View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Open Archive Toulouse Archive Ouverte

Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of some Toulouse researchers and makes it freely available over the web where possible.

This is an author's version published in: https://oatao.univ-toulouse.fr/24113

Official URL: https://doi.org/10.1007/s12243-019-00717-7

To cite this version :

Tran Thai, Tuan and Lochin, Emmanuel and Lacan, Jérôme VIRAL : coupling congestion control with fair video quality metric. (2019) Annals of Telecommunications. 1-10. ISSN 0003-4347

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

VIRAL: coupling congestion control with fair video quality metric

Tuan Tran Thai[1] · **Emmanuel Lochin**[2] · **Jérôme Lacan**[2]

Abstract

Video streaming is often carried out by congestion controlled transport protocols to preserve network sustainability. However, the success of the growth of such non-live video flows is linked to the user quality of experience. Thus, one possible solution is to deploy complex quality of service systems inside the core network. Another possibility would be to keep the end-to-end principle while making aware transport protocols of video quality rather than throughput. The objective of this article is to investigate the latter by proposing a novel transport mechanism which targets video quality fairness among video flows. Our proposal, called VIRAL for virtual rate-quality curve, allows congestion controlled transport protocols to provide fairness in terms of both throughput and video quality. VIRAL is compliant with any rate-based congestion control mechanisms that enable a smooth sending rate for multimedia applications. Implemented inside TFRC a TCP-friendly protocol, we show that VIRAL enables both intra-fairness between video flows in terms of video quality and inter-fairness in terms of throughput between TCP and video flows.

Keywords Video streaming · Congestion control · Flow rate fairness · Video quality fairness

1 Introduction

Multimedia services have a rapid growth thanks to the evolution of technologies such as high-speed networks (ADSL, Wi-Fi, 3G, LTE) and video-enabled devices (laptop, smartphone, tablet etc.). Traditionally, live multimedia traffic is recommended to be delivered over UDP. Unlike TCP, UDP does not introduce End-to-End (E2E) latency resulting from in-order and reliable delivery. This property makes UDP suitable for interactive real-time applications such as Voice over IP (VoIP) and video conferencing. On the other hand, most of commercial progressive download and streaming solutions (e.g. Apple HLS, Microsoft HSS, DASH) use TCP as underlying transport protocol because the end users can tolerate a short start-up delay for buffering. A challenge is thus to deliver multimedia content in a congestion controlled manner while respecting video delivery constraints. Indeed, a TCP source adjusts its sending window (i.e. the maximum amount of consecutive packets TCP is allowed to send) to prevent congestion. The resulting variable sending rate of this window-based mechanism is an issue for video applications with strong delay constraint. Although TCP AIMD (additive increase multiplicative decrease) principle allows TCP flows to reach a steady state, its saw-tooth behaviour prevents multimedia application to adapt efficiently the sending rate. The resulting buffering at the sender side might violate the application delay constraint making TCP able to support real-time traffic (e.g. live streaming) only if the fair share is at least twice bigger than the source bit rate [1]. This explains why the support of real-time applications has turned towards protocols allowing out-of-order delivery and rate-based congestion control such as TCP-friendly rate-based control (TFRC) [2]. TFRC is a rate-based congestion control mechanism specifically designed to carry multimedia traffic. This protocol is the first transport mechanism for such traffic due to its smooth sending property [3, 4]. TFRC allows applications that use a fixed packet size to compete fairly with TCP flows using the same packet size. Although this protocol has not really been adopted as a kernel transport layer solution, TFRC got a certain success as an application layer transport mechanism, often implemented at a user-level on top of UDP such

✉ Emmanuel Lochin
emmanuel.lochin@isae-supaero.fr

Tuan Tran Thai
tuan.tran@expway.com

Jérôme Lacan
jerome.lacan@isae-supaero.fr

[1] Expway, Paris, France

[2] ISAE-SUPAERO, Université de Toulouse, Toulouse, France

as MULTFRC [5] and QSTP [6] or inside other user-level protocols [7].

If some real-time applications such as VoIP found a satisfying solution in TFRC, video conferencing, which is characterized by a variable bit rate and a variable packet size, experiences severe performance issues when its sending rate is controlled by TFRC. As TFRC acts as a token bucket, the burst of packets has to be queued at the sender side before it can be entirely sent, thus impairing the interactivity and inducing losses in case of stringent delay constraint. The usual way to counter this drawback is to use padding and constantly transmit at the burst rate (e.g. I-frames packet rate in case of video). Obviously, it requires the fair share to be much bigger than the application source rate and it reduces the overall network goodput.

Another main objective of the above transport protocols is keeping the fairness among multiple homogeneous/heterogeneous connections in the network. In fact, fair share of network resources among multiple heterogeneous connections is one of key issues especially for the commercial use of the Internet [8] which is inadequate when transmitting video communication flows.

There exist proposals tackling the resource allocation problem for multiple media streaming such as multimedia streaming TCP-friendly protocol (MSTFP [9]) or scalable streaming video protocol (SSVP [10]), which operates on top of UDP and will further be discussed in Section 4. In these approaches, fairness is often addressed in throughput and video quality is not explicitly considered. On the contrary, the authors in [11] proposed Q-AIMD, a congestion control enabling fairness in terms of video quality instead of throughput as for TCP. Q-AIMD is based on AIMD principle but targets video quality instead of congestion window. In the absence of congestion, there is an additive increase in video quality while the video quality is decreased by a coefficient in case of congestion. Q-AIMD is a decentralized approach which does not require the exact information on how many users are competing over the same bottleneck. The main drawback is that Q-AIMD does not behave fairly with TCP and this motivates the present contribution.

In this article, we address the same problem but from a centralized perspective. We propose VIRAL which is *the horizontal average rate-quality curve* (e.g. rate-distortion curve) *from rate-quality curves of all multimedia flows that share the same bottleneck link*. This approach is based on rate-based protocols (such as TFRC) that allow a fair share with TCP flows in terms of throughput. The objective of this approach is to provide *both intra-fairness in terms of video quality* and *inter-fairness between multimedia flows and non-multimedia flows in terms of throughput*.

We first present the rationale of VIRAL idea along with the assumptions and implementation in Section 2. Then, simulation results and analysis are presented in Section 3. Finally, we discuss this proposal and conclude this work in Section 5.

2 VIRAL: virtual curve for fair video quality

Each video has a characteristic specified by the rate-distortion curve which can be easily transformed to rate-quality curve. Each point on the rate-quality curve corresponds to an average quality value (e.g. PSNR) at a specific rate for the whole video sequence. For all video flows that share the same bottleneck link, there exists a virtual average curve that serves as the fairness curve in terms of video quality. In other words, if we consider n rate-quality curves denoted $C_1, C_2, ...C_n$; for each point $(R_i, Q_i) \in C_i$ where Q_i is the quality obtained from a given bit rate R_i; and $(\overline{R}_{vir}, Q) \in C_{vir}$, we define such virtual average curve as $R_{vir} = \frac{1}{n} \sum_{i=1}^{n} R_i$.

To better illustrate the idea, assume there exists a content service provider (CSP) that serves two different video contents (Crew and Harbour videos sequences are taken as reference) to two different clients. This video server is running a rate-based congestion control mechanism which targets throughput as fairness criteria. If this mechanism sends a bit rate R_{vir} for each client to be fair in throughput as shown in Fig. 1 and to not overload the capacity $\mathcal{K}$ of the output server link (i.e. $\sum R_i < \mathcal{K}$), the video server can compute the corresponding video quality Q for each video following R_{vir}. However to achieve the same quality Q, the client watching Crew video only needs a bit rate R_{crew} while the other client needs a higher bit rate R_{harbor}. Therefore, if the video server sends the respective rates R_{crew} and

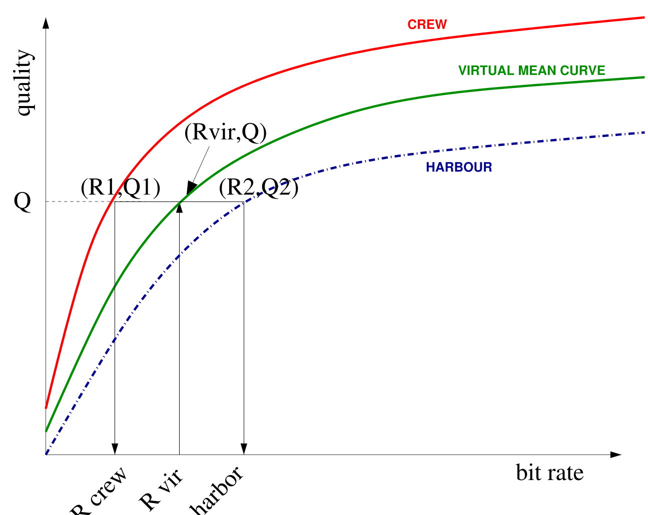


Fig. 1 Example of virtual mean curve

R_{harbor} to both clients instead of R_{vir} for each, the two clients can obtain on average the same video quality.

The rationale behind VIRAL is that if *any rate-based congestion control mechanisms* (e.g. TFRC) provide *a fair share with TCP flows* in terms of throughput, the *virtual curve approach is able to provide both intra-fairness and inter-fairness properties*. The intra-fairness indicates the fairness between video flows in terms of video quality while the inter-fairness refers to the fairness between all video flows and non-video flows in terms of throughput.

2.1 Assumptions

VIRAL assumes the video server operator has the knowledge of the number of its video subscribers and the rate-quality curve of each video stored. We explain later how the rate-quality curve of each video can be obtained from rate-distortion models. This assumption follows CDNs and video-on-demand (VoD) providers such as YouTube [12], NETFLIX or any content service provider (CSP) offering VoD service to clients [13–15]. Basically, the architecture is defined as shown in Fig. 2. Note that these providers are based on streaming technologies such as DASH and HLS. The video content is split into multiple segments of several seconds where each segment is delivered over HTTP/TCP. Another important point is that these streaming technologies are based on a receiver-driven approach : the receivers attempt to its best quality according to its instantaneous network quality regardless of other receivers. In that case, fairness in throughput or quality is not relevant. On the contrary, VIRAL aims at proposing a novel video service which aims at combining fairness both in throughput and in quality.

We also assume that there exists a rate quality of a video denoted R which characterizes the video as previously represented in Fig. 1. The way to produce these curves is out of scope of the present study but basically follows a statistical analysis as illustrated in Fig. 3. Considering [16, 17], the relationship between the rate R and the $PSNR$ can be transformed to the following equation:

$$R = \alpha . e^{\beta . PSNR} \tag{1}$$

where (α, β) represents the video characteristics. Note that for a given video, other classification methods can be used to obtain the corresponding rate-quality curve [18].

To demonstrate the rate-quality relationship, we perform the following experiments: the reference videos in CIf format (e.g. "Akiyo", "Foreman") are encoded using x.264 encoder with different target bit rates to obtain the corresponding quality in PSNR. Figure 3 shows the rate-PSNR curves obtained by these reference videos following an x.264 encoding and their respective fitting curves based on (1).

2.2 Implementation

We implemented VIRAL within TFRC protocol in ns-2 following Alg. 1. The PSNR is used as the video quality

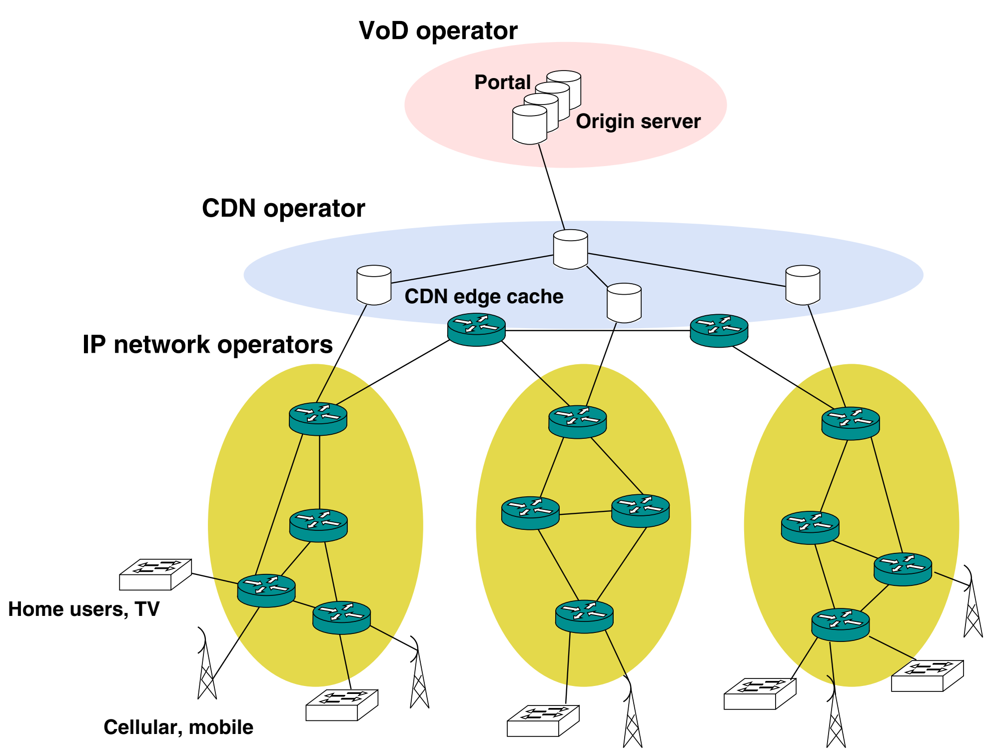


Fig. 2 Example of CSP (e.g. YouTube, NETFLIX) over a CDN operator (e.g. Akamai). Source https://www.slideshare.net/Netmanias/netmanias20130904content-networking-trends2013

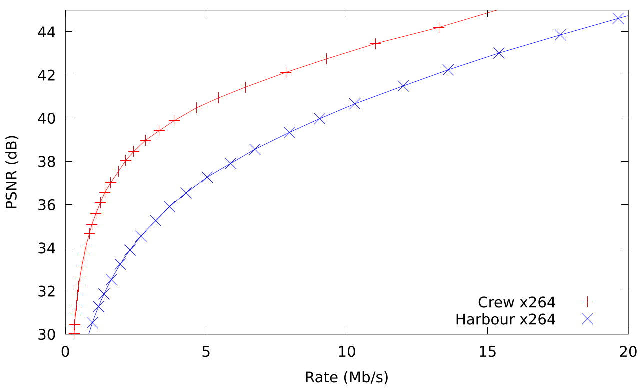


Fig. 3 Rate-PSNR curves for Crew and Harbour in CIf format

metric. Upon reception of a TFRC feedback from the k^{th} receiver (every RTT), the video sender first translates its current sending rate $R_{curr}(k)$ to the quality value $Q(k)$ from the rate-quality curve C_i. Then, following the virtual curve (Fig. 1), the video sender translates this quality value to the virtual rate R_{vir}. If R_{vir} is higher (resp. lower) than the rate indicated in the TFRC feedback R_{rcv}, the video sender decreases (resp. increases) its sending rate. The rate increase/decrease follows the rate increase/decrease principle of a rate-based congestion control mechanism (TFRC in this implementation). The new virtual rate R'_{vir} is translated back to new quality Q' using the virtual curve. The new rate R_{new} obtained after translation from Q' using rate-quality curve C_i is updated for client k^{th}. It is worth noting that the VIRTUAL algorithm can be applied to *any rate-based congestion control mechanisms and any video quality metrics*.

Algorithm 1 Virtual curve inside TFRC protocol.

Upon feedback reception from the k^{th} receiver with $R_{rcv}(k)$
Translate current sending rate $R_{curr}(k)$ to quality Q using rate-quality curve of k^{th} receiver
Translate Q to R_{vir} using virtual curve
if $R_{rcv}(k) > R_{vir}$ **then**
    Increase rate to R'_{vir}
else
    Decrease rate to R'_{vir}
end if
Translate new R'_{vir} to new Q' using virtual curve
Translate Q' to new rate $R_{new}(k)$ using rate-quality curve of the k^{th} receiver
Send the video at rate $R_{new}(k)$

3 Simulation results and hypothesis

As TCP fairness is considered over a long-run experiment, we present the cumulative average values. These values

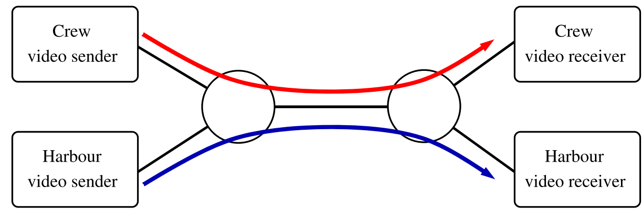


Fig. 4 Topology used in the first tests

correspond to an average throughput (or PSNR) over $t = [0, x]$ seconds. This means the last point of the curve is the average throughput over the whole experiment duration (i.e. rate transfer at 600 s). In a general manner, $f(x)$ is the average throughput after x seconds of experiments.

We used an experimental design approach [19] to perform our simulations. Basically, we performed hundreds of experiments with various conditions[1] and we selected some of them to illustrate the way our algorithm behaves. We also varied the simulation seed. However as we do not have any random variable in our scenario (for instance this would have been an impact if we had used RED queuing algorithm for instance), the standard deviation obtained was extremely weak. We have provided these explanations inside the revised version.

In the simulations, We choose TCP NewReno as TCP variant in order to show the TCP-friendliness. Indeed, when TFRC was proposed, there were no advanced TCP variants such as TCP CUBIC or TCP compound which are set as default TCP congestion control algorithm in Linux-based OS and Windows OS, respectively. The reference videos are Crew and Harbour in 4CIf format. The base RTT is set to 100 ms and the video transmission lasts 600 s. For the simulations where the flows (both TCP and video) share the same bottleneck link, we use a standard butterfly network topology as shown in Fig. 4.

The purpose of these simulations and the choice of this bottleneck size are linked to the size of the video(s) and to the whole traffic injected. Basically, we seek to obtain curves to illustrate the consistency and the behaviour of the algorithm. As a matter of fact, we adjust the bottleneck capacity, specified for each simulation, to obtain reasonable video quality for each flow. Considering a small bottleneck is equivalent to consider that the whole CDN backbone capacity is loaded by a continuous aggregate of video flows. We believe it does not matter to simulate further but a real implementation over an emulated testbed would bring out supplementary results in particular concerning the scalability (in terms of CPU and memory) of this proposal.

[1] https://personnel.isae-supaero.fr/emmanuel-lochin/projects-and-softwares.html

3.1 Intra-fairness with two video flows sharing the same bottleneck

Crew and Harbour videos are well-known reference videos commonly used by the multimedia community to weight up video quality in the context of multimedia research or benchmarking. These videos are freely available [20]. To sum up, Harbour video contains much more information than Crew and as a result, is more difficult to compress. In other words, to reach the same level of quality, the file obtained by compressing Harbour video is bigger than for Crew video. The aim of using such videos allows to be comparable with other quality assessments.

In this simulation, the two video flows share the same bottleneck capacity of 7 Mb/s. Figure 5 shows the cumulative throughput and the corresponding PSNR of Crew and Harbour videos. We observe that both videos achieve a relatively equal fair share in terms of PSNR with a discrepancy lower than 1 dB at the end of simulation while there is a significant difference in throughput. This result illustrates that VIRAL approach can obtain the fairness in terms of video quality.

3.2 Intra-fairness and inter-fairness with two video flows sharing the same bottleneck with a non-video flow

In this simulation, the bottleneck capacity is set to 10 Mb/s. Figure 6 a and b show the results of two video flows versus one TFRC flow. Figure 6 shows that the two video flows achieve good intra-fairness in terms of PSNR and a quite good inter-fairness in terms of throughput (Fig. 6). The two video flows obtain an average throughput of 3.15 Mb/s while TFRC achieves an average throughput of 3.64 Mb/s. The results from Fig. 6 c and d show good intra-fairness among video flows and good inter-fairness between video and TCP flows starting at $t = 300\,s$. This pace of convergence is explained by the different behaviour of both TFRC and TCP protocol. TFRC has a smooth behaviour compared with TCP which is much more aggressive and opportunistic with its saw-tooth behaviour. TCP starts to be fair at $t = 300\,s$ while TFRC might converge a bit later than the experiment duration. To confirm that this fact is not due to VIRAL, Fig. 7 shows the fairness in terms of throughput between two TFRC flows and one TCP flow for the same network settings.

3.3 Intra-fairness and inter-fairness with two video flows sharing the same bottleneck with two non-video flows

The bottleneck capacity is set to 12 Mb/s in this simulation. The four top (resp. bottom) sub-figures in Fig. 8 show the throughput and PSNR between two video flows versus two TFRC flows (resp. versus two TCP flows). From Fig. 8 c, we observe that both video flows achieve good inter-fairness against two TFRC flows where the throughput difference is less than 0.5 Mb/s. Additionally, the good intra-fairness among video flows in terms of PSNR is also obtained with a difference around 1 dB (Fig. 8d). Furthermore, the

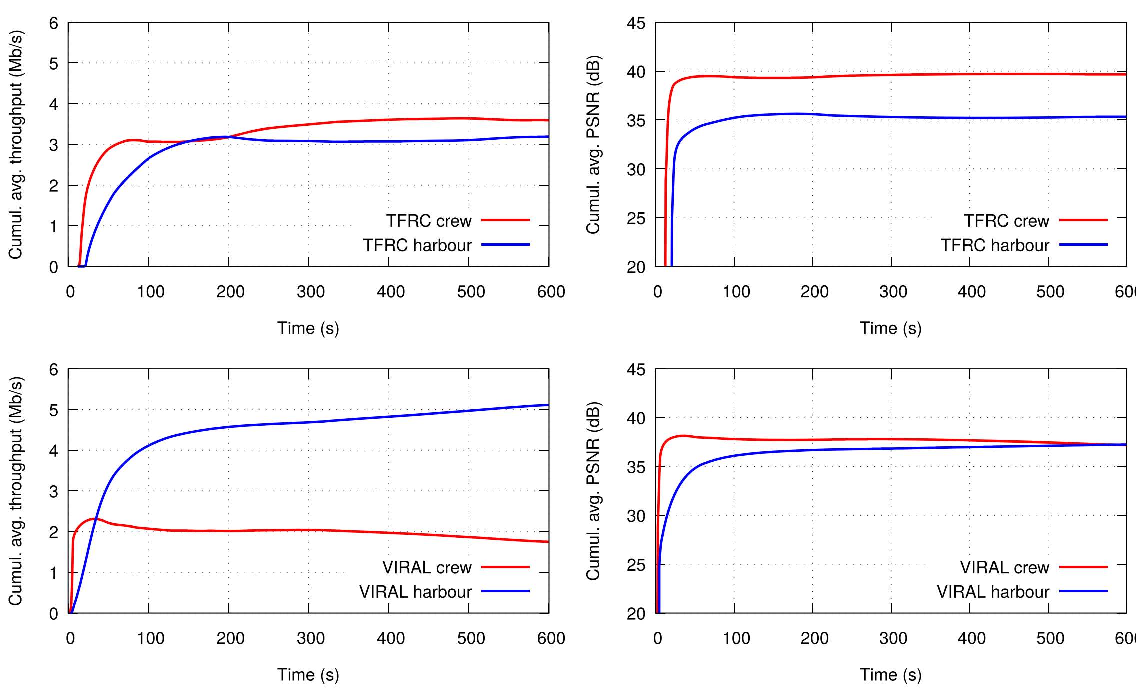
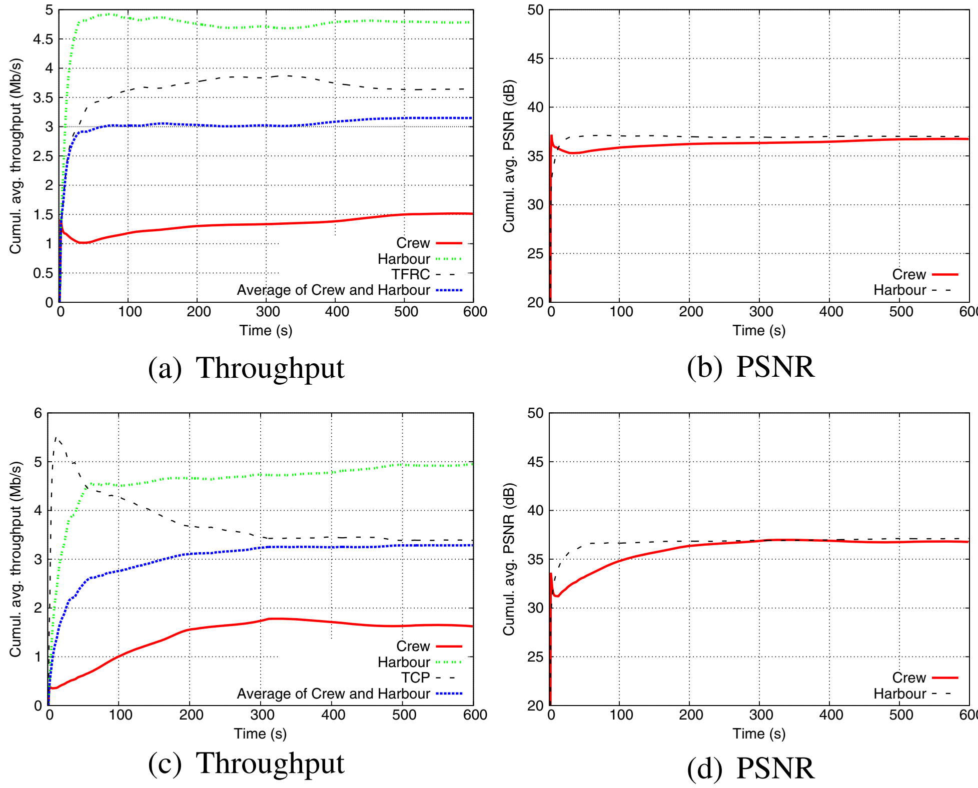


Fig. 5 Throughput (Mb/s) and PSNR (dB) of Crew and Harbour videos using VIRAL or TFRC protocol

Fig. 6 Two video flows versus one TFRC flow (**a**, **b**) and versus one TCP flow (**c**, **d**)



(a) Throughput (b) PSNR (c) Throughput (d) PSNR

intra-fairness between TFRC flows in terms of throughput is also achieved (Fig. 8b). Similar results are obtained in the case of TCP (bottom sub-figures).

3.4 Two video flows not sharing the same bottleneck

In the previous simulations, we investigated VIRAL behaviour when all concurrent flows share the same bottleneck link. In this section, we perform an evaluation where the two video flows do not share the same bottleneck link as illustrated in Fig. 9. The bottleneck capacity for the Crew is set to 2 Mb/s (higher than the achieved throughput of ≈ 1 Mb/s in Fig. 5) while the one for Harbour is set to 4 Mb/s (lower than the achieved throughput of ≈ 6 Mb/s in Fig. 5). From Fig. 10, both video flows achieve their respective bottleneck capacity which corresponds to the respective quality. This simulation shows that VIRAL behaves like TFRC which obtains the link capacity in the absence of concurrent flows. In other words, VIRAL is not impacted by the asymmetry of different bottleneck links.

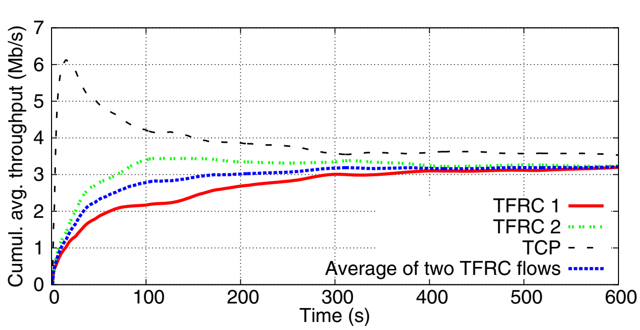


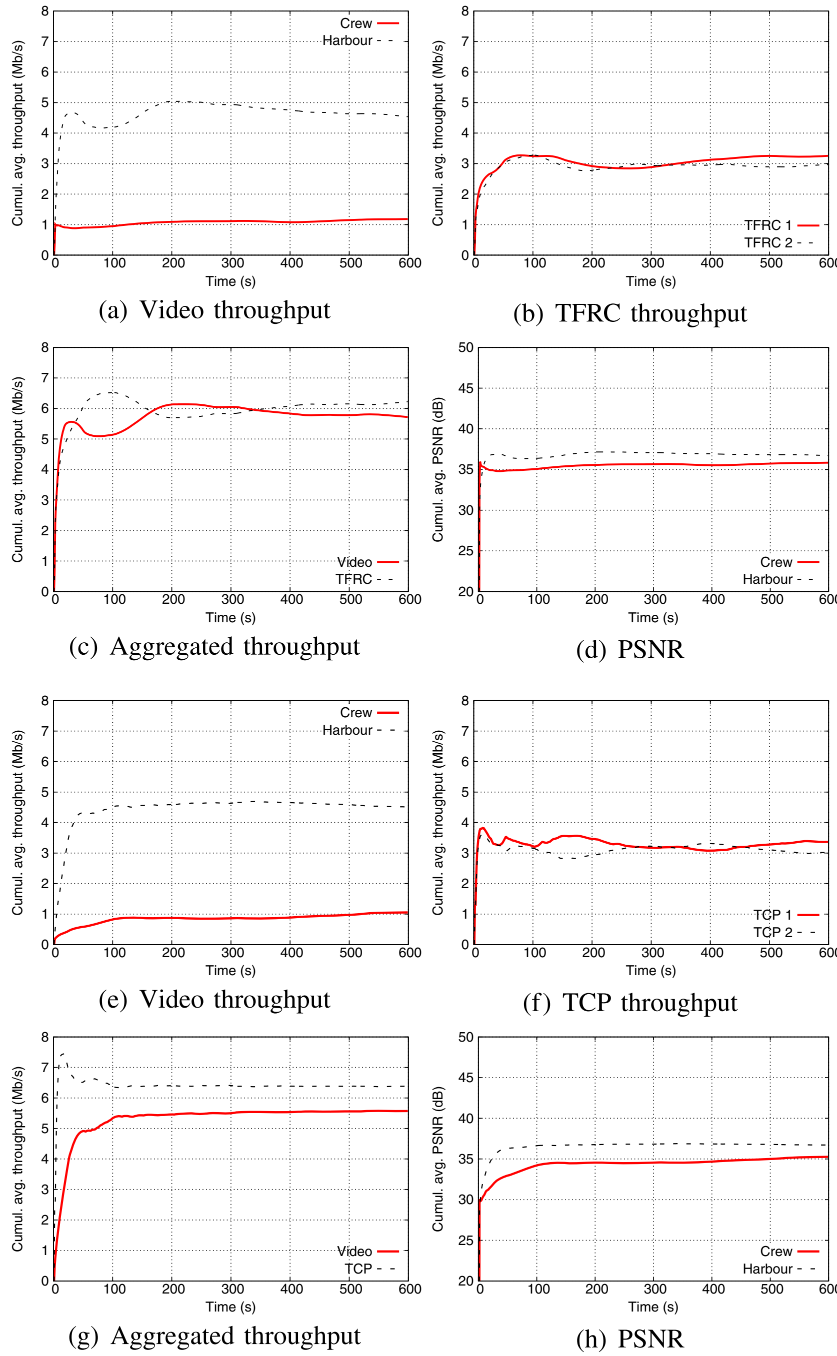
Fig. 7 Two TFRC flows versus one TCP flow

4 State of the art against VIRAL proposal

Before concluding this work, we propose to review existing solutions to better weight up the benefit of this proposal.

Looking at the existing solutions, we first present then discuss in the following the ones proposed in [9, 10, 21, 22]:

1. In [22], an application-transport layer interaction approach for scalable video in the context of unicast congestion control is proposed to maximize the expected delivered video quality at the receiver. A source packetization scheme transforms a scalable video bitstream to provide graceful resilience to network packet drops. The congestion control mechanism targets a low variation in the transmission rate in steady state, and at the same time TCP-friendliness.
2. In [9], the resource allocation problem for multiple media streaming over the Internet is addressed. A proposal, called multimedia streaming TCP-friendly protocol (MSTFP), combines forward estimation of network state with information feedback control to



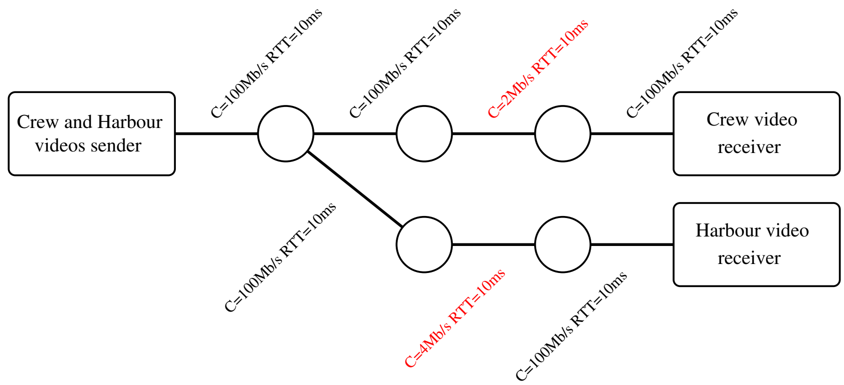
(a) Video throughput (b) TFRC throughput (c) Aggregated throughput (d) PSNR (e) Video throughput (f) TCP throughput (g) Aggregated throughput (h) PSNR

Fig. 8 Two video flows versus two TFRC flows (**a**, **b**, **c**, **d**) and versus two TCP flows (**e**, **f**, **g**, **h**)

track network conditions. Then, MSTFP adapts the media rate to the estimated network bandwidth using each media rate-distortion function under various network conditions. Also in [1], an analytic model to investigate the performance of TCP for both live and stored media streaming is developed. These models provide guidelines for achievable TCP throughput as a function of the video bit rate when direct TCP streaming (i.e. a baseline streaming scheme which uses TCP directly for streaming) leads to satisfactory performance.

3. Scalable streaming video protocol (SSVP), an E2E protocol which operates on top of UDP optimized for unicast video-streaming applications, is proposed in [10]. SSVP employs AIMD-based congestion control and adapts the sending rate by properly adjusting the inter-packet-gap (IPG). The smoothness-oriented modulation of AIMD parameters and IPG adjustments

Fig. 9 Non-common bottleneck topology



reduce the magnitude of AIMD oscillation and allow for smooth transmission patterns, while simultaneously maintaining TCP-friendliness.

4. A recent work on quality-aware congestion control is proposed in [21], where an AIMD-like media-aware congestion control determines the optimal congestion window updating policy for multimedia transmission. The media-aware congestion control problem is formulated as a partially observable Markov decision process (POMDP), which maximizes the long-term expected quality of the received multimedia application. The online learning approach improves the received video quality while maintaining TCP-friendliness of the congestion control in various network scenarios but no video quality fairness is targeted.

Considering these four proposals only, the fairness criteria is always measured in terms of throughput without considering video quality as a factor to condition this throughput fairness. On the contrary and once again, VIRAL considers both intra-fairness between video flows in terms of video quality and inter-fairness in terms of throughput between TCP and video flows.

A resource-aware and quality-fair video content sharing system is presented in [23]. The server uses multiple TCP connections adaptively, depending on the anticipated status of each client playout buffer, to guarantee the bandwidth of each video-streaming session. The proposed algorithm can provide service quality fairness among simultaneous multiple heterogeneous video-streaming services and content download sessions. However, the quality fairness is defined as a quality of service index and not as a video quality metric. Furthermore, the solution is not purely sender based and needs the collaboration of end users by monitoring the playout buffer of the receiving application, making this proposal much more complex than VIRAL.

At the time of writing this paper, RMCAT (RTP Media Congestion Avoidance Techniques) IETF (Internet Engineering Task Force) working group is highly active in proposing congestion control mechanisms for multimedia over RTP (Real-time Transport Protocol).

At last, in a previous work, the authors of this paper proposed Q-AIMD [11] that enables fairness in terms of video quality instead of throughput as TCP does. Q-AIMD is based on AIMD (additive increase multiplicative decrease) principles but targeting video quality instead

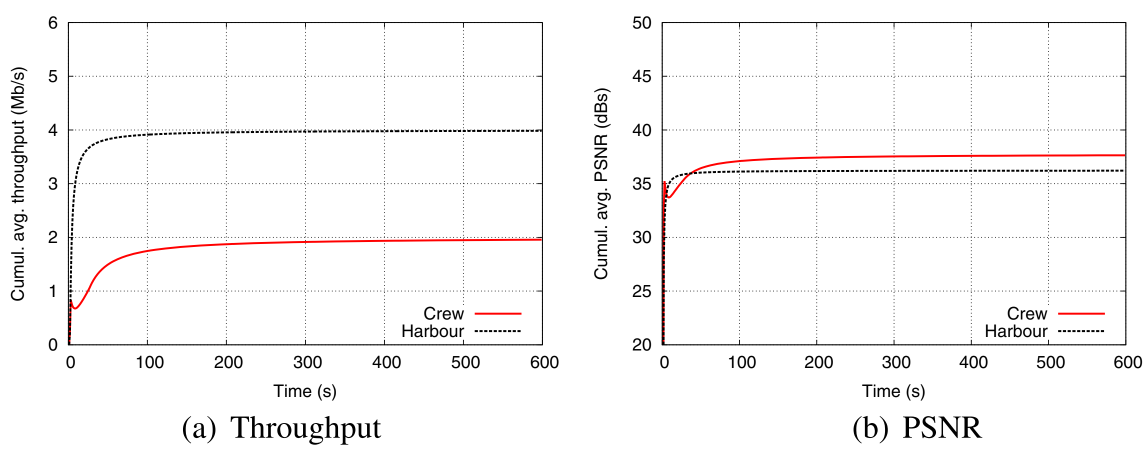


Fig. 10 Throughput (Mb/s) and PSNR (dB) of Crew and Harbou videos where both video flows do not share the same bottleneck link. The link capacity for Crew and Harbour is set to 2 Mb/s and 4 Mb/s respectively

of congestion window as TCP AIMD. In the absence of congestion, there is an additive increase in video quality while the video quality is decreased by a coefficient in case of congestion. Q-AIMD is a decentralized approach which does not require the exact information on how many users are competing over the same bottleneck. Compared with VIRAL, Q-AIMD addresses the same problem but from a centralized perspective. Although this centralized approach can be seen as a step back, there are two main advantages:

– VIRAL is not linked to a given congestion controlled transport protocol. VIRAL is compliant with any transport protocols allowing a fair share with other congestion controlled flows in terms of throughput.
– The proposed approach is compliant with actual VoD operators architecture as explained in Section 2.1.
– Finally, as the congestion control is operated by the sender (i.e. video flows are sent from the VoD operator infrastructure), VIRAL is sender based and does not need a non-scalable end-to-end deployment. Only the sender has to be modified. Note that TFRC can also be used as a sender-based protocol [24].

Today, ICN/CCN paradigm is popular while this study lays on a CSP centralized architecture. Today, the location of the data, whether localized in the physical space or defined by the own topology of the Internet, is not taken into account. The root idea of ICN (also known as CCN) is to consider that the network is mainly used to access content and that we must therefore architect the network around this access to the content. Although this position can be questionable, in our case, ICN/CCN paradigm remains aligned with VIRAL objectives. This approach would, in theory, be more efficient (caching by the network components would improve performance) and scalable (replication of popular data would be simple) and should lead to better resilience (the content would be reachable from several places).

5 Discussion and conclusion

This paper introduced VIRAL, a novel transport mechanism that enables both intra-fairness between video flows in terms of video quality and inter-fairness in terms of throughput between TCP and video flows. Compared with Q-AIMD that provides good intra-fairness between flows in terms of video quality metric, VIRAL also guarantees the inter-fairness with non-video flows (e.g. TCP flows) for all network settings (e.g. available bandwidth, RTT). VIRAL is based on a protocol that already provides TCP-friendliness property. Thus, the inter-fairness with TCP is offered by such protocol. Nevertheless, this approach requires some assumptions (e.g. number of competing flows at bottleneck link, rate-quality curve of each flow). We believe these hypotheses are realistic considering the use of this proposal inside a content delivery network (CDN) domain or inside a content service provider (CSP) providing video-on-demand services. In a future work, an interesting extension of this work would be to drive a large statistical analysis to obtain a generic virtual curve representing a large number of video flows. In addition, another interesting extension could be the case where a content service provider offers different classes of service (e.g. gold, silver) to different clients. For instance, all gold clients have the same video quality but they achieve higher video quality than the silver ones.

References

1. Wang B., Kurose J., Shenoy P., Towsley D. (2008) Multimedia streaming via TCP: an analytic performance study. ACM Trans Multimedia Comput Commun Appl 4(2):16:1–16:22. https://doi.org/10.1145/1352012.1352020
2. Floyd S, Handley M, Padhye J, Widmer J (2008) TCP friendly rate control (TFRC): protocol specification
3. Floyd S, Handley M, Padhye J, Widmer J (2000) Equation-based congestion control for unicast applications. In: ACM SIGCOMM
4. Gu X, Di P, Wolf L (2006) Performance evaluation of DCCP: a focus on smoothness and TCP-friendliness. Annales Des Télécommunications 61(1):46–71. https://doi.org/10.1007/BF03219968
5. Welzl D, Damjanovic M (2009) MulTFRC: providing weighted fairness for multimedia applications (and others too!) SIGCOMM Comput Commun Rev 39(3):5–12
6. Jourjon G, Lochin E, Sénac P (2008) Design, implementation and evaluation of a QoS-aware transport protocol. Comput Commun 3(n. 8):1713–1722
7. Widmer J, Denda R, Mauve M (2001) A survey on TCP-friendly congestion control. IEEE Netw 15(3):28–37. https://doi.org/10.1109/65.923938
8. Murata G, Hasegawa M (2001) Survey on fairness issues in tcp congestion control mechanisms. IEICE Trans Commun E84-B(8):1461–1472
9. Zhang Q, Zhu W, Zhang Y (2001) Resource allocation for multimedia streaming over the internet. IEEE Trans on Multimedia 3(3):339–355
10. Tsaoussidis P, Papadimitriou V (2007) SSVP: A congestion control scheme for real-time video streaming. Comput Netw 51(15):4377–4395
11. Thai T, Changuel N, Kerboeuf S, Faucheux F, Lochin E, Lacan J (2013) Q-AIMD: A congestion aware video quality control mechanism. In: 20th International Packet Video Workshop (PV), pp 1–820th
12. Gill P, Arlitti M, Li Z, Mahanti A (2007) Youtube traffic characterization: a view from the edge. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, ACM, New York, IMC '07, pp 15–28
13. Nafaai A, Murphy L, Murphy S (2008) Analysis of a large-scale VoD architecture for broadband operators: a P2P-based solution. IEEE Commun Mag 46(12):47–55. https://doi.org/10.1109/MCOM.2008.4689207
14. Celis Muñoz E, Le Denmat F, Morin A, Lagrange X (2015) Multimedia content delivery trigger in a mobile network to reduce the peak load. Ann Telecommun - Ann Telecommun 70(7):321–330

15. Liui J, Simon Q, Yang G (2018) Congestion avoidance and load balancing in content placement and request redirection for mobile CDN. IEEE/ACM Trans Networking PP(99):1–13. https://doi.org/10.1109/TNET.2018.2804979
16. Jingyu Y, Qionghai D, Wenli X, Rong D (2005) A rate control algorithm for MPEG-2 to H.264 real-time transcoding. Visual Communications and Image Processing
17. Ma S, Gao W, Lu Y (2005) Rate-distortion analysis for h.264/AVC video coding and its application to rate control. IEEE Trans Circuits Syst Video Technol 15(12):1533–1544
18. Chikkerur S, Sundaram V, Reisslein M, Karam L (2011) Objective video quality assessment methods: a classification, review, and performance comparison. IEEE Trans on Broadcasting 57(2):165–182
19. Jain R (1991) The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling. Wiley professional computing, ISBN 978-0-471-50336-1, I-XXVII, 1–685
20. De Simone et al (2010) A h.264-AVC video database for the evaluation of quality metrics. In: IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, 2010, pp 2430–2433. https://doi.org/10.1109/ICASSP.2010.5496296
21. Habachi O, Shiang H, Van Der Schaar M, Hayel Y (2013) Online learning based congestion control for adaptive multimedia transmission. IEEE Trans Signal Process 61(6):1460–1469
22. Puri R, Lee K, Ramchandran K, Bharghavan V (2001) An integrated source transcoding and congestion control paradigm for video streaming in the internet. IEEE Trans on Multimedia 3(1):18–32
23. Choe Y, Jung Y (2010) Resource-aware and quality-fair video-streaming using multiple adaptive TCP connections. Comput Electr Eng 36(4):702–717
24. Jourjon G, Lochin E, Sénac P (2007) Towards sender-based TFRC. In: 2007 IEEE International Conference on Communications, pp 1588–1593. https://doi.org/10.1109/ICC.2007.266