International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 16 (2015) pp 36872-36880

TCP, UDP, and TFRC in Wired Networks

M. Chenna Keshava
PG Scholar, Department of CSE,
JNTUA College of Engineering (Autonomous),
Pulivendula, A.P., India,
E-mail: keshava1047@gmail.com

P. Chenna Reddy,
Professor, Department of CSE,
JNTUA College of Engineering (Autonomous),
Pulivendula, A.P., India,
E-mail: pcreddy1@rediffmail.com

Abstract – Majority of the present applications have characteristics similar to Real Time Applications (RTAs). The applications face congestion where there is a bottleneck link in between source and destination. Different applications have different requirements. Reliability is essential for file transfer and financial transactions, whereas, delay and jitter are essential for streaming audio/video. All layers contribute to congestion control. The transport layer plays a major role in controlling congestion. The congestion problem is addressed by TCP, but is not suitable for majority of the streaming applications. Streaming applications are using UDP which doesn't support congestion control. A new protocol for datagram transport is designed i.e., TCP Friendly Rate Control (TFRC). In this paper performance comparison of transport protocols TCP, UDP, and TFRC is done in wired network.

Keywords: Wired network, TCP, UDP, TFRC, DCCP, performance.

I. INTRODUCTION

Wired networks are known for their high bandwidths. Among the networks that are used by the users of the Internet, wired networks have best characteristics. The layer which is directly related to user satisfaction is transport layer. TCP and UDP are the two transport protocols that are designed for wired networks. Because of the increasing demand for streaming applications TFRC is designed. From the inception TCP is the transport protocol used by many applications of Internet [1]. TCP is not only a reliable protocol but also controls congestion which has preserved the stability of the Internet. The trend has changed and majority of the streaming and real-time applications are starting to use UDP. UDP doesn't control congestion and is unreliable protocol [2]. The two transport protocols should share the Internet resources efficiently.

Congestion affects the network efficiency, which was successfully addressed by TCP. Key essential of the TCP for stability of the network is its mechanism of congestion control and reliability. But TCP is not suited to Real Time Application characteristics because of its over head.

By using Internet Protocol as underlying protocol, UDP makes available a datagram mode of packet-switched communication in computer networks. UDP uses minimal protocol mechanisms for data transfer and doesn't guarantee delivery of data, and reliable transfer. An application requiring reliable transfer and delivery of data uses TCP. But UDP lacks in congestion control mechanism.

TFRC [3] designed for streaming applications has congestion control mechanism similar to TCP but avoids the overhead associated with TCP. TFRC shares its bandwidth fairly with the other TCP flows and hence is called TCP friendly protocol.

Some of the applications require less variation in throughput like streaming media or telephony applications. TFRC provides low variation in throughput and also smoothness in its sending rate. Due to smoothness in throughput of TFRC, it responds slower than TCP when competing fairly for bandwidth. TFRC should be used only when smooth throughput is required for the applications. In response to a single packet drop, TCP halves its sending rate. Though some applications need to transfer the maximum amount of data in as short time as possible, for those kinds of applications TCP is suitable rather than TFRC. Since TFRC is a receiver based mechanism it uses loss event rate for the calculation of congestion control. TFRC-SP (Small-Packet TFRC) is a variant of TFRC which supports fixed sending rate by using variable sized small packets [4].

Many applications are evolving in day to day life, and these have their own requirements. In the transport layer the three transport protocols TCP, UDP and TFRC met some of the application requirements. But these need to be fine tuned in order to support new application requirements. Hence in this paper these transport protocols are analyzed in wired network with latest specifications of RFC 5166. NS-2.35 is used as simulator.

II. LITERATURE SURVEY

Arjuna Sathiaseelan, Gorry Fairhurst, [5] proposed the introduction of congestion control for multimedia traffic to ensure the stability of the next generation Internet. TFRC algorithm was first specified in RFC 3448 [6]. The specification in RFC 3448 poorly supported interactive multimedia applications, leading to common use of nonstandard congestion control methods and an incentive to use padding to guarantee the required media rate for bursty applications. From a network perspective, padding consumes unnecessary capacity and is therefore undesirable for other flows that share Internet bottleneck. It was thus important to revisit and revise the TFRC mechanism to support bursty media flows and make TFRC more suited to a wider range of multimedia flows.

Authors presented the detailed analysis, considering the performance of two new TFRC improvements designed to better support flows carrying bursty applications. Revised TFRC, specified in RFC 5348, also increases the sending rate compared to RFC 3448, but uses a different metric for calculating the allowed sending rate when an application has used less than the recent allowed rate. Simulation results demonstrate that both new methods allow TFRC to offer acceptable performance with bursty media over shared Internet paths. Although Faster Restart could benefit RFC 3448, the method does not offer significant additional benefit when used in combination with revised TFRC. Our analysis also evaluates the performance of revised TFRC and demonstrates that this can substantially improve the performance of other network traffic sharing a congested network. These results demonstrate that revised TFRC has addressed the principle performance shortfalls of TFRC for bursty applications and has removed the previous incentive for applications to use padding.

Xiao fu, Hu Ting, etc., [7] proposed a real-time video transmission system based on TFRC protocol and therefore the analysis model concerning the system in the framework is improved. It assesses the potency and quality of the video transmission according to the actual video file, and it analyzes loss of frames in different video types during transmission as well as the video quality in the receiver. They analyzed the real-time transport of MPEG-4 video supported UDP and TFRC. Simulation results shows that TFRC protocol is very for video transmission in a wired network, and quality assessment is also essential for a video transmission system. It assesses the quality and efficiency of the video transmission according to the actual video file, and analyzes different types of video frame losses during transmission as well as the picture quality in receiver. Simulation experiment results indicated that compared with the traditional TFRC, the proposed TFRC-JI suites well for real-time service transmission.

Agnieszka Chodorek and Robert R. Chodorek [8] suggest that although TFRC protocol is suitable for multimedia transmission, it is not aggressive enough to meet the QoS requirements of carried streaming media when it competes for bandwidth with the TCP. They proposed to substitute the original TFRC throughput equation with a linear throughput equation. This substitution makes TFRC more aggressive, which allows the protocol to preserve the real-time character of the transmitted flow no worse than the RTP or the UDP protocol. Moreover, in situations when the usage of the RTP causes the collapse of TCP transmission (or, at least, worsening of the QoS of one or more TCP flows), the proposed solution is “friendly” enough for competing TCP flows to equally share the remaining bandwidth. Such results allow us to believe that the proposed linear equation is more suitable for multimedia transmission than the equation originally included in the RFC 3448.

Shahrudin Awang Nor, Suhaidi Hassan, Osman Ghazali [9] show that UDP data flow can coexist with TCP data flow harmonically provided that UDP application data bit rate does not exceed the bandwidth left unutilized by TCP application. If UDP application data rate is set to use the maximum of the total bottleneck bandwidth together with the existence of TCP data flow, the congestion will happen and UDP will try to use the available bandwidth. This will cause the TCP application to be run out of bandwidth. In fact, UDP does not have any built-in congestion control mechanism to tolerate with other transport protocol flows when congestion happens. DCCP flow with TCP-Like or TFRC congestion controls can coexist fairly with TCP flow when congestion happens because of the congestion control mechanisms provided. DCCP can negotiate with TCP on how much bandwidth it will use based on the available bandwidth and try not to be selfish like UDP when competing with other flows. With this criterion, DCCP is a friendlier transport protocol to TCP and they can coexist together harmonically no matter whether the bottleneck link is congested or not. Nevertheless, DCCP with TCP-Like congestion control is slightly better in term of friendliness to TCP if compared with DCCP, TFRC when they coexist on fully utilized large delay link.

Zhaojuan Yue, Yongmao Ren, Jun Li, [10] mainly discussed the throughput, intra-protocol fairness, inter-protocol fairness, and implementation efficiency of RUBDP, Tsunami, UDT, and PA-UDP in point-to-point pattern. Among them, PA-UDP gets the optimal performance, and UDT is most convenient because it does not need to set some parameters such as the sending rate and buffer size.

With the development of e-science applications, communication patterns in high bandwidth-delay product network have changed from point-to-point (a single server transfers large amounts of data to a single client) gradually to point-to-multipoint and multipoint-multipoint structures. Single client gets data from servers distributed across different regions and then computes locally. Because of the change of network communication patterns, we need to consider the throughput, fairness among multiple flows and convergence when flows joining and departing.

Mohammad A. Talaat, Gamal M. Attiya, and Magdi A. Koutb [11] predicted that Video traffic is booming over Internet and to be the prevailing traffic type in the coming few years. TFRC is the most promising candidate congestion control algorithm over Internet that handles such type of traffic appropriately satisfying its QoS requirements.

Sunghee Lee, Hyunsuk Roh, Hyunwoo Lee, and Kwangsue Chung [12] presented that TFRC can balance between accomplishing the TCP friendliness task and allowing for some QoS constraints to be met. However, TFRC has problem in the high bandwidth delay product environment. TFRC inherits the slow-start mechanism of TCP Reno. However, if RTT is large, the slow-start mechanism takes quite a long time until a sender can fully utilize the available bandwidth on a path. This obstructs transmission of high quality video. Moreover, slow-start overshoots transmission rate. Overshooting of slow-start results in bursty packet losses and these losses degrade the quality of streaming service. TFRC also inherits the RTT-unfairness problem of TCP Reno. Therefore, when users are served multimedia streaming from servers that have different end-to-end propagation delays, a long RTT flow uses less bandwidth than a short RTT flow. Therefore, the long RTT flow receives lower quality video than the short RTT flow. To improve the performance of TFRC over the high bandwidth delay product environment, Enhanced TFRC is proposed. Enhanced TFRC includes a fast startup mechanism, and RTT-fair bandwidth estimation. Fast startup mechanism quickly increases transmission rate to find available bandwidth, and mitigates overshooting of the transmission rate, by using a concave increase function until the transmission rate reaches the concave threshold, and a convex increase function after the transmission rate is larger than the concave threshold. Enhanced TFRC also provides RTT-fairness, by only considering the delay caused by congestion, in estimating the transmission rate. Simulations show that the proposed scheme scans reduce the packet losses of slow-start, and provide RTT fairness.

III. SIMULATIONS

Simulation can be classified in to three cases. In the first case performance evaluation of TCP, in second case performance evaluation of UDP, and in third case performance evaluation of TFRC are performed. The topology, bandwidth, and propagation delay are as shown in Fig. 1. For both the TCP and TFRC, packet size is fixed at 1000 bytes and for UDP, packet size is 210 bytes. The total simulation time is 100 sec. NS 2.35 is used as simulator. For simulation of TCP, TCP window size is varied to increase the data rate. FTP is used as traffic source for TCP. For UDP constant bit rate (CBR) traffic is used and rate is varied. For TFRC also CBR traffic is used but interval between the packets is varied.

A. Simulation Environment

Dumbbell topology with multiple bottle neck links consisting of 10 nodes with different bandwidths like 5Mbps, 4Mbps, 10Mbps, and 2 Mbps with transmission delay 10ms is used.

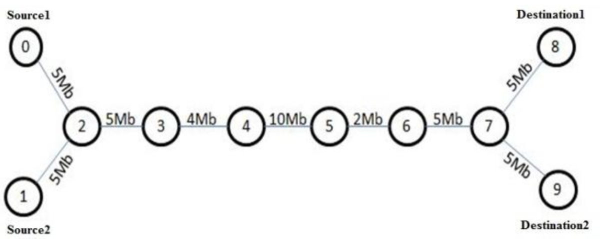

Fig. 1. Dumbbell topology

B. Performance Metrics

The performance parameters that are used are as follows:

Throughput:
Throughput is the rate at which a network sends or receives data. It is rated in terms of bits per second (bit/s).

Packet Loss Rate:
Packet loss rate is the ratio between number of packets dropped or lost and number of packets sent through the network.

Jitter:
Jitter is the differentiation between maximum delay and minimum delay of packets in the network.

End-to-end delay:
End-to-end delay is the time duration taken by a packet to travel from source to destination.

Fairness:
Fairness can be considered between flows of the same protocol and between flows using different protocols. It is the difference between bytes received by the two destinations.

C. Simulation Results and Analysis

Case 1: Simulation of TCP

Table.1. Packet loss rate, end-to-end delay, and jitter of TCP link 0-8 by varying window size

TCP Window Size	TCP Link between Node0-Node8				
	Sent (bytes)	Received (bytes)	Packet Loss Rate	End-to-End Delay (Sec)	Jitter (sec)
10	6462000	6454000	0.00123	0.083764	0.024960
20	11962000	11950000	0.00100	0.095899	0.049920
30	11970000	11956000	0.00116	0.178754	0.095344
40	11982000	11955000	0.00225	0.261435	0.178544
50	11153000	11125000	0.00251	0.206581	0.203504
60	11173000	11117000	0.00501	0.206704	0.203504
70	11184000	11113000	0.00634	0.210559	0.203504
80	11715000	11668000	0.00401	0.211151	0.211824
90	11338000	11253000	0.00749	0.207680	0.211824
100	10959000	10886000	0.00666	0.209426	0.207664

Table.2. Packet loss rate, end-to-end delay, and jitter of TCP link 1-9 by varying window size

TCP Window Size	TCP Link between Node1-Node9				
	Sent (bytes)	Received (bytes)	Packet Loss Rate	End-to-End Delay (Sec)	Jitter (sec)
10	6462000	6454000	0.00092	0.083785	0.033280
20	11962000	11950000	0.00091	0.095945	0.062559
30	11970000	11956000	0.00242	0.178858	0.112480
40	11982000	11955000	0.00300	0.261620	0.174544
50	11153000	11125000	0.00457	0.210124	0.203504
60	11173000	11117000	0.00448	0.210332	0.203504
70	11184000	11113000	0.00573	0.206787	0.203504
80	11715000	11668000	0.00794	0.210671	0.203504
90	11338000	11253000	0.00712	0.207646	0.203504
100	10959000	10886000	0.00644	0.208149	0.203504

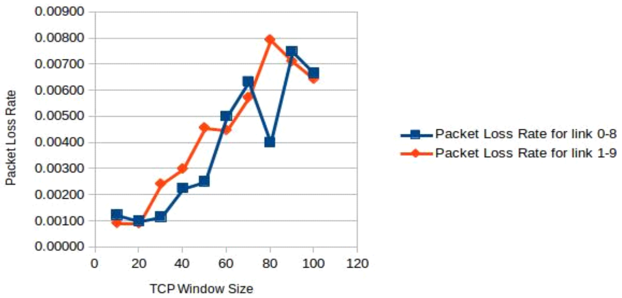

Fig. 2. TCP Window Size Vs Packet Loss Rate

Packet loss rate of TCP is increases with its window size, at initial window size loss rate is very low and it increases as window size increases as shown in Fig. 2.

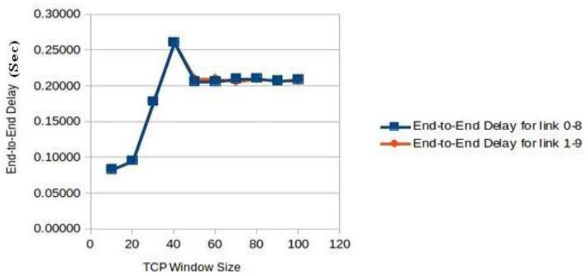

Fig. 3. TCP Window Size Vs end-to-end delay

End-to-End delay of TCP increases up to window size 40, after that remains constant. End-to-End delay of both the flows of TCP is similar. This is shown in Fig. 3.

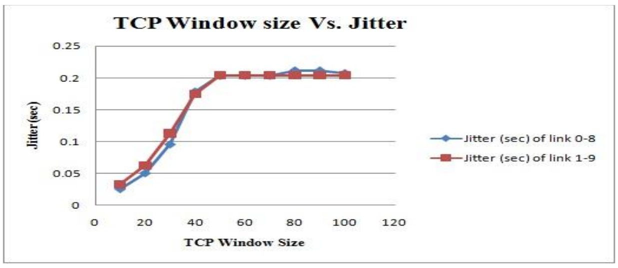

Fig. 4. TCP Window Size Vs Jitter

At initial window size Jitter of TCP is negligible and increases with window size as shown in Fig. 4.

Table.3. Evaluation of fairness by varying window size

TCP Window Size	Received (bytes) of link 0-8	Received (bytes) of link 1-9	Fairness (bytes)
10	6454000	6454000	0
20	11950000	11947000	3000
30	11956000	11941000	15000
40	11955000	11942000	13000
50	11125000	11101000	24000
60	11117000	11101000	16000
70	11113000	11098000	15000
80	11668000	10867000	801000
90	11253000	11150000	103000
100	10886000	11570000	-684000

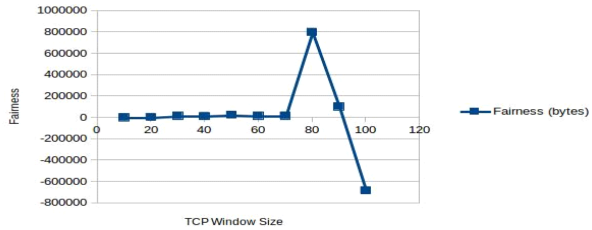

Fig. 5. Fairness between two links when TCP window size is varying

Fig. 5 shows that, TCP is fair with both flows and oscillates at high window size.

Table.4. Throughput when window size is 10

Simulation Time (Seconds)	Throughput (bits/sec) for link 0-8	Throughput (bits/sec) for link 1-9
10	0.0	0.0
20	494986.77395	494194.79511
30	507053.37512	507053.37512
40	511362.12624	511362.12624
50	513508.62850	513309.13019
60	514329.20198	514169.52201
70	515138.00240	514339.33883
80	515148.37505	515148.37505
90	515753.86355	515753.86355
100	516224.48999	516224.48999

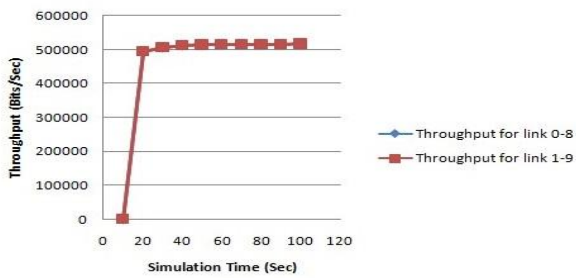

Fig. 6. Throughput of TCP When Window size is 10

Fig. 6 shows that, both the TCP flows are having approximately same throughput in entire simulation.

Table.5. Throughput when window size is 50

Simulation Time (Seconds)	Throughput (bits/sec) for link 0-8	Throughput (bits/sec) for link 1-9
10	0.0	0.0
20	821373.12676	800779.39359
30	866460.76449	858102.622068
40	884251.55162	874417.67503
50	878400.83566	873612.82293
60	886704.32139	880476.79239
70	890909.92345	886251.04891
80	886730.809524	883991.87266
90	980585.52127	887289.64573
100	892341.68393	891009.83067

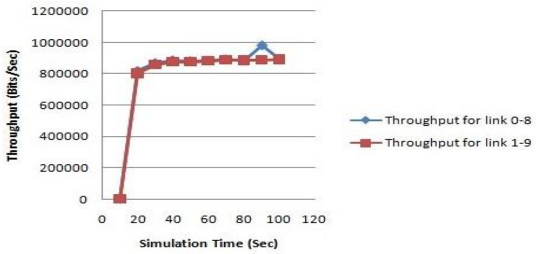

Fig. 7. Throughput of TCP when Window size is 50

Table.6. Throughput when window size is 100

Simulation Time (Seconds)	Throughput (bits/sec) for link 0-8	Throughput (bits/sec) for link 1-9
10	0.0	0.0
20	607511.27895	1004334.16130
30	699696.47978	1025266.28664
40	760123.59716	1003309.99275
50	785433.58846	973363.08817
60	851413.90276	936683.03706
70	880395.11557	905553.06490
80	893462.80574	898027.69426
90	895760.95037	895661.07737
100	886036.94913	914893.74926

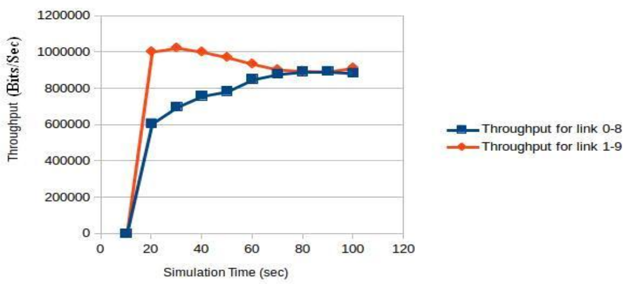

Fig. 8. Throughput comparison between two links when TCP window size is 100

At constant window size 10 and 50, throughput of both flows of TCP are approximately same and increases as simulation time progresses. But there is a slight variation in throughput at window size 100 and at the end of the simulation time throughput of both the flows are equal.

Case 2: Simulation of UDP

Here metrics are calculated by varying data rate from 0.5Mbps to 10Mbps.

Table.7. Packet loss rate, End-to-End delay, and Jitter of UDP between node-0 and node-8 by varying data rate

Data rate (Mbps)	UDP connection between Node0-Node8				
	Sent (bytes)	Received (bytes)	Packet loss rate	End to End delay (sec)	Jitter (sec)
0.5	6250020	6245610	0.00070	0.07277	0
1	12500040	12491010	0.00072	0.07277	0
2	25000080	20935110	0.16259	0.11392	0.04116
3	37500120	14610120	0.61039	0.15011	0.07756
4	50000160	14408940	0.71182	0.15020	0.07770
5	62500200	24971310	0.60046	0.15038	0.07812
6	75000030	24971310	0.66704	0.16663	0.09436
7	87500070	24971310	0.71461	0.16677	0.09444
8	100000110	24971310	0.75028	0.16672	0.09450
9	112500150	24971310	0.77803	0.16673	0.09454
10	125000190	24971310	0.80022	0.16680	0.09458

Table.8. Packet loss rate, End-to-End delay, and Jitter of UDP between node-1 and node-9 by varying data rate

Data rate (Mbps)	UDP connection between Node1-Node9				
	Sent (bytes)	Received (bytes)	Packet loss rate	End to End delay (sec)	Jitter (sec)
0.5	6250020	6245400	0.00073	0.07361	0
1	12500040	12490800	0.00073	0.07361	0
2	25000080	4046700	0.83813	0.11388	0.04032
3	37500120	10371690	0.72342	0.15052	0.07728
4	50000160	10572870	0.78854	0.15062	0.07728
5	62500200	10500	0.99983	0.10676	0.07694
6	75000030	10500	0.99986	0.10840	0.09324
7	87500070	10500	0.99988	0.10934	0.09336
8	100000110	10500	0.99989	0.11005	0.09324
9	112500150	10500	0.99990	0.11060	0.09333
10	125000190	10500	0.99991	0.11103	0.09324

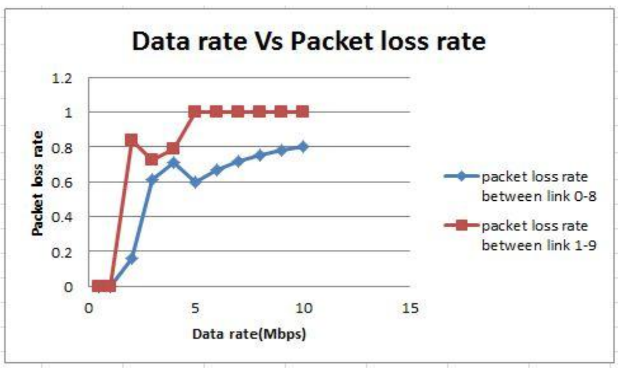

Fig. 9. Packet loss rate of UDP with varying Data rate

Packet loss rate of UDP is less at initial data rates and increases with data rate. As data rate increases packet loss rate of one flow is gradually increasing whereas that of other flow is constant.

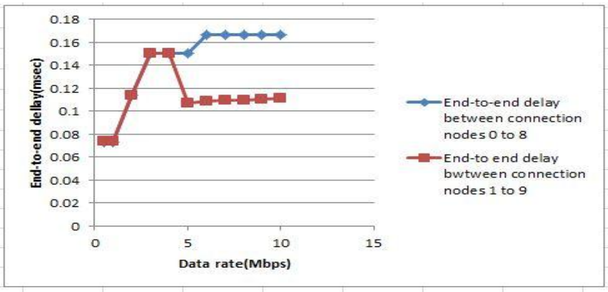

Fig. 10. End-to-end delay of UDP with varying Data rate

Up to data rate 5Mbps End-to-End delay of both flows are same. From 5Mbps onwards significant difference can be observed on both flows.

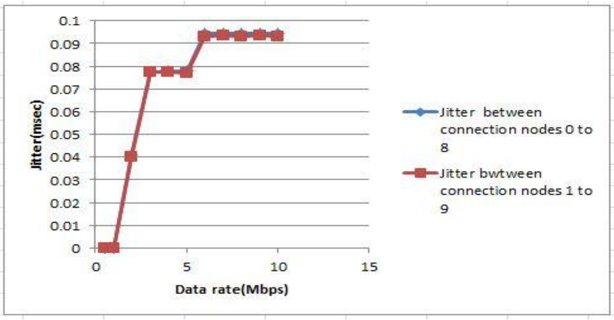

Fig. 11. Jitter of UDP with varying Data rate

Jitter of UDP is evaluated at different data rates and found that jitter is similar for both connections.

Table.9. Throughput of UDP when Data rate is 0.5Mbps

Simulation Time (sec)	Throughput (bits/sec) of Link Node0 to Node8	Throughput (bits/sec) of Link Node1 to Node9
10	150844.0081	134083.5627
20	496506.9860	496506.9860
30	498232.0426	498232.0426
40	498806.7834	498806.7834
50	499098.0054	499098.0054
60	499303.9939	499270.4611
70	499415.5408	499415.5408
80	499493.0684	499493.0684
90	499551.6122	499551.6122
100	499599.0015	499599.0015

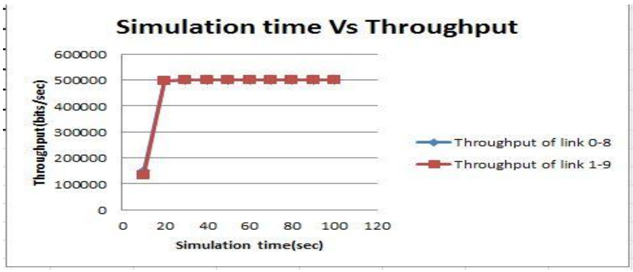

Fig. 12. Throughput of UDP when Data rate is 0.5Mbps

Table.10. Throughput of UDP when UDP Data rate is 5Mbps/10Mbps

Simulation Time (sec)	Throughput(bits/sec) of Link Node0 to Node8	Throughput(bits/sec) of Link Node1 to Node9
10	285462.9777	268671.0379
20	1977576.2376	8150.4950
30	1988665.4906	4095.5158
40	1992442.2752	2734.8779
50	1994297.6575	2094.7624
60	1995439.2132	1676.6424
70	1996181.0316	1397.6705
80	1996731.0706	1198.2878
90	1997143.3724	1048.6885
100	1997465.9267	932.2974

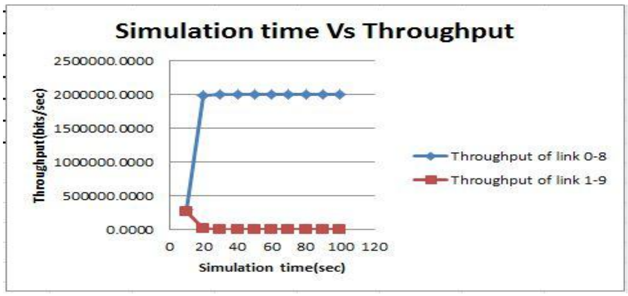

Fig. 13. Throughput of UDP when Data rate is 5Mbps/10Mbps

At constant data rate 0.5 Mbps throughput of UDP is constant, low compared to high data rates at 5 and 10 Mbps. At high data rates throughput of both flows has significant difference.

Table 11: Fairness of UDP with varying data rate

Data rate (Mbps)	Fairness (bytes)
0.5	210
1	210
2	16888410
3	4238430
4	3836070
5	24960810
6	24960810
7	24960810
8	24960810
9	24960810
10	24960810

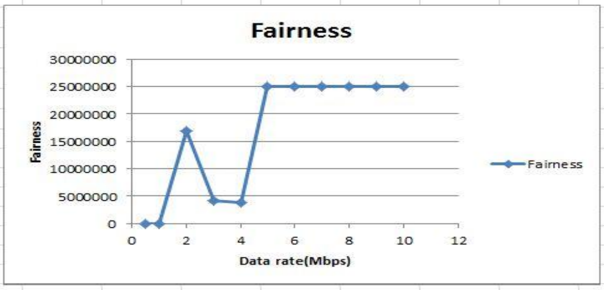

Fig. 14. Fairness with varying Data rate of UDP

The above figure represents fairness calculation of UDP at various data rates. From the results it is concluded that UDP is highly unfair.

Case 3: Simulation of TFRC

Table.12. Packet loss rate, End-to-End delay, and Jitter of TFRC between node-0 and node-8 by varying data rate

Data rate (Mbps)	TFRC link Node0 to Node8				
	Sent (bytes)	Received (bytes)	Packet loss rate	end-to-end delay (sec)	Jitter (sec)
1	1250000	1244500	0.0044	0.08656	0.285
2	1869200	1841100	0.0150	0.25427	0.461
3	1869200	1841100	0.0150	0.25427	0.461
4	1869200	1841100	0.0150	0.25427	0.461
5	1869200	1841100	0.0150	0.25427	0.461
6	1869200	1841100	0.0150	0.25427	0.461
7	1869200	1841100	0.0150	0.25427	0.461
8	1869200	1841100	0.0150	0.25427	0.461
9	1869200	1841100	0.0150	0.25427	0.461
10	1869200	1841100	0.0150	0.25427	0.461

Table.13. Packet loss rate, End-to-End delay, and Jitter of UDP between node-1 and node-9 by varying data rate

Data rate (Mbps)	TFRC link Node1 to Node9				
	Sent (bytes)	Received (bytes)	Packet loss rate	end-to-end delay (sec)	Jitter (sec)
1	1250000	1244500	0.0048	0.08692	0.26714
2	1869200	1841100	0.0092	0.25573	0.38265
3	1869200	1841100	0.0092	0.25573	0.38265
4	1869200	1841100	0.0092	0.25573	0.38265
5	1869200	1841100	0.0092	0.25573	0.38265
6	1869200	1841100	0.0092	0.25573	0.38265
7	1869200	1841100	0.0092	0.25573	0.38265
8	1869200	1841100	0.0092	0.25573	0.38265
9	1869200	1841100	0.0092	0.25573	0.38265
10	1869200	1841100	0.0092	0.25573	0.38265

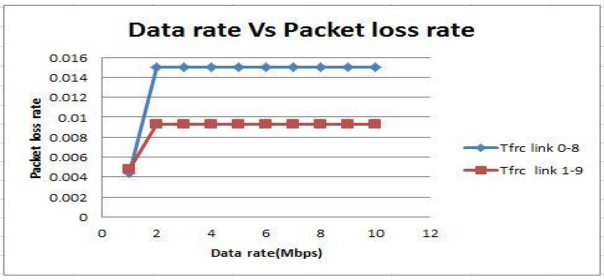

Fig. 15. Packet loss rate of TFRC with varying Data rate

Fig. 16. End-to-End delay of TFRC with varying Data rate

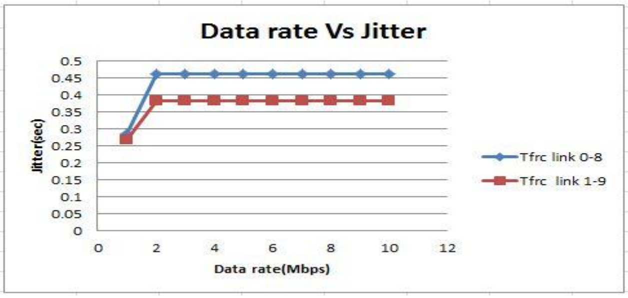

Fig. 17. Jitter of TFRC with varying Data rate

End-to-End delay, packet loss rate and jitter are measured by varying data rates. Those are represented in Fig. 15, Fig. 16 and Fig. 17. Packet loss rate and jitter of link 0-8 is high when compared to link 1-9. End-to-End delay is constant for both the links.

Table.14. Throughput of TFRC when Data rate is 1Mbps

Simulation Time (sec)	Throughput (bits/sec) of link 0 – 8	Throughput (bits/sec) of link 1 - 9
10	0.0	0.0
20	956023.70178	426131.52573
30	978302.81178	437012.40331
40	985225.43892	440653.83807
50	988924.69676	442492.43038
60	991128.14958	443588.52900
70	992602.53542	444321.74644
80	993663.49791	444849.00825
90	994556.80399	445243.44569
100	995159.58480	445548.786271

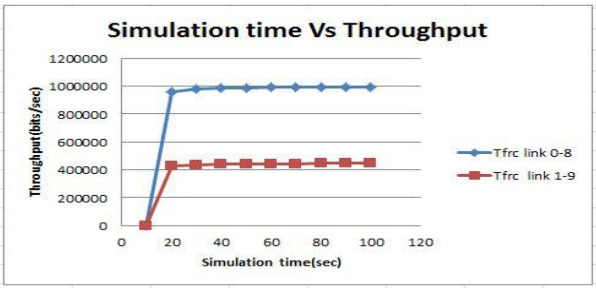

Fig. 18. Throughput of TFRC when Data rate is 1Mbps

Table.15. Simulation time vs. throughput when Data rate is 5Mbps /10 Mbps

Simulation Time (sec)	Throughput (bits/sec) link 0 to 8	Throughput (bits/sec) link 1 to 9
10	0.0	0.0
20	1408306.3739	326334.2103
30	1457096.0148	378502.6796
40	1454085.5091	415415.0339
50	1448370.4560	437902.6378
60	1456605.3908	439919.7382
70	1462210.1292	440992.4458
80	1466019.1349	441996.8947
90	1468861.7195	442845.7785
100	1471072.4344	443417.17394

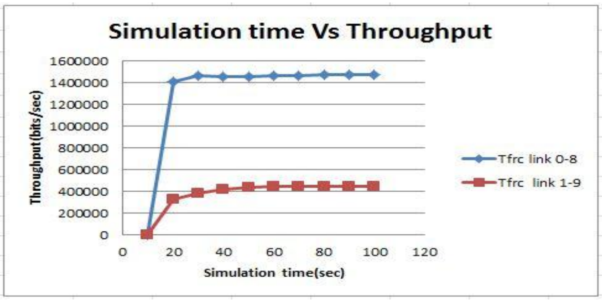

Fig. 19. Throughput of TFRC when Data rate is 5Mbps/10Mbps

At different simulation times and data rates 1, 5 and 10Mbps throughput is calculated. At 1Mbps Throughput of link 0-8 is high than link 1-9. Throughput of TFRC for both the links is similar at data rates 5Mbps and 10Mbps as shown in Fig. 18 and Fig. 19.

Table.16. Fairness between two links of TFRC

Data rate (Mbps)	Received (bytes) of link 0-8	Received (bytes) of link 1-9	Fairness (bytes)
1	12445000	5573000	6872000
2	18411000	5548000	12863000
3	18411000	5548000	12863000
4	18411000	5548000	12863000
5	18411000	5548000	12863000
6	18411000	5548000	12863000
7	18411000	5548000	12863000
8	18411000	5548000	12863000
9	18411000	5548000	12863000
10	18411000	5548000	12863000

Fig. 20. Data rate vs. fairness between two links of TFRC

Fairness of both links of TFRC is similar when evaluated at different data rates.

IV. CONCLUSION

In this paper performance comparison of TCP, UDP, and TFRC is performed. The metrics evaluated are packet loss rate, end-to-end delay, jitter, throughput, and fairness. TFRC has higher loss rate than TCP but less than UDP and its loss rate is stable. End-to-End delay of UDP is relatively less than other two. End-to-End delay of TFRC is more than that of TCP but it is same at different data rates. End-to-End delay of TCP varies with data rates. TFRC jitter is relatively (significantly higher, double) higher than TCP but its jitter is same for all data rates. Jitter of TCP varies with data rate and UDP has low jitter. TFRC is highly unfair; TCP is fair at most of the data rates, UDP fairness oscillates. UDP exploits the available bandwidth but is highly unfair. TCP fails to exploit the available bandwidth and its throughput is less than that of TFRC but is fair. TFRC throughput though better than TCP throughput is unfair.

REFERENCES

[1] Jon Postel, ransmission Control Protocol (TCP): Protocol Specification, IETF RFC 793, September – 1981.
[2] Jon Postel, User Datagram Protocol, RFC 768, August – 1980.

[3] J. P. S. Floyd, et all, Protocol Specification: TCP Friendly Rate Control (TFRC), IETF RFC 5348, September - 2008.

[4] E. Kohler, M. Handley, S. Floyd, Datagram Congestion Control Protocol (DCCP), RFC 4340, March – 2006.

[5] Arjuna Sathiaseelan, Gorry Fairhurst, TCP-Friendly Rate Control (TFRC) for bursty media flows, Computer Communications, Volume 34, Issue 15, 15th September 2011, Pages: 1836-1847.

[6] M. Handley, et all, Protocol Specification: TCP Friendly Rate Control (TFRC), IETF RFC 3448, January - 2003.

[7] Xiao fu, Hu Ting, etc., "A Novel Video Transmission Evaluation Framework based on TCP-Friendly Congestion Control Mechanism", IJCNIS, Volume 2, Number 2, December 2010, Pages: 19-25, ISSN: 2074-9104.

[8] Agnieszka Chodorek and Robert R. Chodorek, "Streaming Video over TFRC with Linear Throughput Equation", Advances in Electronics and Telecommunications, Volume 1 Issue 2, Pages: 26-29, November 2010.

[9] Shahrudin Awang Nor, et all, "Friendliness of DCCP towards TCP over large delay link networks", 2010 IEEE 2nd ICETC, Volume 5, Pages: 286-291, 22-24 June 2010, Shanghai, ISBN: 978-1-4244-6367-1, DOI: 10.1109/ICETC.2010.5530062.

[10] Zhaojuan Yue, Yongmao Ren, Jun Li, "Performance Evaluation of UDP-based High-speed Transport Protocols", 2011 IEEE 2nd ICSESS, Pages: 69-73, 15-17 July 2011, Beijing, ISBN: 978-1-4244-9699-0, DOI: 10.1109/ICSESS.2011.5982257.

[11] Mohammad A. Talaat, et all, "Enhanced TCP-friendly rate control for supporting video traffic over internet", CJECE, Volume 36, Issue 3, Pages: 135-140, 2013, ISSN: 0840-8688, Publisher: IEEE, DOI: 10.1109/CJECE.2013.6704695.

[12] Sunghee Lee, Hyunsuk Roh, Hyunwoo Lee, and Kwangsue Chung, "Enhanced TFRC for High Quality Video Streaming over High Bandwidth Delay Product Networks", JOURNAL OF COMMUNICATIONS AND NETWORKS, Volume 16, Issue 3, Pages: 344-354, JUNE 2014, ISSN: 1229-2370, Publisher: IEEE, DOI: 10.1109/JCN.2014.000055.