ISSN 2319 5975

Volume 2, No.6, October - November 2013

International Journal of Networks and Systems

Available Online at http://warse.org/pdfs/2013/ijns02262013.pdf

TFRC for Congestion Control in Real Time Applications

M. Janani[1], P. Chenna Reddy[2]

[1]P.G student, J.N.T.U.A College Of Engineering Pulivendula, INDIA, muppalla.janani@gmail.com

[2]Professor, J.N.T.U.A College Of Engineering Pulivendula, INDIA, pcreddy1@rediffmail.com

ABSTRACT

Abstract: The amount of traffic generated by Real Time Applications (RTA) has increased substantially over the years. RTA will face congestion while there's any form of bottleneck restricting traffic, this can lead to packet loss or delayed traffic that is unacceptable for RTAs. Therefore it is desirable for RTA to implement congestion control mechanism to improve the steadiness of networks. The congestion problem has been addressed successfully by Transmission Control Protocol (TCP). TCP is connection oriented protocol that presents reliable and ordered delivery of packets and also presents end-to-end congestion control mechanism. But its congestion control mechanism does not suit the characteristics of the RTA. User Datagram Protocol (UDP) applications can send data at constant bit rate. It is non TCP based protocol. It cannot adjust its flow rate when congestion is detected and it continues to send at original rate. So these non TCP applications don't have congestion control mechanism and don't share bandwidth fairly with TCP based applications. A new congestion control protocol for datagram transport was defined i.e., TCP Friendly Rate Control (TFRC) standardized by Internet Engineering Task Force (IETF). TFRC is a congestion control algorithm that supplies a smooth transmission rate for RTAs. TFRC is a congestion control mechanism for unicast flows functioning in a best effort Internet environment. It's reasonably fair when competing for bandwidth with TCP flows in congested network, although encompasses a lot of lower variation of throughput over time compared with TCP.

Key words: Real Time Applications, TFRC, TCP, UDP.

1. INTRODUCTION

The most widespread use of the Internet has been the exchange of asynchronous information through a dependable transport protocol as TCP. The recent improvements in the network technologies have increased the popularity of RTAs that exchange synchronous information through an unreliable transport protocol UDP. These two distinct kinds of flows share the same network infrastructure and should implement mechanisms that warranty an optimal network usage.

TCP [1] is one of the important protocols in TCP/IP protocol stack. Whereas the IP protocol deals solely with packets, TCP permits two hosts to establish a connection and exchange streams of data. TCP provides ordered, reliable, error-checked delivery of a stream of octets between programs running on computers connected to a local area network, intranet or the public Internet. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent. Web browsers use TCP after they connect with servers on the World Wide Web, and it's used to deliver email and transfer files from one location to another location.

Applications that do not require the reliability of a TCP connection may instead use the connectionless UDP, which emphasizes low-overhead operation and reduced latency rather than error checking and delivery validation. UDP [2] is a communication protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP). UDP is an alternative to the TCP and, together with IP, is sometimes referred to as UDP/IP. Just like the TCP, UDP uses the Internet Protocol to really get a data unit (called a datagram) from one computer to another. Unlike TCP, UDP does not provide the service of dividing a message into packets and reassembling it at the other end. Specifically, UDP doesn't offer sequencing of the packets. This implies that the application program that uses UDP must be able to make sure that the entire message has arrived and is in the right order.

TFRC [3] [4] is a congestion control mechanism designed for unicast flows operating in an Internet environment and competing with TCP traffic. TFRC is designed to be reasonably fair when competing for bandwidth with TCP flows, where a flow is "reasonably fair" if its sending rate is generally within a factor of two of the sending rate of a TCP flow under the same conditions. However, TFRC encompasses a much lower variation of throughput over time compared with TCP that makes it more appropriate for applications such as telephony or streaming media where a relatively smooth sending rate is of importance.

TFRC mechanism works as follows:

- The receiver calculates the loss event rate and the received rate, then inform it to the sender in a feedback message.
- This feedback to the sender besides the loss event rate and the received throughput includes the echoed timestamp of the last data packet and a delayed time between the arrival of the last data packet and the generation of the feedback. These last two

parameters are necessary to calculate the round trip time (RTT) at the sender.
- The feedback packets are sent at least each round trip time or immediately after a new loss event rate is detected (without waiting one RTT).

Wireless communication technology [5] will be playing an increasingly important role in access networks, as evidenced by the widespread adoption of wireless local area network (WLAN), wireless home networks, and cellular networks. These wireless access networks are usually interconnected using wired backbone networks, and many applications on the networks run on the TCP/IP protocol.

A RTA is an application program that functions within a time frame that the user senses as immediate or current. The latency should be less than a defined value, sometimes measured in seconds. Whether or not a given application qualifies as an RTA depends on the worst-case execution time, the maximum length of time a defined task or set of tasks requires on a given hardware platform.

Examples of RTAs include:

- Video conference applications
- VoIP (Voice over Internet Protocol)
- Online gaming
- Community storage solutions
- Some e-commerce transactions
- Chatting
- IM (Instant messaging)

VoIP [6] is a methodology and group of technologies for the delivery of voice communications and multimedia sessions over IP networks, like Internet. Other terms commonly associated with VoIP are Internet telephony, IP telephony, voice over broadband (VoBB), broadband telephony, broadband telephone service and IP communications.

QoS Requirements of VoIP

- Data rates – 48Kbps to 512Kbps
- Loss ought to be no more than 1 percent.
- End-to-End delay should be no more than 150 ms

Videoconferencing [7] is the conduct of a videoconference by a set of telecommunication technologies which allow two or more locations to communicate by simultaneous two-way video and audio transmissions. It has also been called 'visual collaboration' and is a type of groupware. Videoconferencing differs from videophone calls in that it's designed to serve a conference or multiple locations instead of individuals.

QoS Requirements of VIDEO CONFERENCE

- Data rates – 384Kbps to 2Mbps
- Loss ought to be no more than 1 percent.
- End-to-End delay should be no more than 150 ms

Video streaming [7] is a streaming of frame one by one. A client media player will begin playing the data (such as a movie) before the entire file has been transmitted. Identifying delivery methodology from the media distributed applies specifically to telecommunications networks, as most alternative delivery systems are either inherently streaming (e.g., radio, television) or inherently no streaming (e.g., books, video cassettes, audio CDs). For instance, in the 1930s, elevator music was among the earliest popularly available streaming media; nowadays Internet television is a common form of streamed media. The term "streaming media" can apply to media other than video and audio such as live closed captioning, ticker tape, and real-time text, which are all considered "streaming text".

QoS Requirements of VIDEO STREAM

- Data rates – 2.5Mbps or high rates
- Loss should be no more than 5 percent
- End-to-End delay should be no more than 4 or 5 seconds

2. RELATED WORK

Sally Floyd and Eddie Kohler [8] are working on a variant of TFRC for VoIP that provides fairness to applications that send small packets. They argue that it is acceptable for VoIP flows to assume that network limitations are in bytes per second, that measure congestion in terms of the available bandwidth, rather than the more common measurement of packet per second which relates to routing functions such as header processing and packet forwarding. They proposed the subsequent changes to the basic TFRC protocol:

- Set the actual packet size to 1460 bytes,
- Reduce the formal transmit rate to account for the packet header,
- Impose a bottom interval between packets of 10ms.

These changes not only help the flow to share the available bandwidth in bytes per second, but also discourage application from using extremely small packet sizes. A faster restart is also introduced to improve voice application responsiveness after idle periods. Faster restart allows a idle flow to quadruple its sending rate in every congestion free RTT up to their previously achieved transmission rate. The sending rate account for the audio packet size and the headers of the IP, UDP and RTP protocols that are usually used for the audio packets. During the silent periods the source sending rate will be around one kbps, since the application still sends a packet every couple of RTT to maintain network state information. Using the faster restart and forward, no congestion is detected during the restart phase, the flow needs only 2 RTTs to achieve the previous stable sending rate of 12 kbps, current implementations of TCP and TFRC would require 4 RTTs.

Francesco Beritelli, Giuseppe Ruggeri, and Giovanni Schembra [6] addressed the problem of TCP-friendly algorithms for real time applications which are more concerned with QoS than fairness with TCP. They analyzed the problem of conveying voice utilising TCP-friendly protocols and proposed VoIP architecture for advancing the subjective quality of the transmissions. Their voice transmission system has three main constituents, the TCP friendly algorithm, the voice encoder and also the encoder controller. The voice encoder obtains the input from the voice source and uses a multi-rate encoder that adapts to the bandwidth imposed by the TCP-friendly rate control mechanism. The voice encoder sends the voice border to the packetized that prepares the packet for the TCP-friendly rate controller. The encoder controller utilizes the data from the TFRC algorithm to choose the appropriate voice cryptography. Since for voice applications a high decrease rate may not produce acceptable quality for voice communication, they suggested a change for TCP-friendly protocols to take into concern the RTT delay variation. Once the delay variation surpasses a choice threshold, the application cuts its rate in half and goes into a slow-start stage. Their experiment focused on a set of voice sources distributing a common bottleneck link. Their outcomes showed a rise in perceived user quality on group of sources that consists of 10 to 40 unique voice flows over a single link of capacity 150kbps.

3. PERFORMANCE METRICS

3.1 Packet delivery ratio: The ratio of total number of packets successfully received by the destination nodes to the number of packets sent by the source nodes.

3.2 End-to-End delay: The average delay of all the packets while travelling from source node to destination node.

3.3 Packet loss ratio: The ratio of number of lost packets to the sum of number of packets received and number of lost packets.

4. SIMULATION ENVIRONMENT

The network simulator NS-2, version 2.34 is used for simulation. NS2 [9] is an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. NS-2 supports TCP, UDP traffic with FTP, CBR and Telnet applications. Simulation for TCP, UDP and TFRC traffic is presented in wired network and wireless network. The topology we are using is dumbbell topology with bottleneck link of capacity 4Mbps at (R1-R2) and other links capacity of 4Mbps. In wireless environment, the mobility of the nodes is created using random way point model in a rectangular field of 1000 x 1000 sqm. A node chooses its initial position randomly, chooses the next position also randomly and moves towards it with chosen speed and pause time. In this simulation different pause times and speeds are used. Pause time is the amount of time a node remains stationary at a fixed position before moving from that position. A pause time of zero means continuous movement and a pause time equivalent to simulation time means node is static. Traffic models supported by NS2 are used to generate the traffic. Simulation time is restricted to 100sec. Post processing of the trace files generated by NS2 is done using awk scripts.

5. RESULT AND ANALYSIS

In this, we compare the performance of TCP, UDP and TFRC through Packet delivery ratio, End-to-End delay and Packet loss ratio.

5.1 Wired Environment

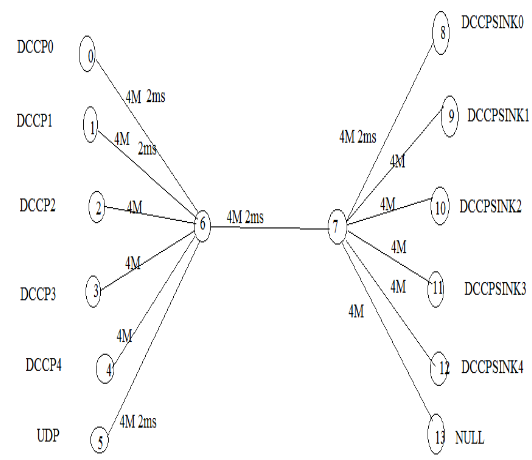

Figure 1: Simulation Topology

In this section, we show the performance of real time applications like VoIP, Video conference and Video stream in wired environment.

VoIP

Figure 2 shows the VoIP traffic with an encoding rate of 84 Kbps and packet size of 160 bytes at different number of nodes.

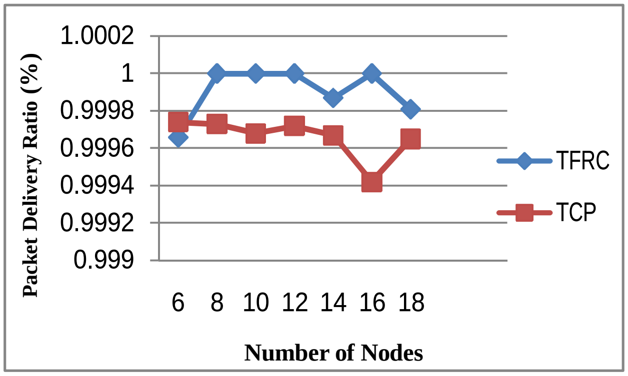

Figure 2: Packet Delivery Ratio of TFRC and TCP in Wired Environment: VOIP

Figure 2 shows that the packet delivery ratio of TFRC is more than that of TCP. In this, number of nodes is varied from 6 to 18.

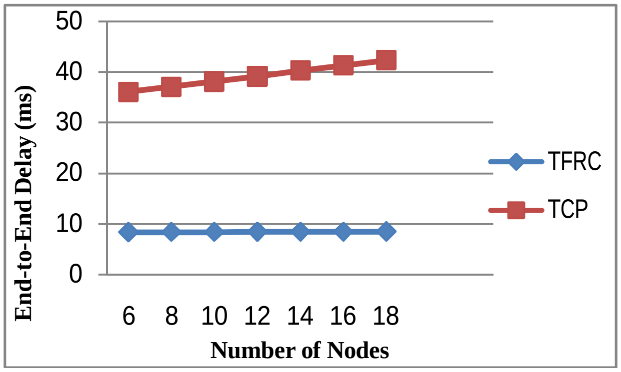

Figure 3: End-to-end delay of TFRC and TCP in Wired Environment: VOIP

The results show the end-to-end delay of TFRC is less than the TCP. The number of nodes is varied from 6 to 18.

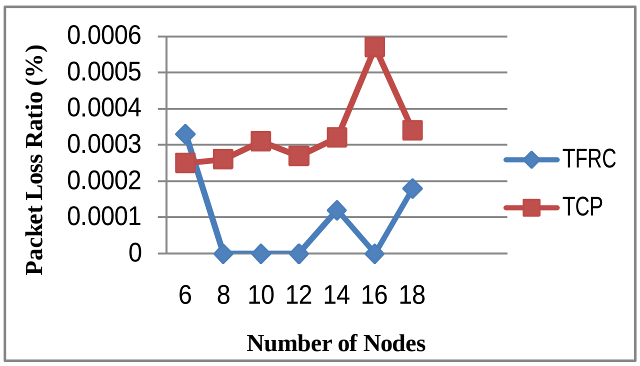

Figure 4: Packet Loss Ratio of TFRC and TCP in Wired Environment: VOIP

Figure 4 shows the packet loss ratio of TFRC is less than the TCP. In this the TFRC rate set to be 84Kbps and TCP window size is set to 10.

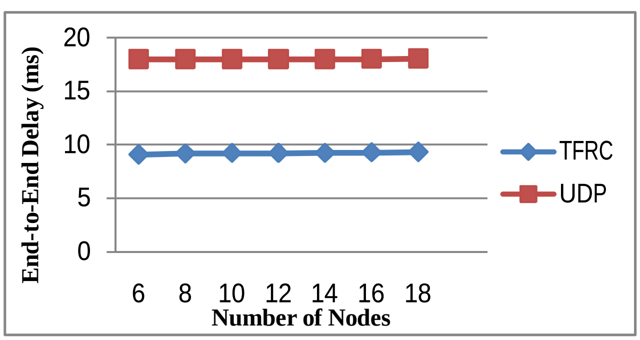

Figure 5: End-to-end delay of TFRC and UDP in Wired Environment: VOIP

Figure 5 shows that the end-to-end delay of TFRC is less than that of TCP. In this, the TFRC rate is set to 84Kbps and UDP rate is set to 1Mbps.

Video Conference

Table 1: Performance of TFRC and TCP in Wired Environment: Video Conference.

No. of Nodes	TFRC		TCP	
	Packet Delivery ratio	Packet loss ratio	Packet Delivery ratio	Packet Loss ratio
6	0.99947	0.0005	0.99967	0.00032
8	0.99929	0.0007	0.99936	0.00063
10	0.99865	0.0013	0.99871	0.00128
12	0.98271	0.0172	0.9979	0.00209
14	0.97563	0.0243	0.99743	0.00256
16	0.97488	0.0251	0.99743	0.00256
18	0.95899	0.041	0.99617	0.00382

In this work, TFRC rate is set to 456 Kbps, Packet size is set to 1000 and TCP window size is set to 10 at different number of nodes.

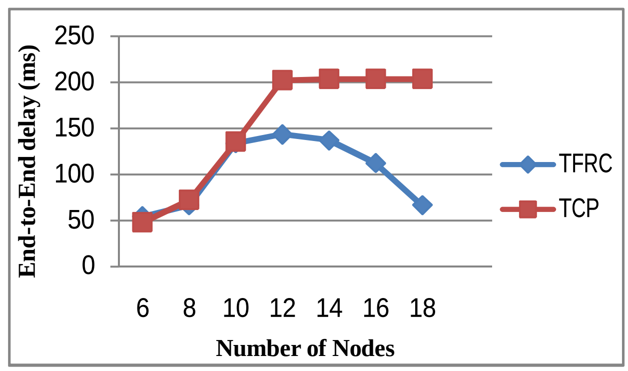

Figure 6: End-to-End Delay of TFRC and TCP in Wired Environment: Video Conference.

Figure 6 shows, the end-to-end delay of TFRC is less than the TCP. In this, the number of nodes set to be 6 to 18.

Table 2: Performance of TFRC and UDP in Wired Environment: Video Conference

No. of Nodes	TFRC		UDP	
	Packet Delivery ratio	Packet loss ratio	Packet Delivery ratio	Packet Loss ratio
6	0.99982	0.0001	0.99989	0.00011
8	0.99982	0.0001	0.99987	0.00012
10	0.97731	0.0226	0.98051	0.01948
12	0.96762	0.0323	0.97147	0.0285
14	0.95412	0.0458	0.96002	0.03997
16	0.94236	0.0576	0.95131	0.04868
18	0.92841	0.0715	0.94017	0.05982

In this, TFRC rate is set to be 456 Kbps, Packet size is set to be 1000 and UDP rate is set to 1Mbps at different number of nodes.

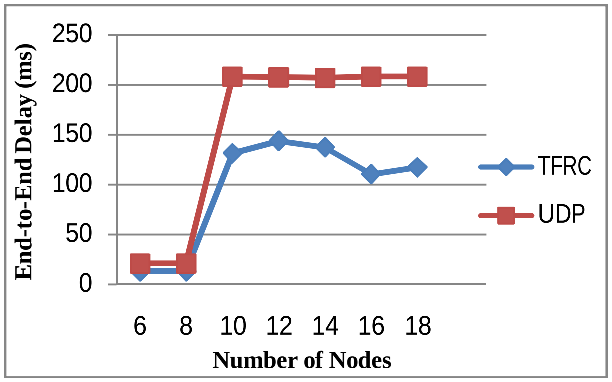

Figure 7: End-to-end delay of TFRC and UDP in Wired Environment: Video Conference

Figure 7 shows, the end-to-end delay of TFRC is less than the UDP. In this, the number of nodes is varied from 6 to 18.

Video Stream

Table 3 shows the performance of TFRC and TCP through packet delivery ratio and packet loss ratio.

Table 3: Performance of TFRC and TCP in Wired Environment: Video Stream

No. of Nodes	TFRC		TCP	
	Packet Delivery ratio	Packet loss ratio	Packet Delivery ratio	Packet loss ratio
6	0.99782	0.0021	0.99869	0.0054
8	0.98835	0.0116	0.99582	0.0060
10	0.98584	0.0141	0.98108	0.0070
12	0.98105	0.0189	0.96441	0.0096
14	0.97738	0.0226	0.94326	0.0143
16	0.9717	0.0282	0.98726	0.0127
18	0.96661	0.0333	0.98040	0.0195

In this, TFRC rate is set to be 2.5Mbps, Packet size is set to be 1000 and TCP window size is set to 10 at different number of nodes.

Table 4 shows the performance of TFRC and UDP through packet delivery ratio and packet loss ratio.

Table 4: Performance of TFRC and UDP in Wired Environment: Video Stream

No. of Nodes	TFRC		UDP	
	Packet Delivery ratio	Packet loss ratio	Packet Delivery ratio	Packet loss ratio
6	0.99663	0.0033	0.9953	0.0046
8	0.98755	0.012	0.98899	0.0110
10	0.98355	0.0164	0.98410	0.0158
12	0.97330	0.0266	0.97488	0.0251
14	0.96388	0.0361	0.96657	0.0334
16	0.95621	0.0437	0.95930	0.0406
18	0.94926	0.0507	0.95352	0.0464

In this, TFRC rate is set to be 2.5Mbps, Packet size is set to be 1000 and UDP rate is set to be 1 Mbps at different number of nodes.

5.2 Wireless Environment

In this section, we show the performance of real time applications like VoIP, Video conference and Video stream in wireless environment.

VoIP

Figure 8, Figure 9 and Figure 10 shows the VoIP traffic with an encoding rate of 84 Kbps and packet size of 160 bytes at different number of nodes.

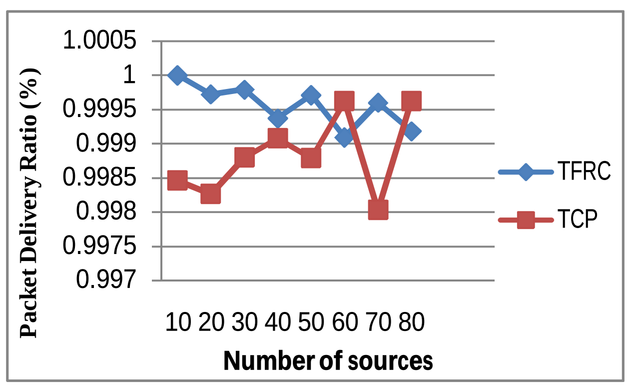

Figure 8: Packet Delivery Ratio of TFRC and TCP in Wireless Environment: VOIP

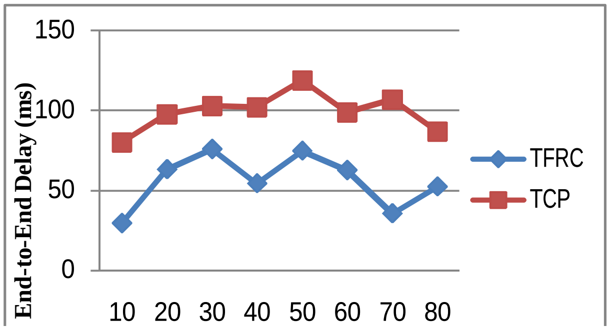

Figure 9: End-to-end delay of TFRC and TCP in Wireless Environment: VOIP

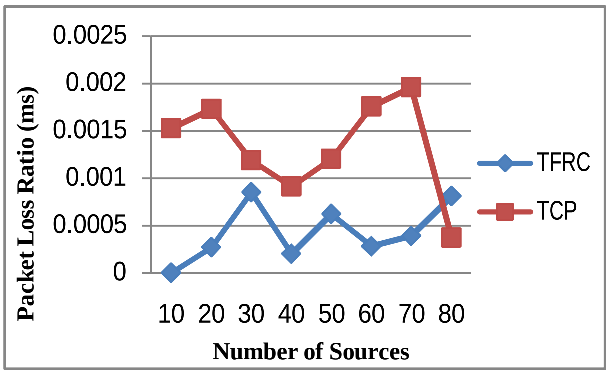

Figure 10: Packet Loss Ratio of TFRC and TCP in Wireless Environment: VOIP

The result shows packet delivery ratio of TFRC is more than TCP, end-to-end delay of TFRC is less than TCP and packet loss ratio of TFRC is less than TCP.

Table 5 shows the performance of TFRC and UDP through packet delivery ratio and packet loss ratio at varying pause times.

Table 5: Performance of TFRC and UDP at varying pause times in Wireless Environment: VOIP

Pause Time	TFRC		UDP	
	Packet Delivery ratio	Packet loss ratio	Packet Delivery ratio	Packet Loss ratio
10	1	0	1	0
20	1	0	1	0
30	0.99991	0.00008	0.99993	0.00006
40	0.9998	0.00016	0.99986	0.00013
50	0.99991	0.00008	0.99993	0.00006
60	1	0	1	0
70	0.99991	0.00084	0.99993	0.00006
80	1	0	1	0

In this, packet size is set to 160, TFRC rate is set to 84Kbps and speed set to 10 at varying pause times.

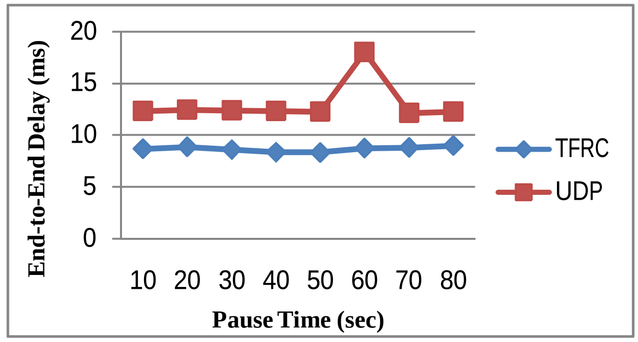

Figure 11 End-to-end delay of TFRC and UDP at varying pause times in Wireless Environment: VOIP

Video Conference

Table 6 shows the performance of TFRC and TCP through packet delivery ratio and packet loss ratio at varying speeds.

Table 6: Performance of TFRC and TCP at varying speeds in Wireless Environment: Video Conference.

Speed	TFRC		TCP	
	Packet Delivery ratio	Packet loss ratio	Packet Delivery ratio	Packet loss ratio
10	0.98875	0.01124	1	0
20	0.99016	0.00983	0.99842	0.0015
30	0.98134	0.01865	0.99885	0.0011
40	0.9812	0.01879	0.99789	0.0021
50	0.98782	0.01217	0.99930	0.0006
60	0.9929	0.00709	0.99771	0.0028

In this, the Packet size is set to 1000, TFRC rate is set to 512Kbps and pause time is set to 10 sec at varying speeds.

Figure 12, Figure 13 and Figure 14 shows the Video conference traffic with the rate of 512 Kbps and packet size of 1000 bytes at varying speeds.

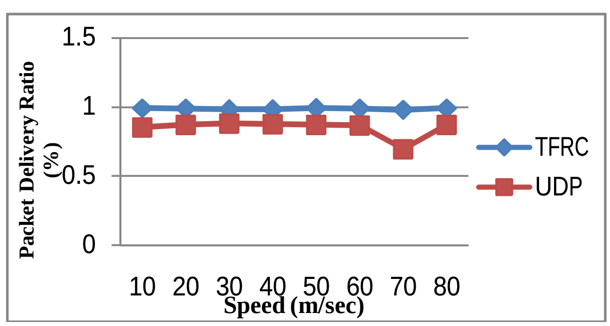

Figure 12: Packet Delivery Ratio of TFRC and UDP at varying speed in Wireless Environment: Video Conference

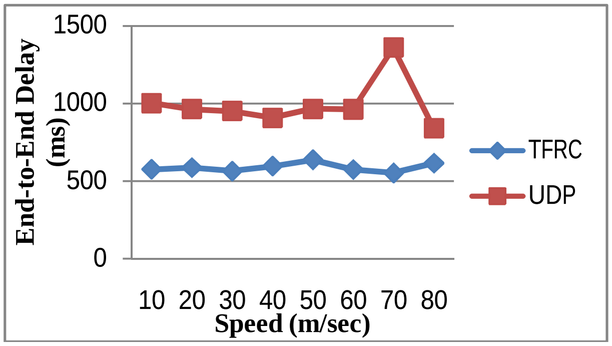

Figure 13: End-to-End Delay of TFRC and UDP at varying speeds in Wireless Environment: Video Conference

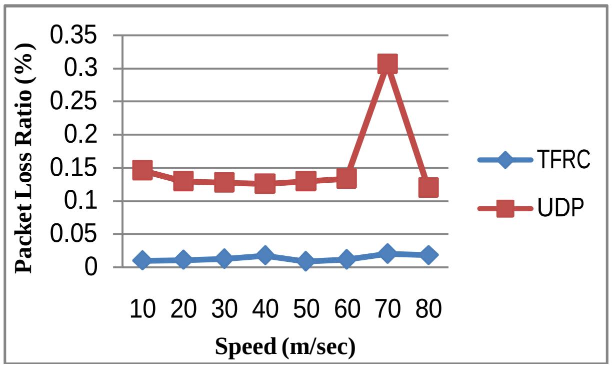

Figure 14: Packet Loss Ratio of TFRC and UDP at varying speeds in Wireless Environment: Video Conference

The result shows, packet delivery ratio of TFRC is more than UDP, end-to-end delay of TFRC is less than UDP and packet loss ratio of TFRC is less than UDP.

Video Stream

Figure 15, Figure 16 and Figure 17 shows the Video stream traffic with the rate of 2.5Mbps and packet size of 1000 bytes at different sources.

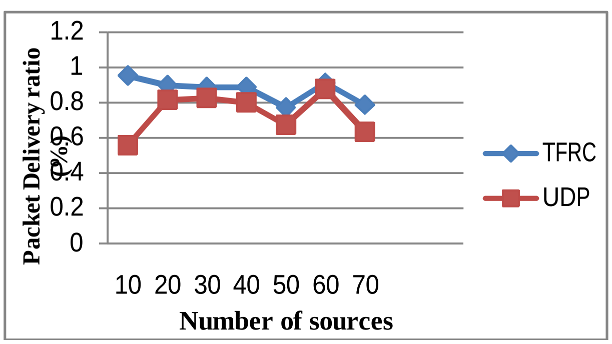

Figure 15: Packet Delivery Ratio of TFRC and UDP in Wireless Environment: Video Stream

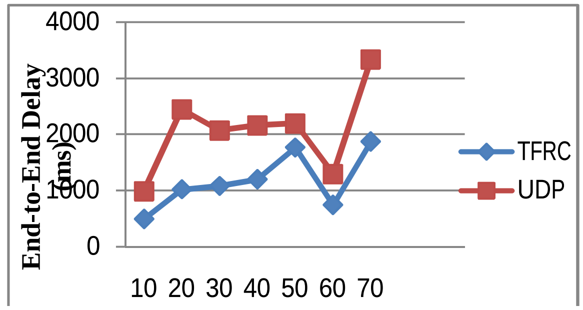

Figure 16: End-to-End Delay of TFRC and UDP in Wireless Environment: Video Stream.

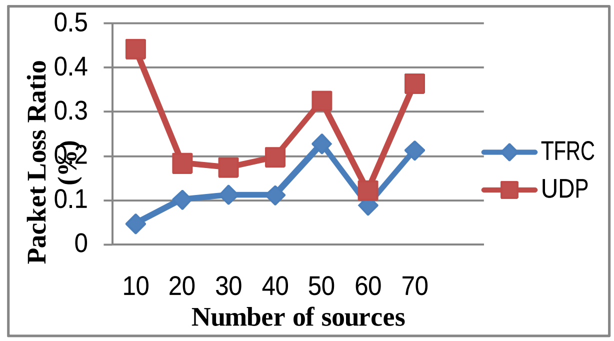

Figure 17: Packet Loss Ratio of TFRC and UDP in Wireless Environment: Video Stream

The result shows, packet delivery ratio of TFRC is more than UDP, end-to-end delay of TFRC is less than UDP and packet loss ratio of TFRC is less than UDP.

6. CONCLUSION

TFRC is designed to provide optimal service for unicast multimedia flows operating in the best-effort Internet environment. In this work we study TFRC as congestion control protocol for RTAs and shows that TFRC is better than TCP and UDP. From the results obtained, it can be concluded that TCP shares bandwidth fairly with TCP and TCP successfully addresses the congestion problem but it does not support RTAs. The average end-to-end delay in the simulation of TFRC is very less compared to simulation of UDP protocol over Real Time Applications. The packet loss ratio of TFRC is less when compared to UDP in Real Time Applications. The packet delivery ratio of TFRC is more when compared to UDP in real time applications.

REFERENCES

1. J. Postel. **Transmission Control Protocol**, *RFC-793*, September 1981.
2. J. Postel. **User Datagram Protocol**, *RFC 768*, August 1980.
3. Handley, Floyd, Widmer and Padhye. **TCP-Friendly Rate Control (TFRC): Protocol Specification**, *RFC 5348*, April 2008.

4. S. Floyd, J. Padhye, and E. Kohler. **Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)**, *RFC 4342*, March 2006.
5. Bin Zhou, Cheng Peng Fu, Victor 0. K. Li. **TFRC Veno: An Enhancement of TCP Friendly Rate Control over Wired/Wireless Networks**, *IEEE International Conference on Network Protocol*s, Beijing, China, 16-19 October 2007.
6. F. Beritelli, G. Schembra and G. Ruggeri. **Tcp-friendly transmission of voice over ip**, *in International Conference on Communication (ICC)*, New York, April 2002.
7. Evan Tan, Jing Chen, Sebastien Ardon, and Emmanuel Lochin. **Video TFRC**, *IEEE International Conference on Communications*, pp. 1767-1771, Beijing, China, May 2008.
8. J. Kempf and S. Floyd. **Iab concerns regarding congestion control for voice traffic in the internet**, *RFC 3714*, March 2004.
9. The VINT Project. **The ns Manual (formerly ns notes and documentation),** *http://www.isi.edu/nsnam/ns/ns-documentation.html*, November 2011.
10. Heekyoung Woo, Jong-won Lee and Seongho Cho. **ATFRC: Adaptive TCP Friendly Rate Control Protocol**, *International Conference on Information Networking (ICOIN)*, volume 2662, page 171-180, January 2003.