On performance evaluation of handling streaming traffic in IP networks using TFRC protocol

Kacper Kurowski and Halina Tarasiuk

Institute of Telecommunications, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
kacper.kurowski@gmail.com, halina@tele.pw.edu.pl

Abstract. This paper deals with the performance evaluation of handling streaming traffic in IP best effort networks using TFRC protocol. In our studies we check and discuss an influence of video on demand traffic and different network conditions on TFRC congestion control mechanism. Moreover, we check TFRC performance when it is sharing bottleneck with different versions of TCP. We illustrate our studies by simulation results performed in ns-2 simulation tool and compare results with those obtained for UDP.

Key words: IP network, TCP-Friendly, Video on Demand, Best-effort.

1 Introduction

In the last years we have observed the dynamic growth of standardization activity on the Internet towards the multi-service network based on Quality of Service (QoS) architectures, e.g. [20]. However, current version of Internet i.e., best effort network, lacks mechanisms guaranteeing QoS. Therefore, it is not possible to guarantee target values of throughput, IP packet delay (IPTD), packet losses (IPLR) or packet delay variation (IPDV) that are required by applications [21](e.g. audio, video on demand). Moreover, applications generate different types of traffic e.g. streaming or elastic and have different QoS requirements e.g. real time applications are delay sensitive while non-real time applications are loss rate sensitive. In order to provide better service for streaming and elastic traffic (to avoid congestion collapse and starvation of elastic flows by streaming transmissions) without building QoS networks new transport protocols such as TFRC (TCP-Friendly Rate Control [2], [15]) have been designed. TFRC is intended to provide congestion control mechanism for streaming traffic (e.g. audio, video). The protocol is reasonably fair while competing for bandwidth with TCP flows [1]. It is also responsible for maintaining smoothly changing sending rate, which is adjusted according to the changes in network conditions. TFRC may be implemented directly into the application or the transport protocol [7]. This protocol is receiver-based and to calculate allowed transmission rate uses TCP Reno throughput equation derived in [11] and [10]. The performance of TFRC congestion control mechanism has been intensively analyzed. However, in this paper we will present some aspects that have not been presented before. We analyze TFRC when it carries VoD traffic under different network conditions such as level of congestion, different RTT (Round Trip Time) values or TCP version used as background traffic. We will compare obtained results with simulations where only UDP was used. The reason to study TFRC performance for handling VoD streams is that TFRC uses throughput equation derived for TCP Reno to control transmission rate and, on the other hand, VoD is non-real time application.

The paper is organized as follows: section 2 provides related works dealing with performance evaluation of TFRC; section 3 contains studied system and problem statement; section 4 presents simulation results; section 5 concludes the paper.

2 Related Works

There are plenty of papers evaluating effectiveness of TFRC congestion control mechanisms: [4], [5], [16] and [17]. However, majority of them evaluate TFRC when RED (Random Early Detection) [6] was used for buffer management. It is important to emphasis that configuration parameters of RED have strong influence on its performance [19]. Moreover, based on our best knowledge RED is not amply used in current Internet. Therefore, in this paper, similar as in [17], we consider best effort environment where queues are drop-tail with FIFO (First In First Out) queuing discipline. In [4] the smoothness of TFRC throughput was evaluated and compared with the results obtained for AIMD (Additive Increase Multiplicative Decrease). The simulations have shown that TFRC maintains slowly changing sending rate and obtains better results in comparison with AIMD. Authors of [17] compared three TCP-Friendly protocols: GAIMD (Generalized Additive Increase Multiplicative Decrease), TEAR (TCP Emulation at Receivers), and TFRC in terms of friendliness, smoothness and quickness in responding to changes in network conditions. The simulations showed that TFRC maintains smooth sending rate only when loss rate is low and, when losses rise above 10^{-1} TFRC throughput changes rapidly and is close to zero. However, [17] did not consider the influence of RTT and traffic characteristics on the congestion control mechanism implemented in TFRC.

Based on the studies of current state of the art, we can conclude that there are hardly any publications concerning the impact of VoD traffic on the TFRC's performance. Only in [18] was considered the transmission of pre-encoded media using TFRC. Authors analyzed the impact of TFRC on user-perceived media quality. The results from their studies can be used by streaming media applications to assess the movie playback rate, which will be based on the current network conditions. We can conclude that until now researchers have focused on evaluation and adjustment of TFRC protocol and verification of its friendliness with TCP flows. The impact of the type of traffic carried by TFRC was not deeply analyzed and simulations were mainly carried with the continuous traffic resembling FTP application. Taking above into account, we propose to evaluate the impact of traffic generated by VoD server on the performance of congestion control mechanism built into TFRC. We analyze TFRC in the best-effort environment, where FIFO/DropTail queuing scheme is used.

3 Problem Statement

In this paper we study the impact of traffic characteristics on TFRC congestion control mechanism and the behavior of TFRC under different network conditions. In our studies we employed popular versions of TCP that exist in the Internet [8] because TFRC is supposed to share bandwidth with them. We performed simulations with TCP Reno [13] for which TFRC throughput equation was derived. Moreover, we considered TCP NewReno, which improves protocol performance in situations where there are multiple drops in a single congestion window [14], [3]. Furthermore, we used the most aggressive version of TCP – TCP SACK in which receiver explicitly informs sender about lost packets [12].

As first step of our studies we checked bandwidth sharing between TFRC and each of mentioned TCP versions. We assumed that both TFRC and TCP are associated with greedy source. We wanted to verify if TFRC stays friendly despite the version of TCP that it shares bandwidth with. However, due to space limit we present only main conclusions of these studies. We observed that, TFRC is the friendliest with TCP SACK, but its friendliness decreases along with the increase in RTT. Therefore, we found interesting to evaluate whether increase in delay has a similar effect on TFRC carrying VoD traffic. Moreover, we used different versions of TCP as background traffic for VoD transmission. As reference values for measured metrics obtained for VoD transmission, IPTD and IPLR, we used those proposed in [21]. The reference metrics are mean IPTD=1 s and IPLR=10^{-3}. Let us recall that IPDV is not critical for VoD. For this purpose we studied the following system (see Fig. 1).

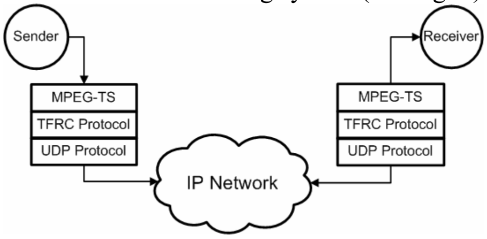


Fig. 1. Studied system for VoD transmission.

Data packets coming from the source are encapsulated by MPEG-TS (MPEG Transport Stream) ISO/IEC 13818-1 and sent to receiver by UDP with transmission speed controlled by TFRC (there is no transmission rate control in case of standard UDP). Finally, we compared results obtained for TFRC and UDP carrying VoD.

4 Simulations Results

As we have mentioned in previous paragraphs we study impact of different RTT values and traffic characteristics on performance of TFRC. Moreover, through experiments we verify whether using TFRC in best effort network can bring any advantage in comparison with actually used UDP/RTP for handling VoD application.

We prepared well known single bottleneck topology (Fig. 2). It consists of two IP routers R1 and R2 (with FIFO/drop-tail queue), a number of sources (it depends on evaluated scenario), associated with TFRC, TCP or UDP senders, and a number of receivers. In Fig. 2 we denoted by *X* link rate between R1 and R2 routers, by (d11,...., d1N), d2, and (d31, ..., d3N) we denoted propagation delay introduced by links between sender – router R1, router R1 – router R2, and router R2 – receiver, respectively. We changed link rate and delay depending on tested scenario.

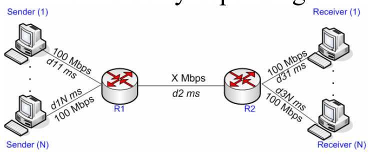


Fig. 2. Simulation topology.

For the simulations we used the ns-2 implementation [9] of TFRC with several modifications required for our needs. Especially due to the TFRC requirement for fixed packet sizes, a middle layer MPEG-TS was introduced to avoid sending small video packets of different sizes and, as a consequence, packets of fixed size (1316 bytes) were generated by the VoD source. Moreover, we modified TFRC source code in ns-2 to measure the value of delay introduced by the TFRC sender while buffering packets before transmission. To properly assess the impact of traffic characteristics on TFRC congestion control mechanism, we used traces [22] of two different movies ("V for Vendetta" and "Cinderella Man") with different peak to mean rates. The movies were previously captured and stored in the trace files. The first movie had much quicker changes of bit rate in comparison with the second one (on the frame level peak/mean ratio equals 18.2 and 12.1 respectively).

4.1 Impact of TFRC on VoD transmission

In this section we evaluate the impact of VoD traffic on TFRC congestion control mechanism. We performed simulations in the single bottleneck topology depicted in Fig. 2. We begin our studies from testing performance of single VoD connection in isolation. Therefore, we set the link rate between IP routers to 2 Mbps. The link is faster than the peak value of the movies' bit rates (mean rate about 640 kbps). To test the transmission over distinct network conditions we changed d2 delay values to 20 ms and 60 ms and router R1 buffer size to 20 and 100 packets. The simulations showed that using TFRC to carry VoD transmission slightly changes traffic characteristics. The influence of TFRC increases along with delay and it is also higher for movies with quick changes of bit rate ("V for Vendetta"). As one could expect, TFRC causes the increase of end-to-end packet delay, which in extreme case can be higher of about 1.5 seconds in comparison to UDP. The difference between mean IPTD values for TFRC and UDP grows with the increase of d2 delay value. The reason for greater delays for TFRC is the queuing time in the sender, which is required to shape outgoing traffic in accordance to the allowed value of transmission rate calculated at the sender. However, the increase of IPTD (IPTD is the sum of the packet queuing and end-to-end transfer time) leads to the decrease of IPLR for TFRC (see Table 1).

Table 1. Simulation results for considered movies

	Protocol	**TFRC**	**UDP**	**TFRC**	**UDP**
	d2	**20 ms**	**20 ms**	**60 ms**	**60 ms**
Cinderella Man	IPLR	none	1.13E-03	none	1.13E-03
	mean IPTD [s]	0.027	0.025	0.122	0.075
	max IPTD [s]	0.19	0.13	0.82	0.18
V for Vendetta	IPLR	8.59E-03	2.28E-02	4.30E-03	2.28E-02
	mean IPTD [s]	0.079	0.039	0.26	0.079
	max IPTD [s]	0.73	0.13	1.69	0.18

We observed, that the IPLR value for TFRC and UDP depends on the type of the movie transmitted (value of peak/mean ratio) and on the delay introduced by the network (in case of TFRC). When delay increases, IPLR decreases when TFRC is used. This was due to the fact that for greater delay, TFRC sender waited for a longer time with the increase of sending rate and, therefore, through the introduction of additional delay, reduced the number of lost packets. During the simulations we noticed that the upper limit (twice the receiving rate during last RTT) for the sending rate had some negative influence on VoD traffic. This effect consisted in the introduction of additional delay even though the sender could transmit packets faster due to availability of network resources. However, changing the upper bound for the sending rate caused the increase in IPLR. Nevertheless, the results that we obtained suggest that if there are enough resources for the transport of one VoD stream, TFRC mechanism has a positive impact on the service especially because of the reduction of IPLR value in comparison to UDP (see Fig.3).

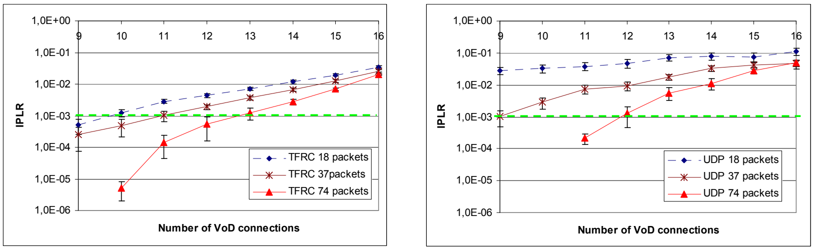
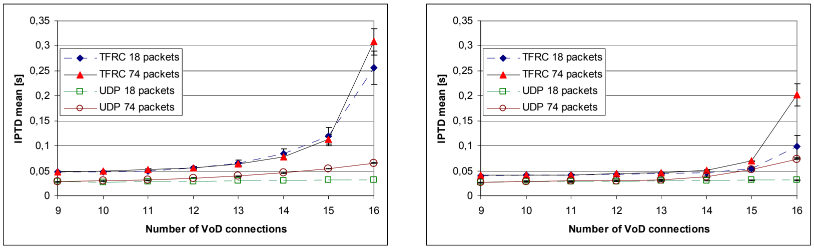


Fig. 3. IPLR for V for Vendetta movie, d2–20ms. Characteristics differ on buffer size expressed in packets (*TFRC – Left, UDP – Right*).

Having seen that TFRC does not disturb VoD transmission in different network conditions we verified how it deals with scenarios, where there are multiple connections. Therefore, we changed resource usage from about 50% to almost 100% by increasing the number of simultaneous VoD connections. We set the buffer size of bottleneck (10 Mbps) router R1 for three different values: 0.5BDP, 1BDP, 2BDP (Bandwidth Delay Product, BDP [8]) and compared the results obtained for TFRC and UDP. We noted that in our studies TFRC performed especially well in scenario, where the delay introduced by the bottleneck link was low (20ms) and buffers had small sizes (18 or 37 packets). Sample results are shown in Fig. 4. The results are presented with the 0.95 confidence interval. With the increase in buffer space or delay, TFRC and UDP obtained almost identical results.



(a) V for Vendetta (b) Cinderella Man

Fig. 4. Mean IPTD for d2–20ms. Characteristics differ on buffer size expressed in packets (18, 74).

TFRC packets always had higher value of IPTD due to the fact that packets were queued in the sender when equation based sending rate did not allow their transmission. We observed that IPTD value was greater for movies with greater peak/mean ratio such as “V for Vendetta”. It was due to the fact that quick changes in sending rate influenced the value of receiving rate, and receiver reported this and as a consequence limited the allowed sending rate. Nevertheless, comparing to UDP, TFRC allows transmitting more simultaneous VoD connections, which meet reference values of IPTD and IPLR (see section 3). We obtained better results for movies with quick changes in bit rate (“V for Vendetta”). The gain of additional VoD connections decreases while d2 delay or buffer size increases.

The results that we presented in this section show that in an environment with mainly VoD traffic, the use of TFRC reduces IPLR for tested connections and allows for simultaneous transmission of more VoD connections, which meet QoS requirements. The only drawback of TFRC is the increase of IPTD.

4.2 VoD transmission sharing bottleneck with TCP traffic

In the next simulations we used single VoD source and increased the number of simultaneous FTP sessions (TCP Reno or TCP SACK). This approach is based on the fact that the most amount of network traffic is TCP. Our simulations have shown that the use of TFRC reduces the value of IPLR for VoD traffic. However, the use of TFRC allows activating only one (two for TCP Reno) more TCP connections in comparison with UDP. Sample results (IPTD and IPLR as a function of number of TCP sources) obtained for “Cinderella Man” are presented in Fig. 5. From the graphs we see that, as in all previous simulations, IPTD increases when TFRC is used, which remains within the boundaries defined for VoD.

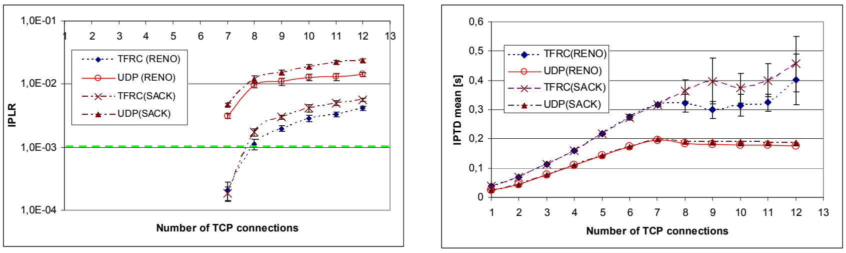


Fig. 5. IPLR and mean IPTD characteristics for Cinderella Man traffic trace sharing bottleneck with TCP (*IPLR – Left, mean IPTD – Right*), d2–20 ms , X–15Mbps.

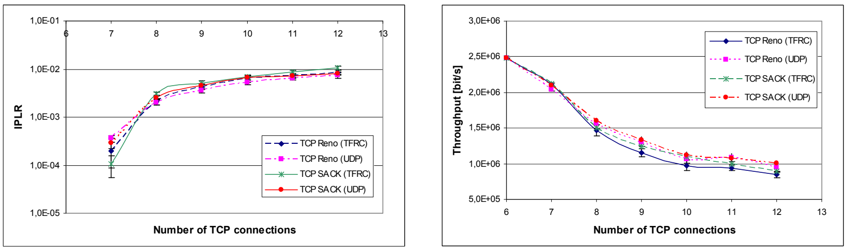


Fig. 6. IPLR and throughput characteristics for selected TCP flow (*IPLR – Left, mean IPTD – Right*), d2–20 ms, X–15Mbps.

Because in our experiments VoD traffic coexists with TCP connections, it is important to assess the impact that VoD has on the parameters of TCP packet transfer (Fig. 6). The results presented for TCP were obtained for one selected TCP connection. From Fig. 6 we can observe that TCP connections coexisting with VoD transmitted using TFRC have slightly higher IPLR values and lower throughput. This effect we may only observe in case where IPLR of the VoD trace exceeds reference values defined for VoD (see section 3). This is due to the fact that TFRC tends to obtain fair share of the bandwidth and its transmission is more aggressive in comparison with UDP. Nevertheless, the differences were within the error margin and can be considered negligible. From the obtained results we noticed that, due to the dynamic shaping of traffic by TFRC, its packets obtain higher end-to-end delay. The difference with UDP can be as high as 0.6 seconds in our studies. On the other hand, the use of TFRC for VoD reduces the packet loss rate.

5 Conclusions

In this paper, we considered the transmission of VoD traffic using TFRC protocol. Our results show that when RTT increases TFRC is able to reduce the IPLR value. However, TFRC performs better than UDP only when buffer size in router R1 and end-to-end delay are small. In this case, it allows transmitting more VoD connections at the same time or running more TCP connections taking into account the reference values of IPTD and IPLR. Moreover, when TFRC is used to transmit, VoD traffic has only negligible influence on TCP throughput. In the future, it can be interesting to compare the effectiveness of TFRC with shaped UDP stream and to verify results in real networks. We would like to consider TFRC for performance evaluation studies on multi-service QoS networks.

6 References

1. M. Handley, et al., TCP Friendly Rate Control (TFRC): Protocol Specification, Internet RFC 3448, January 2003.
2. M. Handley, et al., TCP Friendly Rate Control (TFRC): Protocol Specification, Internet draft, March 2007.
3. K. Fall, S. Floyd, Simulation-based Comparisons of Tahoe, Reno, and SACK TCP, SIGCOMM Comput. Commun. Rev, Vol. 26, No. 3, July 1996, pp. 5-21.
4. S. Floyd, M. Handley, J. Padhye, A Comparison of Equation-Based and AIMD Congestion Control, February 2000.
5. S. Floyd, et al., Equation-Based Congestion Control for Unicast Applications, ACM SIGCOMM 2000, Stockholm, 2000.
6. S. Floyd, V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Trans. on Netw., Vo.1 No.4, August 1993, pp. 397-413.
7. E. Kohler, M. Handley, S. Floyd, Datagram Congestion Control Protocol (DCCP), Internet RFC 4340, March 2006.
8. H. Lee, Soo-hyeong Lee, Yanghee Choi, The Influence of the Large Bandwidth-Delay Product on TCP Reno, NewReno, and SACK, ICOIN, Japan, January 2001.
9. The Network Simulator (NS-2). http://www.isi.edu/nsnam/ns/.
10. J. Padhye, et al., Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation, IEEE/ACM Trans. on Netw., Vol. 8, No. 1, 2000, pp. 133-145.
11. J. Padhye, et al., Modeling TCP Throughput: A Simple Model and its Empirical Validation, SIGCOMM Symp. on Commun. Architectures and Protocols, Vancouver, 1998.
12. M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP Selective Acknowledgement Options, Internet RFC 2018, October 1996.
13. M. Allman, V. Paxson, W. Stevens, TCP Congestion Control, Internet RFC 2581, April 1999.
14. S. Floyd, T. Henderson, The NewReno Modification to TCP's Fast Recovery Algorithm, Internet RFC 2582, April 1999.
15. S. Floyd, E. Kohler, TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant, Internet RFC 4828, April 2007.
16. J. Widmer, Equation-based Congestion Control, Master’s thesis, University of Mannheim, February 2000.
17. Yang Richard Yang, et al, Transient Behaviors of TCP-friendly Congestion Control Protocols, IEEE INFOCOM 2001, Anchorage Alaska, April 2001.
18. L. Xu, J. Helzer, Media streaming via TFRC: An analytical study of the impact of TFRC on user-perceived media quality, Computer Networks, Vo. 51, Issue 17, Elsevier December 2007, pp. 4744-4764.
19. T. Ziegler, C. Brandauer, S. Fdida, A quantitative Model for the Parameter Setting of RED with TCP traffic, IWQoS2001, Germany, June 2001.
20. ITU-T Rec.Y.2001, General Overview of NGN, 2004.
21. ITU-T Rec. Y.1541, Network Performance Objectives for IP-based Services, 2002.
22. VoD traces used: http://tnt.tele.pw.edu.pl/include/tools/vod_traces.php.