Simulation Analysis of TFRC in Wireless Networks

Sreekanth Bandi
M.Tech, CSE Dept., JNTUA College of Engineering, Anantapuramu, A.P., India.

C. Shoba Bindu
Associate Professor, CSE Dept., JNTUA College of Engineering, Anantapuramu, A.P., India.

Abstract – In wireless networks packet losses occurs mostly because of congestion in the network. Various congestion control mechanisms are used to handle the congestion. But these may degrades the end-to-end performance and decreases the throughput in wireless networks. In this paper, TCP Friendly Rate Control (TFRC) is used to evaluate the performance in Real Time Applications (RTA's). TFRC is friendlier than UDP towards TCP traffic in many different congestion situations and is suitable for use by internet applications such as streaming multimedia and RTA's that currently run over UDP and cause unfriendliness towards TCP applications. The paper also measures the performance of TFRC with various congestion control mechanisms in wireless Environment such as UDP, TCP, TCP New Reno and TCP Vegas. In the performance evaluation of TFRC with various TCP variants we use NS2 simulator.

Index Terms – Congestion Control, NewReno, Real Time Applications, Vegas, TFRC.

1. INTRODUCTION

The Transmission Control Protocol (TCP) is designed for reliable, ordered data transmission in wired networks. TCP treats packet loss as an indication of the congestion, so it provides various end-end congestion control mechanisms but TCP is not suitable for Real Time Applications (RTA's) such as multimedia streaming, online gaming, voice – over – IP, etc. It may results in decreasing the performance [1]. User Datagram Protocol (UDP) is designed for Real Time Applications. But UDP does not have congestion control mechanism and does not guarantee reliable data transfer. Various conventional protocol like TCP NewReno are designed to be used in wired networks with low link error rates, their original design do not consider link errors. Whenever congestion occurs TCP sender reduces its sending rate as a remedial measure. When conventional TCP are used over wireless network may decrease the throughput. Various end to end congestion control mechanisms [2][3] are developed to improve the throughput over wireless networks. Most of these mechanisms are reactive. Whenever packet loss occurs, they need to find out the actual reason for the loss, whether due to the network congestion or wireless link loss. Those are the loss based approaches. In addition new proactive approach for the TCP congestion control has been developed. These are delay based approaches. One of the delay based approach TCP Vegas offers effective solutions for wireless link loss. Both TCP NewReno and TCP Vegas work better in wired environment but do not work effectively in Real Time Applications.

In this paper, TFRC is used to measure the performance with various UDP, various TCP variants. It has TCP Friendly behavior over wireless networks. This mechanism is suitable for real-time applications like multimedia streaming, online gaming, voice-over-IP, etc.

The rest of the paper is represented as follows. *Section 2* summarizes the most recent relevant works. In *Section 3* we present how delay can be handled in our TFRC and its improvements. In *Section 4* we discuss in detail about simulating various congestion control mechanisms with TFRC and measure the performance. Finally we conclude in *Section 5*.

2. RELATED WORK

All Various congestion control algorithms have been proposed [3][4][5][6]. Congestion Control deals with controlling traffic entry into a telecommunication network, so as to avoid congestion collapse by avoiding over subscription of any of the processing or link capabilities of the intermediary nodes and networks and taking the resource reducing steps, such as minimizing the rate of sending packets.

Few of the congestion control mechanisms are,

2.1 TCP NewReno

TCP NewReno improves re-transmission during the Fast Recovery phase of the TCP Reno [7]. It includes a small change to the TCP Reno algorithm at the sender. The variation concerns the sender's behaviour during the Fast Recovery when the partial ACK received. The partial ACK do not acknowledges all the packets that are outstanding at the start of the Fast Recovery phase but acknowledges only some of them. This means there exist multiple packet losses in the fixed window size of data.

In TCP Reno, partial ACK's take TCP out of Fast Recovery by making the usable window size same as that of congestion window size. But in TCP NewReno partial ACK's do not take TCP beyond Fast Recovery. Instead, partial ACK's received during the Fast Recovery are preserved as an indication that the packet immediately followed the acknowledged packets in the order space has been lost and would be re-transmitted.

Thus, whenever multiple number of packets are missing from a window of data TCP New Reno can recover without the re-transmission timeout, re-transmitting one lost packet per RTT until all lost packets from window have been transmitted. TCP NewReno remains in Fast Recovery until all the data outstanding whenever Fast Recovery was initiated all outstanding data should be acknowledged.

Problem with TCP NewReno is, it suffers from the fact that it takes one RTT to detect each packet loss. When the ACK for the first retransmitted segment is received only then can we assume which other segment was lost.

2.2 TCP Vegas

TCP Vegas uses Round Trip Time to measure the network situation. It is used to check increasing or decreasing the congestion window value by using expected and actual efficiency [8].The idea is that when the network is not congested, the actual flow rate will be close to the expected flow rate. Otherwise, the actual flow rate will be smaller than the expected flow rate. Using the difference in flow rates estimates the congestion level in the network.

Diff = (Expected Rate – Actual Rate) BaseRTT

Expected Rate = CWND/Base RTT

Actual Rate = CWND/Actual RTT

TCP Vegas tries to keep at least α packets but no more than β packets in the queues [9]. Based on Diff value the sender updates its window size asWhen Diff < α, Vegas increase the Congestion Window (cwnd) linearly during next RTT, and when Diff > β, Vegas decrease the cwnd linearly during the next RTT. Vegas leaves the cwnd unchanged when α < Diff < β.

The Problem with TCP Vegas is, it works better in wired environment because of its retransmission mechanism and modified slow-start mechanism. But it does not work effectively in real time applications.

3. Improvements of TFRC

The TFRC is a congestion control mechanism for uni -cast flows operating in a best effort Internet environment [9]. It is practically fair when competing for bandwidth with TCP streams, but has a much lower variation of throughput over time compared with TCP, making it more suitable for RTA's, where it has a relatively smooth sending rate [10].

TFRC has TCP friendly behavior over wireless networks. It is an equation based Congestion control approach and works based on the following throughput (T) equation.

$$T=\frac{s}{R\sqrt{\frac{2p}{3}}+RTO\left(3\sqrt{\frac{3p}{8}}\right)p(1+32p^2)} \qquad (1)$$

Where

R	-	Round Trip Time
S	-	Packet Size
P	-	Loss Event Rate
RTO	-	Retransmission timeout value in Seconds.

To calculate the Loss Event Rate (p), receiver needs to find the loss event of one or more packets lost or marked in particular RTT. Timestamp along with RTT is used by receiver to determine losses belong to same loss event or not. Loss event rate and RTT is then fed to TCP throughput equation at senders end to calculate the TCP friendly rate. Sender then adjusts its sending rate according to this calculated rate. TFRC provides smooth sending rate while as well as providing sufficient responsiveness to competing traffic. It allows moderate bandwidth changes and is more appropriate to video streaming.

3.1 Functionality of TFRC

TCP Friendly Rate Control (TFRC) mechanism described in Figure1. It has TCP friendly behavior of the protocol [10] [11]. To determine the network condition various measurements have been considered at the receiver as well as the sender. Sender sends the data packets to the receiver to calculate the sending rate at the receiver side. Receiver generates receiver reports to the sender.

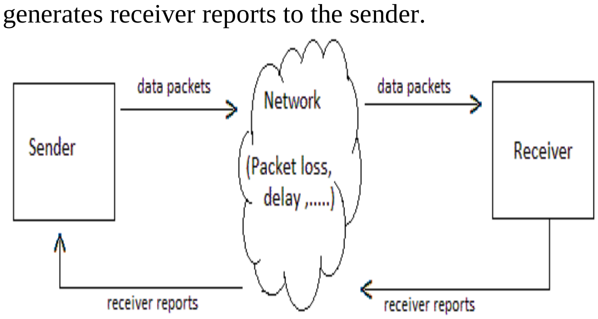

Figure 1: Functionality of TFRC

The sender needs to transmit the data packets to the receiver with a specific data rate. Those data packets contain packet loss or delay in the network. At the receiver side, the loss event rate (p) is calculated and then sends feed back to the sender. After receiving these receiver reports, the sender is able to determine the RTT between the sender and the receiver. The estimated Loss Event Rate (p) and RTT both parameters are used to calculate the transmission rate based on the actual network situation. After that, the sender adjusts its transmission rate to the calculated value.

Previously [12], the performance of TFRC is compared with UDP and TCP in wired environment. TFRC performance found that, it is better than both TCP and UDP with respect to throughput, end to end delay and packet loss ratio. In this paper, we compare the performance of TFRC with various TCP variants like TCP NewReno, TCP Vegas.

4. Evaluation

The Network Simulator NS-2, version 2.3.5 is used for simulation. We consider the performance metrics *Packet delivery ratio (PDR), End-to-End delay, Packet loss ratio* to evaluate the performance of TFRC.

The following congestion control mechanisms have been considered for comparison:

- TFRC with UDP having Constant Bit Rate (CBR).
- TFRC with TCP having congestion window size.
- TFRC with TCP NewReno having congestion window size.
- TFRC with TCP Vegas having congestion window size.

4.1 Simulation

We evaluate the performance of TFRC along with their effects on the various congestion controls using simulation scenario of Figure2.

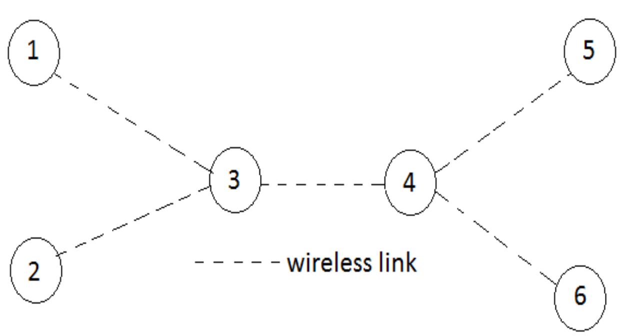

Figure 2: Simulation Scenario

The topology we are using is dumbbell topology. 6 nodes are created in wireless environment. Using this topology, we have performed tests to analyse the performance of TFRC with UDP and various TCP variants in the occurrence of a CBR flow. You can think of that CBR flow as an un-responsive UDP flow. It uses a particular amount of bandwidth and does not care about the dropped packets and not perform congestion control. Basically, it just sends packets blindly at a constant bit rate. Add a CBR source at node2 and a TFRC rate at node1, and then add a single TFRC stream from node1 to a sink at node6. In TCP, it sends packets by changing the TCP window size at node2 and change the TFRC rate node1. Now compare UDP, various TCP variants with TFRC.

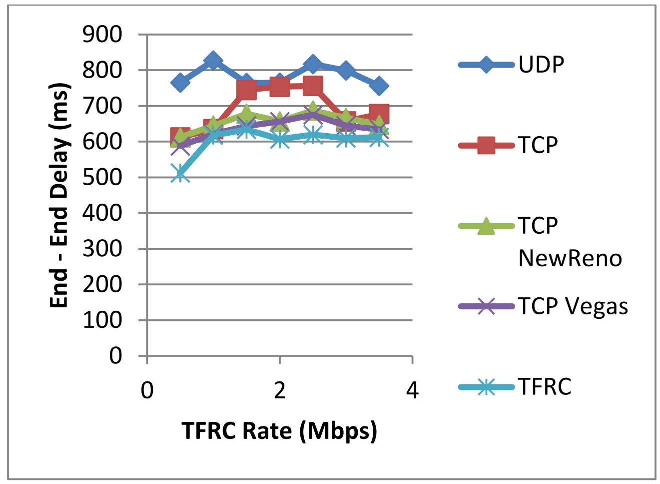

Figure 3: End –end –end Delay of TFRC, UDP and TCP Variants with 6 nodes

In Figure 3, consider the packet size as 1000 bytes for TFRC and TCP. TFRC rate and CBR rate increased simultaneously for allowing the same sending rate. It varies from 0.5 Mb to 4.0 Mb. TCP window size change from 10 to 60, and measure the end to end delay. It shows that end to end delay of TFRC is less than the UDP, various TCP variants. Compare to TCP Vegas, TFRC performance increased somewhat with respect to delay.

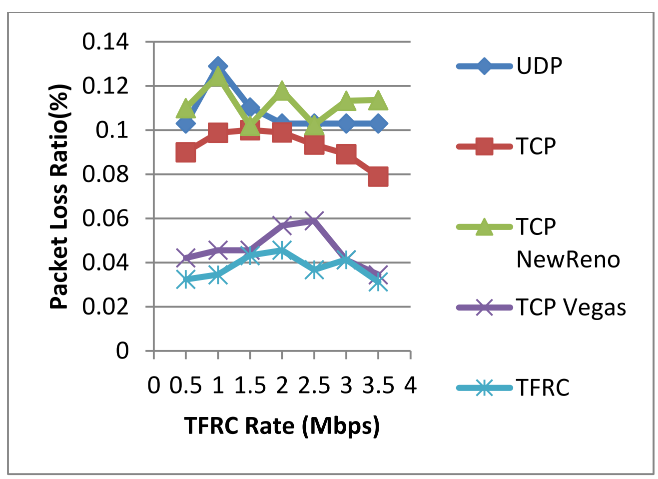

Figure 4: Packet Loss Ratio of TFRC, UDP and TCP Variants with 6 nodes

In Figure 4, the packet loss ratio of TFRC compare with UDP, various TCP variants. Here consider packet size as 1000 bytes for TFRC and TCP. CBR rate and TFRC rate increased simultaneously for allowing the same sending rate. The data rate changes from 0.5 Mb to 4.0 Mb. TCP window size also increased according to the sending rate. The results show that packet loss ratio of TFRC is less than the UDP, various TCP variants. TFRC performance is more compare to TCP Vegas regarding Loss.

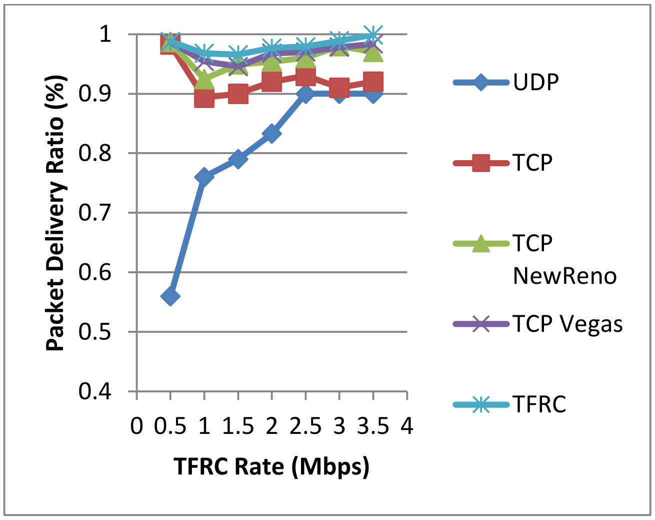

Figure 5: Packet Delivery Ratio of TFRC, UDP and TCP Variants with 6 nodes

Figure5 shows, the Packet Delivery Ratio of TFRC is more than the UDP, Various TCP variants. Consider the same packet size for both TFRC and TCP. TFRC rate, CBR rate is increased simultaneously for allowing the sending rate incremented 0.5Mb. TCP window size change from 10 to 60, and measure the packet delivery ratio by changing the TFRC rate. TFRC is friendlier with TCP Vegas in packet delivery ratio but more aggressive with TCP and UDP.

TFRC has better performance than TCP, UDP and various TCP variants, friendlier than TCP Vegas. TFRC is cable of replacing TCP variants and UDP for multimedia streaming and Voice –over –IP applications. So Real Time Applications may use the TFRC protocol to get the better performance.

5. CONCLUSION

In this paper performance of TFRC is compared with UDP and various TCP variants in wireless environment. TFRC performance is found to be better than UDP and various TCP variants. TFRC is fair with TCP variants and better performance with UDP. TFRC has the congestion control mechanism and it competes with other TCP data flows. These results show that TFRC is getting far better results than other congestion control variants in wireless environment.

REFERENCES

[1] Satoshi Utsumi, Salim Zabir , "A new high-performance TCP friendly congestion control over wireless networks", Journal of Networks and Computer Applications 41(2014).

[2] M. Allman, V. Paxson, W. Stevens "TCP Congestion Control", IETF RFC 5681, September 2009.

[3] Balakrishnan H, Padmanabhan V. "A comparison of mechanisms for improving TCP performance over WirelessLinks", IEEE/ACM Transactions on Networking, 1997: S (6):756-69.

[4] Brakmo LS, Peterson LL."TCP Vegas: end to end congestion avoidance on a global internet". IEEE J Sel Areas Communications 1995:13(October (8) :1465-30.

[5] Jacobson V. "Congestion Control and Avoidance". In: Proceedings of ACM special interest group on data communications (SIGCOMM), 1988.

[6] Jacobson V, Barden R, Borman D. "TCP extensions for high performance" , RFC 1323, 1992.

[7] Madiha Kazmi, Azra Shamim "Comparison of TCP Tahoe, Reno, New Reno, Sack and Vegas in Networks under Constant Bit Rate Traffic", International conference on advance computational Technologies & creative media, Aug. 14-15, 2014 pattaya.

[8] Jehan.M, G.Radhamani, T.Kalakumari, "VEGAS: Better performance than other TCP congestion Control algorithms on MANETs", International Journal of Computer Networks (IJCN), volume (3): Issue (2): 2011.

[9] Ghassan A. Abed, Mahamod Ismail,Kasmiran Jumari, "Exploration and Evaluation of traditional TCP congestion control Techniques", Journal of Kind Saud University – Computer and Information sciences (2012) 24,145-155.

[10] Handley, Floyd, Widmer and Padhye, "TCP-Friendly Rate Control (TFRC): Protocol Specification", IETF RFC 5348, April 2008.

[11] Christos Bouras, Vassilis Papapanagiotou "The TFRC protocol and its usage for wireless video transmission", In proceedings of: Software, Telecommunications & Computer Networks, October 2009.

[12] P.Sreenivasa Rao, M. Janani and P. Chenna Reddy "TFRC for congestion control in wired Environment" International Journal of Inventive Engineering and Sciences ISSN:2319-9598, Volume-2, Issue-1, December 2013.

Authors

Sreekanth Bandi received B.Tech degree in Computer Science and Engineering from BIT Institute of Technology, hindupur, affiliated to JNTUA, Anantapuramu, A.P, India, during 2002 to 2006. And worked as an Assistant professor in Mekapati Rajamohan Reddy Institute of Technology and Science, Nellore, India from 2006 to 2010.Worked as an Assistant professor in Sree Vidyanikethan Engineering College, Tirupuati, India from 2010 to 2013. Currently pursuing M.Tech in Computer Science from JNTUA College of Engineering, Anantapuramu, A.P, India, during 2013 to 2015 batch. His area of interesting is Computer Networks, Wireless Communication Systems.

C. Shoba Bindu is an Associate Professor of Computer Science and Engineering at Jawaharlal Nehru Technological University College of Engineering, Ananthapuramu. She obtained her Bachelor degree in Electronics and Communication Engineering, Master of Technology in Computer Science from Jawaharlal Nehru Technological University Hyderabad and Ph.D. in Computer Science and Engineering from Jawaharlal Nehru Technological University Anantapuramu. She has published several Research papers in National / International Conferences and Journals. Her research interests includes network security and Wireless communication systems.