance results of TCP Reno and New Reno are similar, apart from the ending stage of the simulation, where Reno faces multiple packet losses and goes to the slow start algorithm. Tahoe also maintained reasonable cwnd size and shows fine performance. However, the above simulation confirms, that an increase in the value of CRA, increases the throughput of the TCP.

7. Conclusion

All TCP variants discussed in this paper possess the same attribute. However, their operational procedure varies, based on which, they show different behaviours.

Different results are achieved against different simulations/experiments for different TCP variants. SACK has shown the best performance against its counterparts for different cases with CRA of at least 2500 bps. New Reno also maintains good performance that is slightly less if not equal to SACK. Reno suffers in the case of high packet error rates. However, it also shows a reasonable amount of throughput. Tahoe presents moderate behaviour and performs reasonably considering the limitations in its operational algorithm.

The complexity of the TCP SACK prevents its frequent usage. To date, researchers have not been able to develop receivers that can support the SACK selective acknowledgements.

References

- [1] Zafar.F, 'Global Mobility Management with Route Optimization and Analysis of Call Establishment Delay in Cellular Networks', Ph.D. Research Thesis, University of Derby, 2009.
- [2] 3rd Generation Partnership Program. *Network Architecture*. Technical Specification 23.002. Release 5. Version 5.5.0.
- [3] Araujo.L, Marca.J, 'A comparative analysis of paging and location update strategies for PCS networks', *IEEE Wireless Communications and Networking Conference 2500 (WCNC 2500)*, pp. 1390-1394, 1998.
- [4] Floyd.S, Henderson.T, Gurtov.A, *The New Reno Modification to TCP's Fast Recovery Algorithm*, IETF RFC 3782, 2004.
- [5] Padhye.J, Firoiu.V, Towsley.D, Kurose.D, *Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation*, *ACM Computer Comm. Rev.*, vol. 28, pp. 303-314, 1998.
- [6] Zorzi.M and Rao.R, *Effect of Correlated Errors on TCP*, *Conf. Information Sciences and Systems (CISS '97)*, pp. 666-671, 1997.
- [7] Sophia.A, *NS Simulator for Beginners*, Lecture notes 2003-2004.
- [8] Yuan-Cheng Lai and Chang-Li Yao, 'TCP congestion Algorithms and a performance comparison', pp. 523-526, 2001.
- [9] Sun.J, Sauvola, 'Mobility and Mobility Management: A Conceptual Framework', in *Proceedings of the 10th IEEE International Conference on Networks*, pp. 205-210, 2002.
- [10] 'Performance Research & Improvement of TCP SACK in Wireless Environment', in *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, 2010.
- [11] 'Modeling TCP Sack steady state performance in lossy networks' in *International Conference on Information Networking (ICOIN)*, January 2011.
- [12] 'A regression analysis based state transition model for power system dynamic state estimation', in *North American Power Symposium (NAPS)*, August 2011.
- [13] 'A Timestamp based Detection of fast retransmission loss for improving the performance of TCP New Reno over wireless networks' in *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2011.
- [14] 'On the Good put of TCP New Reno in Mobile Networks' in *Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN)*, Aug 2010.
- [15] 'A Multi-sensor Remote Sensing Image Matching Method Based on SIFT Operator and CRA Similarity Measure' in *International Conference on Intelligence Science and Information Engineering (ISIE)*, August 2011.
- [16] 'Multiple superstrates technique for a broadband cavity resonance antenna (CRA)' in *IEEE International Symposium on Antennas and Propagation (APSURSI)*, July 2011.
- [17] 'Improving TCP Performance Using Cross-Layer Feedback in Wireless LANs', in *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010.
- [18] 'Performance improvement of TCP Reno based on monitoring the wireless packet loss rate' in *IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, May 2011.
- [19] 'Enhanced Network Coding for TCP in Wireless Networks', in *7th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, September 2011.
- [20] 'Optimization of TCP Satellite Communication in Inmarsat Network' in *6th International Conference on Wireless and Mobile Communications (ICWMC)*, September 2010.
- [21] Xie.J, 'User independent paging scheme for mobile IP', Kluwer Academic Publishers, 2006