Middle-East Journal of Scientific Research 11 (5): 634-641, 2012
ISSN 1990-9233

A Novel TCP Scheme for Enhancing the Packet Reception Time over Wireless Networks

[1]*B.D. Deebak and* [2]*N.K. Sakthivel*

[1]Research Scholar, School of Computing, Sastra University, Thanjavur-613 401, India
[2]School of Computing, Sastra University, Thanjavur-613 401, India

Abstract: Over few decades, the Transmission Control Protocol (TCP) has been employed for developing the major Internet applications. Eventhough of which TCP has some major challenging issues when it operates in the Wireless Networks (WN). Some of the issues are identified from the existing techniques which are Packet Dropping Rate (PDR) and Packet Reception Time (PRT). The identified issues are mainly used to estimate the end-to-end performance of TCP which reveals poor result for existing TCP-Variants of TCP-Newreno which is found by the proposed technique of TCP-EPRT. TCP-EPRT is revealed the following results for its most estimable achievement when the existing and proposing techniques were comparatively analyzed. With the technique of Explicit Loss Notification (ELN), we were achieved higher throughput as compared to TCP-Newreno. The proposed work is thoroughly evaluated by Network Simulator Tool (Version NS2.34). The resultant of TCP-EPRT and TCP-Newreno shown that the PDR of TCP-EPRT is .38% and PDR of TCP-Newreno is .53%, the PRT of TCP-EPRT three-fourth time better than PRT of TCP-Newreno, the throughput of TCP-EPRT is 99.3% which is most effective than the throughput of TCP-Newreno which is 97.1% over Wireless Networks.

Key words: TCP · TCP-Newreno · TCP-EPRT · PDR · PRT · NS2.34

INTRODUCTION

For the past few years, TCP protocol has been employed for developing the internet applications. The congestion control algorithms of TCP are very essential for the stability of the Internet. However, the end-to-end throughput of TCP degrades rapidly when TCP operates in wireless networks [1, 2]. There are two main reasons for the degradation of TCP throughput over wireless networks. They are: 1) the unnecessary reduction of congestion window size caused by non-congestion packet losses and 2) the occurrence of frequent retransmission timeouts [3]. The consequence of often reduction of congestion windows will bring down the performance of end-to-end throughput, it will increase the percentage of PDR and it will achieve less PRT that was revealed in the study analysis of TCP-EPRT and TCP-Newreno. Because of network congestion, TCP loses several packets and the lost packets maybe belong to same/different windows. After retransmission timeout, the lost packets can be retransmitted to the TCP receiver. The time to reach the retransmission timeout will degrade the network performance. Also, by detection of Triple Duplicate Acknowledgement, the lost packet can be retransmitted to the TCP receiver. The size to increase the congestion window will take a long time to reach the destination. In most of the variants of TCP, the packet loss is treated as network congestion even if the packet loss was because of random loss or bit error rate [1]. When TCP sender detects packet loss through retransmission timeouts, it goes to slow start state and set the size of congestion window to one maximum segment size *(mss)*. Also, it takes many Round Trip Times *(RTTs)* to bring the transmission rate to the previous level. This leads to TCP to reduce the flow of packets drastically and thereby decrease the throughput of TCP dramatically [2, 4]. In [2, 5], the authors of N. J. Kothari, D.Ciullo et al. were shown that 70% of the dropped packets are recovered after the expiration of retransmission timeouts. According to technique of Detecting Avoidable Timeouts which was proposed by N. J. Kothari in [2], 14% of timeouts are caused by the loss of retransmitted packets. In Wireless Networks, a retransmission timeout is inevitable, when the retransmission of a lost packet fails

Corresponding Author: B.D. Deebak, Research Scholar, School of Computing, Sastra University, Thanjavur-613 401, India.

to reach the destination particularly due to changing level of congestion and bit error rate in wireless channel [6-8]. The aforementioned is evidently shown that the lost packet can be retransmitted only after the retransmission timeout or by the detection of triple duplicate acknowledgement that is implemented to be performed in all the variants of TCP when the packet loss occurs. Other than on the aforementioned, TCP variants do not have any other mechanism for retransmitting the loss of packets. By retransmission timeout, the network performance will be degraded that happens of network idle time and it will increase the percentage of PDR. By triple duplicate acknowledgement, the sender has to do window inflation that will reduce the percentage of PRT. Thus we realized that the TCP Variants are needed of some specific techniques which should be like Detecting and Differentiating the Loss of Retransmission detecting avoidable timeouts Loss Recovery Using Limited Transmit Delayed Ack Techniques that would be able to reduce the percentage of PDR and would enhance the percentage o f PRT. Thereby the performance of TCP would be achieved better than the existing variants of TCP over Wireless Networks. For which, this research work has proposed with an efficient technique called TCP-EPRT which is compared with the existing variants of TCP-Newreno for its comparative and performance analysis. From the results of performance simulation of TCP-EPRT and TCP-Newreno, we found that our proposed scheme would be able to increase the TCP performance by enhancing the percentage of PRT, reducing the percentage of PDR and above all improving the network throughput through the proposed technique of Explicit Loss Notification.

The other sections of the paper are described as follows: Section 2 concisely discusses about the Related Work. Section 3 elaborates the Proposed Work in which a new mechanism of TCP Variant, called TCP-EPRT is catered in clear. Section 4 demonstrates the performance analysis of TCP-EPRT. Eventually Section 5 concludes our work.

Related Work: In the section of related work we describe the Extensive research work which has been taken for achieving good throughput performance by reducing the packet dropping rate and enhancing the packet reception time over Wireless Network. The research study has found some of the algorithms and resolving techniques for retransmission timeout and triple duplicate acknowledgement by referring the various researches works [1, 9-13]. The following ideas are being come out of those research works and they are briefly discussed in this section.

TCP, Packet Dropping Rate (PDR), Packet Reception Time (PRT): The current implementation of TCP contains of four main congestion control algorithms such as Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery and the algorithms are aimed to control and recover packets from network congestion losses [13]. TCP is defined of two main variables: congestion window size (*cwnd*) and slow start threshold (*ssthresh*). The former variable is used to estimate the maximum number of segments that can be sent to the receiver without overloading the network and the latter variable is used to determine the state of a TCP sender [14].

TCP is deduced a packet loss based on the receipt of three duplicate acknowledgments or retransmission timeouts. When TCP transmits a segment, the sender starts a timer which keeps track of how long it takes for an acknowledgment (*ack*) for that segment to return. This timer is known as the Retransmission timer. If an *ack* is returned before the timer expires (by default is often initialized to 1.5 seconds), the timer is reset with no consequence. However, an *ack* for the segment does not return within the timeout period, the sender would retransmit the segment and double the retransmission timer value for each conservative timeout up to a maximum of about 64 seconds. When sender detects packet loss by the expiration of timeout, the sender retransmits the first unacknowledged segment and resets the size of *cwnd* to one *mss* and increases according to the slow start algorithm. When the sender detects packet loss via three duplicate acknowledgments (*dupacks*), it invokes fast retransmit algorithm immediately without waiting for the retransmission timeout, which sets the value of *ssthresh* to half of the size of *cwnd* and retransmits the missing segment. After retransmitting the missing segment, fast recovery algorithm takes over. In fast recovery, the value of *cwnd* sets to *ssthresh* plus three and increment by one for each additional *dupacks* received. The sender sends new packets allowed by the value of *cwnd*. When the sender receives a complete *ack*, *cwnd* resets its value to *ssthresh* and put the sender in congestion avoidance algorithm [1, 13].

The aforementioned is clearly shown the disadvantage of existing variants of TCP which have the mechanism of retransmission timeout and triple duplicate acknowledgement. By which the existing TCP variants

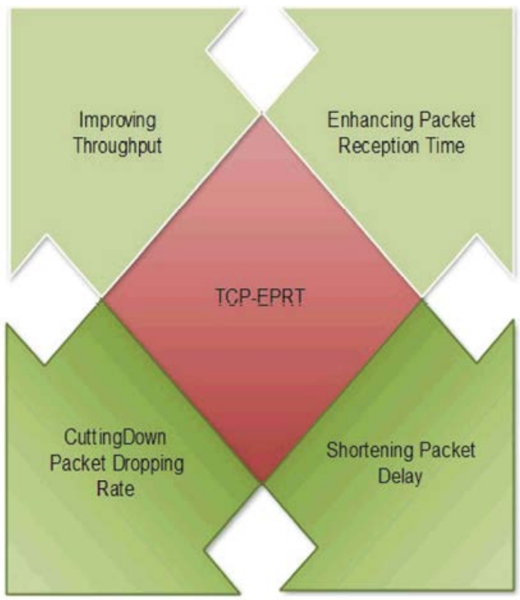

Fig. 1: Parameter Scheme of TCP-EPRT

would be degraded in the performance of the network. This would happen because of time reaching delay to retransmission timeout. In that duration the networks would become idle and no packet transmission would happen between TCP Sender and TCP Receiver. Thus, the performance of network would drop off and the packet reception time would become worst in TCP Receiver. By inflating the size of the window, the packet delay would be more when compared to the previous window size. The prolonging delay could lead the packet to be dropped. Hence it would experience the increasing packet dropping rate.

For enhancing the packet reception time and reducing the packet dropping rate, our proposed work would like to introduce the explicit loss notification in TCP Sender thereby the lost packet would be able to detect in on time for retransmission without waiting for retransmission timeout and triple duplicate acknowledgement as well. The proposed work outsmarted the existing TCP variant of TCP-Newreno by achieving estimable packet reception time and respectable packet dropping rate.

Fig 1 depicts the parameter scheme of TCP-EPRT. It has had of two upper layer schemes which are achieved by the deduction of packet loss before retransmission timeout. As well as it has had two lower layer schemes that is achieved by the retransmission of packet loss.

Surveying View of Other TCPs: The Limited Retransmit algorithm [9] is introduced, which proves that 25% of retransmission timeout based packet retransmission in [15] would have been avoided. But the problem of delay spikes still remains. E-RTO [16] authors were Michelle Berger and et al. proposed an algorithm on the basis of SACK option. It contributed to reduce the performance degradation of TCP in an erroneous wireless links. TCP NewReno [17] changes the fast retransmit algorithm for eliminating Reno's waiting time for the retransmission timeout when multiple segments are lost within a single window.

Although TCP NewReno can recover multiple packet losses without retransmission timeout but it has a problem that due to the large RTT variations, retransmission timeouts take place frequently. TCP Vegas [14] seeks to eliminate unnecessary expiration of the retransmission timer. It introduced a fine-grained timer to calculate RTT more precisely. When a single duplicate acknowledgment is returned to the sender, the sender retransmits a segment without waiting for the three dupacks, if the calculated RTT is greater than the timeout value. In [10], the authors Dongmin Kim, Beomjoon Kim and et al. proposed a solution for reducing retransmission timeout during fast recovery. EFR modified the fast recovery algorithm of TCP NewReno by updating the value of the variable *Recover*, when the sender retransmits a lost packet during fast recovery of TCP NewReno.

The TCP receiver with SACK options [18] can inform the sender which segments have been correctly received, when it hold non-contiguous segments and thereby reduce the retransmission timeout and unnecessary retransmissions. An increase of the initial *cwnd* size is advocated in [6] through an increase in the permitted upper bound for the initial window from one segment to between two and four segments, without changing the segment size. This approach already implemented in some OSs and can help avoid retransmission timeout in the initial window, but could potentially increase the network congestion. The above approaches do not have a mechanism to reduce retransmission timeouts cause by the loss of retransmitted packets [1, 2, 13].

Moreover, many loss differentiating algorithms proposed for wireless networks for improving the end-to-end throughput of TCP. Among that, the packet marking scheme of TCP NewJersey [19] can effectively differentiate packet losses caused by network congestion from those caused by wireless link errors. TCP Veno [12] monitors the level of network congestion and uses that information to determine whether the packet losses are likely to be due to congestion or random bit errors. In contrast, DDLRP can detect and differentiate the loss of retransmitted packets and can react accordingly.

In [6], the authors were Prasanthi.S and Sang-Hwa Chung, proposed a solution for detecting the loss of retransmitted packets over multi-hop wireless mesh networks. Contrast to [6], DDLRP can differentiate the type of loss and can increase the accuracy of detecting the loss of retransmitted packets [1].

As of above research study, we found that all TCP Variants which are discussed in this section is able to react for the loss of packet by the occurrence of retransmission timeout and triple duplicate acknowledgement. However, this proposed work is revealed with an efficient scheme which is required for accomplishing Throughput, Packet Reception Time, Packet Delay and Packet Dropping Rate as well. For addressing these issues, this work is proposed of an efficient technique, called as TCP-EPRT which is used to deduce the packet loss by explicit loss notification through which TCP Sender retransmits the lost packet without waiting for retransmission timeout and triple duplicate acknowledgement. The explicit loss notification is used at TCP Sender of Wireless Network. It deduces the packet loss before it occurs at TCP Receiver thereby the packet can be retransmitted and the transmission speed can also be minimized according to the availability of Window Size of TCP Receiver of Wireless Network.

Identified Problems: To overcome the performance degradation of TCP due to retransmission timeouts caused by retransmission loss, we introduced a new retransmission scheme by modifying the fast retransmission and fast recovery algorithms of TCP NewReno as it is the most widely deployed transport protocol used in computer networks. In fast retransmission, when the sender receives three dupacks the current implementation of TCP NewReno stores the highest sequence number transmitted in a variable “Recover”, retransmit the lost packet and set cwnd to ssthresh plus 3*mss. Then, TCP sender enters into fast recovery and increment cwnd by one mss for each additional dupacks and transmits new packets, if allowed by the new value of cwnd and the receiver’s advertised window. When the sender receives a new ack including Recover, the sender sets cwnd to ssthresh and goes to congestion Avoidance (CA) state [1].

On the other hand, if this new ack does not include Recover, then the sender consider it as a partial ack, retransmit the first unacknowledged segment and add back one mss to cwnd and send a new segment if permitted by the new value of cwnd. But, if the retransmitted packet is again lost, TCP NewReno has no mechanism to detect the lost packet [1]. As a result, TCP NewReno is unable to retransmit the lost packets and waits for the expiration of retransmission timeouts. Thus, for addressing the solution for this problem, we have proposed a novel scheme on TCP, called TCP-EPRT which deduces the packet loss through the Explicit Loss Notification for immediate retransmission of the lost packet.

Proposed Work: In the section of proposed work, we describe TCP-EPRT and TCP-Newreno which are almost behave in same but the difference of what TCP-EPRT has only the mechanism of Explicit Loss Notification which deduces the packet loss and does retransmission without waiting for the techniques which are employed in the TCP-Newreno. The experimental result showed that the proposed technique of explicit loss notification has outsmarted the existing techniques of retransmission and triple duplicate acknowledgement. As well as, the Throughput of proposed TCP-EPRT is comparatively better than the throughput of existing TCP-Newreno.

TCP-EPRT: The retransmission for packet loss can only be happened by the deduction of retransmission timeout or by the triple duplicate acknowledgement in the existing TCP variants of TCP-Newreno therein the network performance is degraded. To achieve better network performance, the proposed work has identified two major issues wherein the performance of network is gotten degrading. The issues are Packet Dropping Rate and Packet Reception Time which are stately given in the related work of this paper.

To improve the packet reception time, TCP Sender uses the technique of explicit loss notification that check mark the last sequence number of TCP sender and the current acknowledgement number of TCP Receiver if that is equalized in same then that will be considered as packet loss for immediate retransmission thereby the network idletime is cut down to achieve better packet reception time and end-to-end throughput compared of exiting TCP-Newreno. As well as the packet dropping rate is minimized to enhance the overall performance of the network.

Pseudo-Code of Proposed Technique: In the proposed technique, Explicit Loss Notification is adopted to enhance the packet reception time of the wireless networks and to cut down the packet dropping rate of the wireless network as well.

The adopted solution brings out the better throughput performance compared of TCP-Newreno by thorough investigation using Network Simulator tool NS2. The throughput of the network could be enhanced by increasing packet reception time and reducing the packet dropping rate that is achieved by the Explicit Loss Notification.

Algorithm 1 - Pseudo - Code of Proposed Technique:

```
if (Sender SeqNo and Receiver AckNo are same)
if (ELN Flag is raised) {
retransmit the lost packet;
return; }
if (dupacks && !nofastRetrans) {
Enter into duplicate action ;}
else {
Enter into send-one function ;}
```

The proposed pseudo code is successfully built with TCP-EPRT to achieve better packet reception time, to cut down the packet dropping rate, to pull down the delay of packet. Overall of the achievement of proposed technique has accomplished the good network throughput performance compared of the existing techniques of TCP-Newreno.

When the sender successfully sends the packet with SeqNo1 to the receiver, it starts waiting for the receiver acknowledgement number that is SeqNo2. If that is not received by the sender then the sender will have to wait for the deduction of packet loss which is done either by retransmission timeout or by triple duplicate acknowledgement for doing the packet retransmission. This is ever done by all the existing technique of TCPs. Whereas the proposed technique will deduce the packet loss by explicit loss notification. If ELN is detected by the sender for the packet which was sent before then TCP Sender does not await for retransmission timeout as opposed it will immediately resend the loss packet without making a network on idle(by waiting for retransmission timeout) thereby the network performance is enhanced in the proposed technique of TCP-EPRT.

Fig 2 shows that the receiver has lost the packet whose SeqNo is 2 which gets replicated in both Sender and Receiver that is notified by ELN bit of the sender(which is activated at TCP Sender) for immediate retransmission. As by the enhancement code of explicit loss notification of TCP-EPRT, we have been able to give most estimable results of throughput, packet reception time, packet dropping rate, shortening packet delay as compared of existing variant of TCP-Newreno.

Performance Analysis: The section of performance analysis is dealt of thorough investigation of existing technique (TCP-Newreno) and the proposed technique (TCP-EPRT). The comparative study has shown the proposed technique results are meliorated than the existing technique. The proposed technique reveals the meliorated results in terms of Throughput, Packet Reception Time, Packet Dropping Rate and Delay when corroborated with the existing technique. TCP-EPRT and TCP-Newreno are thoroughly evaluated by the Network Simulator tool NS2 for depicting the aforementioned results.

Wireless Experimental Setup: In this section, we describe our Wireless Network Environment and parameter settings used for evaluating the performance of TCP-EPRT. We designed a fully wireless environment consists of three senders (namely S1, S2, S3), three receivers (namely Rx1, Rx2, Rx3) and two routers (namely R1, R2) with one bottleneck link L1. Figure3 shows the wireless simulation network environment of our work which has S1, S2, S3 are used to represent TCP senders and Rx1, Rx2, Rx3 are used to represent TCP receivers. The R1 and R2 has one bottleneck link (L1) which is used to transfer the data for TCP Senders and TCP Receivers. FTP is the traffic source is used in all of the simulations. The maximum segment size of TCP is set to 1460 bytes. The size of an acknowledgment packet is same as the size of a data packet. The data transmission rate is set to 2Mbps, which is shown in the Fig 3.

All nodes are static in our simulations. All the simulations have been carried out with IEEE 802.11b MAC layer protocol. We enabled ELN at TCP Sender and the router buffers are set to a constant value which is equal to 90 packets. The duration of the simulation lasts upto 400 seconds. We did not impose random loss on the links using the exponential error model which is supported by NS2. For causing retransmission loss we did not adjust the time setting which enables packet transmission error. The necessary modification of code has also done in NS2 to undertake the simulations. This proposed technique (TCP-EPRT) is thoroughly evaluated with existing technique (TCP-Newreno) to accomplish the melioration

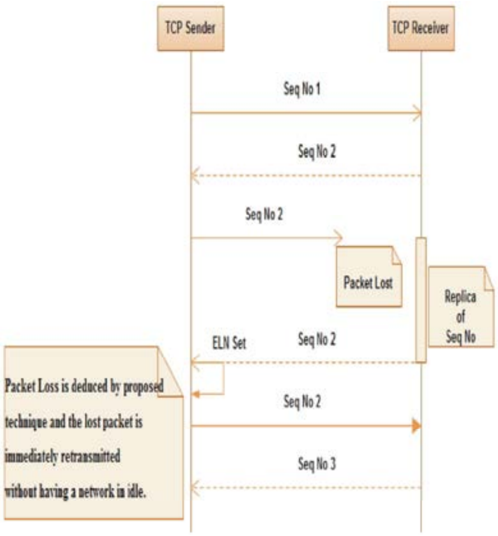

Fig. 2: Working Flow of TCP-EPRT

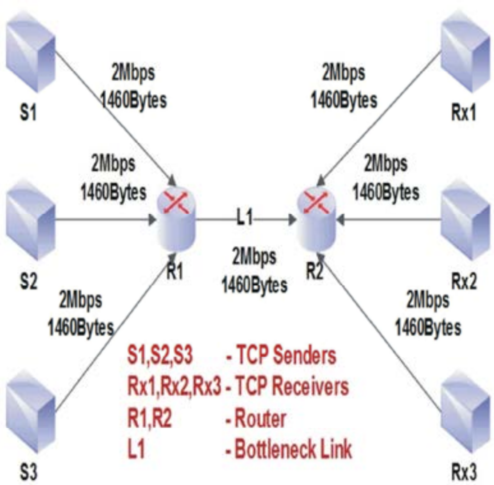

Fig. 3: Wireless Simulation Environments

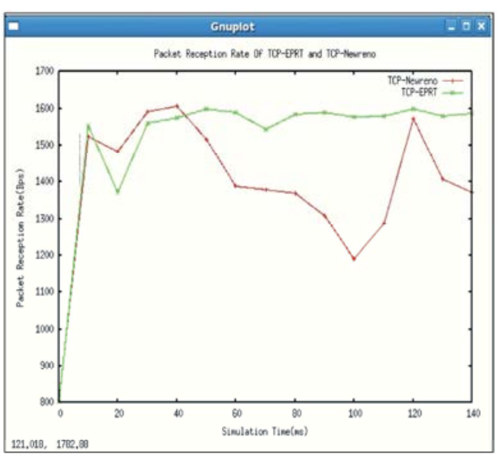

Fig.4 : PRT of TCP-EPRT Vs PRT of TCP-Newreno

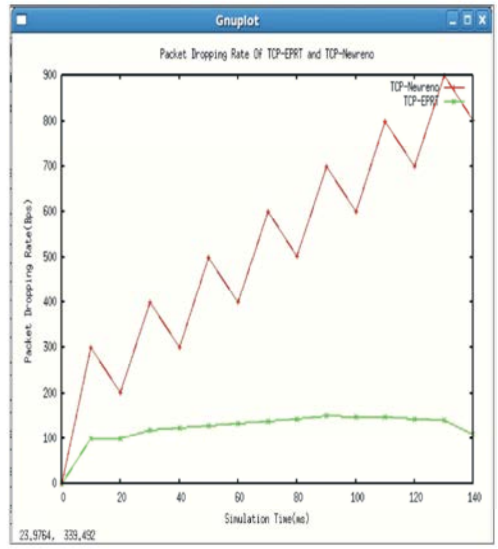

Fig. 5: PDR of TCP-EPRT Vs PDR of TCP-Newreno

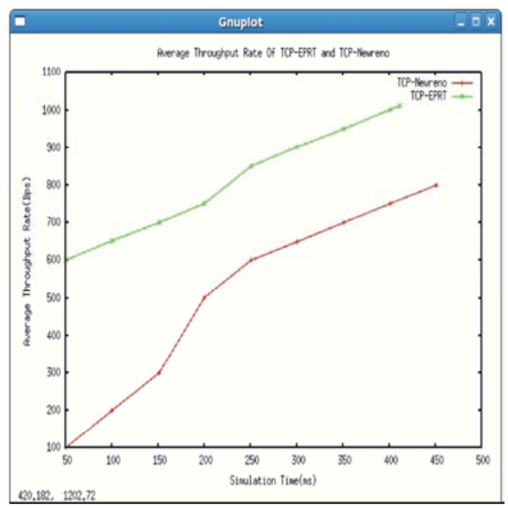

Fig. 6: Average Throughput Rate of TCP-EPRT and TCP-Newreno

Comparative Analysis Of TCP-EPRT and TCP-Newreno

Parameters	TCP-EPRT	TCP-Newreno
Average Throughput	99.3%	97.1%
Average Delay	0.38 Seconds	0.53 Seconds
Maximum Delay	9.3 Seconds	41 Seconds
Packet Dropping Rate	Less	More
Packet Reception Rate	90% of Packets Delivered	<60% of Packets Delivered & <10% of Packets Partially Acknowledged

Fig. 7: Analysis of TCP-EPRT and TCP-Newreno

results in terms of Throughput, Packet Reception Time And Packet Dropping Rate as well. The meliorated results of TCP-EPRT and TCP-Newreno are depicted in the following graph plotting.

- Packet Reception Time of TCP-EPRT and TCP-Newreno
- Packet Dropping Rate of TCP-EPRT and TCP-Newreno
- Throughput Performance of TCP-EPRT and TCP-Newreno

Packet Reception Time of TCP-EPRT and TCP-Newreno: Packet Reception Time is one of the important metrics used for evaluating the throughput performance of TCP.

Here, we present the packet reception time of TCP-EPRT and TCP-Newreno. The graph is plotted for Simulation Time (ms) Vs Packet Reception Rate (Mbps). Fig 4 depicts clearly that TCP-EPRT is progressed steadily in receiving the packets for every simulation time of generating packet whereas TCP-Newreno is fluctuated up and down in receiving the packets for every simulation time of generating packet. Also the comparative study reveals that the PRT of TCP-EPRT has 90% in receiving the generated packets of TCP Senders of the wireless networks whereas the PRT of TCP-Newreno has < 60% in receiving the generated packets of TCP Senders of the wireless networks. Thus, we hereby established that TCP-EPRT gives trust to the communication node of the wireless networks in delivering the packets.

Packet Dropping Rate of TCP-EPRT and TCP-Newreno: Packet Dropping Rate is also an important metrics used for evaluating the throughput performance of the network. Here, we present the packet dropping rate of TCP-EPRT and TCP-Newreno. The graph is plotted for Simulation Time (ms) Vs Packet Dropping Rate (Mbps). The technique of explicit loss notification is used to avoid packet retransmission delay and network idle time thereby we can minimize the packet delay and enhances the throughput performance of the wireless networks. The comparative study reveals that the average packet delay of TCP-Newreno is 0.53 seconds whereas the average delay of TCP of TCP-EPRT is 0.38 seconds. Moreover, the maximum delay of TCP-Newreno is comparatively untouched when compared to the maximum delay of TCP-EPRT. From the Fig 5, it reveals that the PDR of TCP-Newreno has much packet dropping rate compared of PDR of TCP-EPRT. As well as, PDR of TCP-Newreno has steadily increasing the packet dropping since generation of packets whereas the PDR of TCP-EPRT has comparatively less packet dropping and also it is above to meet 1% dropping rate at the end of the simulation.

Throughput Performance of TCP-EPRT and TCP-Newreno: Throughput performance is a pointing objective of Wireless Network and the melioration of the Throughput Performance comes from the Packet Reception Time and Packet Dropping Rate as well. In the viewpoint of aforementioned, we have found that as the problems which deteriorate the performance of the wireless network. Thus, we have proposed a novel scheme which is capable of overriding the existing problem of TCP-Newreno to meet out the Expected Throughput Performance. The plotted graphs which are shown in the Fig 4 and Fig 5 would reveal that the proposed technique is overcome the problem of packet dropping rate which is more in the existing technique of TCP-Newreno.

Moreover, the proposed technique is accomplished that 90% of generated packet of TCP Senders is able to reach TCP Receivers, whereas the existing technique is above to only 60%. Thus, the throughput of the proposed technique of TCP-EPRT is considerably high as compared with existing technique TCP-Newreno which is shown in the Fig 6.

The melioration results (PDR, PRT) bring the TCP-EPRT to accomplish steady throughput performance of the wireless network. Moreover, the comparative study divulges that network throughput performance of TCP-EPRT is 99.3% whereas TCP-Newreno is 97.1%. Therefore, the accomplishment of TCP-EPRT is relatively more compared of TCP-Newreno and also TCP-EPRT could be used to accomplish the meliorated results in terms of Throughput, Packet Dropping Rate and Packet Reception Time over Wireless Networks.

Comparative Analysis of TCP-EPRT and TCP-Newreno: The comparative study divulged the accomplishment of TCP-EPRT and TCP-Newreno over wireless network. The resultant of TCP-EPRT has proved as more estimable than the resultant of TCP-Newreno. The tabular view is shown in Fig 7.

CONCLUSION

We have proposed of new novel scheme, called as TCP-EPRT for improving the network performance over wireless networks. The objective of our novel scheme is to overcome the pitfalls of packet dropping rate and packet reception time which are degrading the throughput performance of the existing variants of TCP. The solution for which has reached by the novel scheme of TCP-EPRT which resolves the aforementioned pitfalls. The thorough investigation is done in Network Simulator (NS2) whose version is 2.34. The existing TCP Variants which has taken for thorough evaluation is TCP-Newreno. The resultant graph and comparative study have proven that the proposed novel scheme of TCP-EPRT is accomplished the most estimable result in the perception of Packet Dropping Rate, Packet Reception Time, Packet Delay and Throughput as compared with existing novel scheme of TCP-Newreno over Wireless Networks.

We would like to develop an efficient technique which would proactively deduce the packet loss and besides it would inform TCP Senders for retransmitting the lost packet whereby we would be able to enhance the network performance by showing a Proactive Measurement Technique of packet loss deduction for avoiding time delay retransmission. The stated above could be done in the future work.

REFERENCES

1. Prasanthi, S. Sang-Hwa Chung and Changwoo Ahn, “An enhanced TCP Mechanism for Detecting and Differentiating the Loss of Retransmission over Wireless Networks”, 2011 International Conference on Advanced Information Networking and Application, doi: 10.1109.
2. Kothari, N.J. Gambhava and Dasgupta, “RTT utilization by detecting avoidable timeouts”, Proc. 14th IEEE International Conference on Networks (ICON 06), Sept 2006, pp: 1-6. doi: 10.1109/ICON.2006.302570.
3. Jaehyung Lee, Jungre, Minu, Jahwan and Hyunseung, “NJ+: An Efficient Congestion Control Mechanisms for Wireless Networks”, KSII Transactions on Internet and Information Systems,
4. Shagufta Henna, 2009. “A Throughput Analysis of TCP variants in MobileWireless networks”, Proc. NGMAST’09, USA, 2009.
5. Ciullo, D., M. Mellia and M. Meo, 2009. ”Two Schemes to Reduce Latency in Short Lives TCP Flows”, IEEE Communications Letters, 13: 10.
6. Prasanthi, S., Sang-Hwa Chung, “An Efficient Algorithm for the Performance of TCP over Multi-hop Wireless Mesh Networks”, Proc. Seventh International Conference on Information Technology: New Generations, pp: 816-821. doi: 10.1109/ITNG.2010.129.
7. Anusuya and Krishna “Performance Analysis of an Efficient TCP Variant under Lossy Environments”, Proc. of ICACC’09.
8. Cheng-Yuan Ho, Yaw-Chung Chen, Yi-Cheng Chan and Cheng-Yun Ho, 2008. “Fast Retransmit and Fast Recovery Schemes of Transport Protocols : A survey and taxonomy”, J. Computer Networks.
9. Allman, M., H. Balakrishnan and S. Floyd, 2001. “Enhancing TCP’s Loss Recovery Using Limited Transmit,” RFC 3042,
10. Dongmin Kim, Beomjoon Kim, Jechan Han and Jaiyong Lee, 2004. "Enhancements to the Fast Recovery Algorithm of TCP NewReno”, Proc. of ICOIN 2004, LNCS 3090, Springer-Verlag, 3090: 332-341.
11. Kim, D., B. Kim, J. Han and J. Lee, 2004. “Enhancement to the Fast Recovery Algorithm of TCP Newreno”, ICOIN.
12. Cheng Peng Fu, and Liew, S.C. TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks. IEEE J. on Selected Areas in Communications, 21(2): 216-228.
13. Hosam, “TCP CERL: Congestion control enhancement over wireless networks”, J. Wireless Networks.
14. Brakmo and L.S, 1994. “Vegas: New techniques for Congestion Detection and Avoidance”, Proc. ACM SIGCOMM,
15. Altman, E. and T. Jimenezl, ”Novel Delayed Ack Techniques for Improving TCP performance in Multi Hop Wireless Networks”, Personal Wireless communications, LNCS 2003, pp: 237-250, doi: 10.1007/978-3-540-39867-7_26.
16. Michelle Berger, Lima, Manoussakis, Pulgarin and Sanchez, ”A performance Comparison of TCP protocols over Mobile Ad Hoc Networks”, Proc. IEEE CERMA 2006, pp: 88-94. doi: 10.1109/CERMA.2006.9.
17. Floyd, Henderson and Gurtov,”The NewReno Modification to TCP’s Fast Recovery Algorithm”, RFC 3782, 2004.
18. Fu, Z., Z. Petros, Haiyum, Songwu, Lixia and Mario Gerla, “The impact of Multihop ireless Channel on TCP Throughput and Loss”, Proc. IEEE Annual Conference on Computer Communication (INFOCOM’03), pp. 1744-1753, doi: 10.1109/INFCOM.2003.1209197.
19. Xu, Tian and Ansari, “Improving TCP performance in Integrated Wireless Communication Networks”, Computer Networks, 2005, doi:10.1016/j.comnet.2004.07.006.