To cite this document: Tran Thai, Tuan and Changuel, Nesrine and Kerboeuf, Sylvaine and Faucheux, Frederic and Lochin, Emmanuel and Lacan, Jérôme *Q-AIMD: A Congestion Aware Video Quality Control Mechanism.* (In Press: 2013) In: Packet Video Workshop, 12 December 2013 - 13 December 2013 (San Jose, United States).

Q-AIMD: A Congestion Aware Video Quality Control Mechanism

Tuan Tran Thai[1], Nesrine Changuel[2], Sylvaine Kerboeuf[2], Frédéric Faucheux[2], Emmanuel Lochin[1], Jérôme Lacan[1]

[1] University of Toulouse; ISAE/DMIA; TéSA; Toulouse, 31055, France
[2] Alcatel-Lucent - Bell-Labs; Nozay, 91620, France

***Abstract*—Following the constant increase of the multimedia traffic, it seems necessary to allow transport protocols to be aware of the video quality of the transmitted flows rather than the throughput. This paper proposes a novel transport mechanism adapted to video flows. Our proposal, called Q-AIMD for video quality AIMD (Additive Increase Multiplicative Decrease), enables fairness in video quality while transmitting multiple video flows. Targeting video quality fairness allows improving the overall video quality for all transmitted flows, especially when the transmitted videos provide various types of content with different spatial resolutions. In addition, Q-AIMD mitigates the occurrence of network congestion events, and dissolves the congestion whenever it occurs by decreasing the video quality and hence the bitrate. Using different video quality metrics, Q-AIMD is evaluated with different video contents and spatial resolutions. Simulation results show that Q-AIMD allows an improved overall video quality among the multiple transmitted video flows compared to a throughput-based congestion control by decreasing significantly the quality discrepancy between them.**

I. Introduction

The rapid growth of consumer broadband is driving a significant increase in the use of multimedia applications. In addition, the emergence of high speed networks provides the infrastructure and the possibility for handling a wide set of new applications among which the multimedia contents delivery. Recent studies show that the part of multimedia traffic is in constant progression. In particular, authors in [1] estimate that the overall day use of multimedia traffic over a standard European ISP (Internet Service Provider) is around 20% and that the increasing usage of multimedia applications is mainly responsible for the further increase of the network throughput after noon.

The delivery of multimedia traffic, real-time or non-real-time, is usually performed over TCP [2]. This traffic requires an adapted congestion control mechanism to transmit data but also quality aware control mechanism to provide a continuous playout video and high quality at reception. This awareness is not offered by TCP or UDP [3].

When using TCP, a source adjusts a sending window size which corresponds to the maximum amount of packets it can send to the network to prevent congestion at router queues. The resulting variable sending rate of this window-based mechanism is an issue for video applications with strong delay constraint. In fact, despite the fact that TCP AIMD (Additive Increase Multiplicative Decrease) reaches a steady-state, its saw-tooth behavior prevents the application to adapt efficiently its sending rate. Furthermore, the buffering at the sender side might overtake the delay constraint of the application. As a result, TCP is able to support real-time traffic (e.g., live streaming) if the fair-share is at least twice bigger than the source bit rate [4]. For all these reasons, 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) [5] which does not implement retransmissions mechanism. TFRC [6] is a rate-based congestion control mechanism specifically designed to carry multimedia traffic. This protocol is widely adopted as transport mechanism for such traffic due to its smooth sending property. It allows applications that use fixed packet size to compete fairly with TCP flows using the same packet size.

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 [7] which is inadequate when transmitting video communication flows.

Before diving into the description of our proposal, we propose to first look at the existing related work in order to better position our contribution.

A. *Related work*

Many research works have been conducted to better adapt existing transport protocols to multimedia delivery.

In [8], 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 so as to provide graceful resilience to network packet drops. The congestion control mechanism targets low variation in transmission rate in steady state and at the same time TCP-friendliness.

In [9], the resource allocation problem for multiple media streaming over the Internet is addressed. A multimedia streaming TCP-friendly protocol (MSTFP) is proposed, which combines forward estimation of network conditions with information feedback control to track the network conditions and adapt media rate to the estimated network bandwidth using each media R-D function under various network conditions. Also in [10], an analytic model to investigate the performance of TCP for both live and stored media streaming is developed. These models help providing guidelines for achievable TCP throughput as function of the video bitrate as to when direct TCP streaming (i.e., a baseline streaming scheme which uses TCP directly for streaming) leads to satisfactory performance.

An end-to-end protocol, namely Scalable Streaming Video Protocol (SSVP), which operates on top of UDP optimized for unicast video streaming applications is proposed in [11]. 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 reduce the magnitude of AIMD oscillation and allow for smooth transmission patterns, while TCP-friendliness is maintained.

In all previous cases, fairness is always addressed in throughput and video quality is not explicitly considered.

A resource-aware and quality-fair video content sharing system is presented in [12]. 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 services. However, here the quality fairness is defined as quality of service and not in video quality metric.

More recently, a quality-centric congestion control for multimedia streaming over IP networks has been proposed in [13]. The proposed solution adapts the sending rate to both the network condition and the application characteristics by explicitly considering the distortion impacts and delay deadlines. The proposed media-TCP aims to achieve quality-based fairness among multimedia users.

The latest work on quality aware congestion control is proposed in [14], 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, which maximizes the long-term expected quality of the received multimedia application. The on line 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.

B. Main contributions

As in [13], in this paper, we target video quality fairness between multiple video flows. As the trend is to deliver more and more multimedia services over web platforms, the considered system can be mapped to real time web communication system as such targeted by the IETF webRTC working group [15]. To the best of our knowledge, no mechanisms including the following requirements have been already proposed for video flows delivery:

- Video quality fairness between multiple video sessions;
- A low complexity congestion aware algorithm.

The proposed Q-AIMD allows to fulfill the above desired requirements of real-time multimedia flows. Our contributions in this paper are *i)* propose a novel quality driven AIMD congestion aware mechanism, called Q-AIMD, to enable the fairness of video quality instead of throughput, *ii)* discuss the control granularity and the system to deploy this algorithm, and *iii)* evaluate the Q-AIMD using different quality metrics and by taking into account different contents and different spatial resolutions

The paper is organized as follows. In Section II, we present the Q-AIMD algorithm and its application in an end-to-end system and discuss the control granularity. Section III discusses the possible video quality metrics that can be applied to Q-AIMD algorithm and the Q-AIMD convergence. Simulation results and analysis are the topic of Section IV. We conclude the paper and propose the future work in Section V.

II. System description and Q-AIMD algorithm

In this section, we first introduces the application system of the Q-AIMD algorithm, then, we detail the algorithm operation. Lastly, we discuss the control granularity and how fast the system should adapt to the feedback from the network.

A. System description

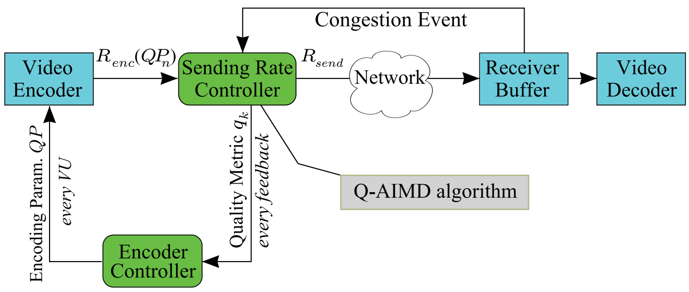


Fig. 1. System Description

Consider a communication system in which N encoded video streams sent by remote servers share a network bottleneck. The video flows are transported over a wired network, with possibly wireless connection on last mile, where the videos are served to the clients from a wireless access point or a mobile base station. The server-client system is depicted on Figure 1. Encoded Video Units (VUs), representing a single frame or a Group of Pictures (GoP), are provided by the encoder at the server side. All frames are assumed to be of the same duration T_{frame} and the frame rate $F = \frac{1}{T_{frame}}$ is assumed constant over time. The encoding parameters (quantization steps, frame rate, etc.) are controlled by the encoder controller. The video encoder can run several modes for the rate control: Constant Bit Rate (CBR), Variable Bit Rate (VBR) or Constant Quality mode. In the following, we choose to operate the encoder in the Constant Quality mode as this mode targets a constant quality bit rate control. An example of Constant Quality mode is the CRF (Constant Rate Factor) [16] mode in x.264 encoder which takes the value of quantization parameter (QP) as an input control parameter and encodes frames or macroblocks at the target QP while allowing some small bounded variations of QP around the target value to take into account the complexity of the frame or macroblock.

At time index j, the encoder delivers the j-th VU. VUs are packetized into various number of constant size packets. The video data are transported over UDP protocol. We assume that the server receives feedback information from the receiver such as RTT (Round Trip Time) and packet loss. At the server side, the sending rate controller can adjust its sending rate R_{send} based on the provided feedback information to prevent congestion in the network. We assume that the video encoder can adjust its output rate for time index n in order to adapt it to the sending rate by an internal rate control mechanism controlled by the encoding parameter denoted QP. The sending rate controller is running the Q-AIMD algorithm for congestion control. For each time index k it provides the quality value q_k of the Q-AIMD algorithm to the encoder controller. The Q-AIMD algorithm will be explained in detail in Section II-B.

It must be emphasized that the time index n at which the encoder can change its encoding parameter can be different from VU index j and the time control index k. The encoder controller calculates the target encoding parameter QP_n associated to the set of q_k over the time index n. Thus the encoded VUs over the time index $n + 1$ are delivered at bit rate $R_{enc}(QP_n)$ to the sending rate controller. The difference between the sending rate and the encoder output rate is absorbed by a rate shaping buffer located inside the sending rate controller.

B. Q-AIMD algorithm

All TCP variants (e.g. TCP NewReno, TCP Westwood) are based on an AIMD or AIMD-like (e.g. CUBIC) principle during the congestion avoidance phase. This principle increases the TCP congestion window every RTT if there is no congestion signal and decreases this congestion window when congestion occurs. In [17], authors show that the AIMD principle converges to fairness among all competing flows crossing the same bottleneck with the same RTT. In this paper, we use this result to propose a video quality AIMD algorithm (Q-AIMD) to achieve fairness in terms of video quality (*e.g.,* PSNR) among competing flows instead of throughput as with TCP. Q-AIMD algorithm in congestion avoidance phase is depicted in the Algorithm 1.

Algorithm 1 General Q-AIMD algorithm

Upon reception of feedback from the receiver
if $CongestionEvent$ **then**
    if $q < q_{worst}$ **then**
        $q = q_{worst}$
    else
        $q = q_{worst} + (q - q_{worst}) * \beta_q$
    end if
else
    $q = q + \alpha_q$
    if $q > q_{best}$ **then**
        $q = q_{best}$
    end if
end if

Upon reception of the feedback (*i.e.* once per RTT), if there is no congestion, the video sender increases linearly its video quality by an Additive-Increase value α_q. This value uses the same unit as the video quality metric (*i.e.* dB for PSNR). An increase of the quality results in an increase of the encoding video bit-rate. We set a threshold denoted q_{best} that limits the maximum video quality. The reason is that it is not necessary for a given video (*i.e.* 'Foreman' in CIF format) to achieve the full available capacity (*i.e.* 10 Mb/s) where the average PSNR is greater than 50 dB as in this case picture defects will be undetected by most users. Indeed, the objective of Q-AIMD is not to reach the full capacity of the link if the video has already achieved its best predefined video quality q_{best}. If a congestion event occurs, the sender decreases the video quality by a Multiplicative-Decrease factor β_q ($0 < \beta_q < 1$). Our goal is to achieve the best predefined video quality while reacting to congestion events. Thus, we might decrease to a lower quality, which means decrease to a lower bit-rate to prevent congestion collapse as TCP AIMD does. We also set a minimum threshold q_{worst} where the video quality should not be lower than this value. Taking the same running example, it is not acceptable in terms of visual perception for an encoded CIF video to get less than 25 dB. However, in a critical condition where the available bandwidth is not enough to ensure the minimum quality q_{worst}, the video sender might reduce the video frame rate or drop less important packets (*i.e.*, packets of B frames) or adjust the q_{worst}. Note that Q-AIMD can be applied to any video quality metrics. We propose to discuss some of them in Section III-A. The discussion about the convergence of Q-AIMD is given in the next Section III-B.

C. Control granularity

An efficient congestion control algorithm should react fast enough to packet loss detection. The reaction time needed for the congestion control should be less than a few RTTs. Thus, we set the time index k as the reception time of the k-th received feedback. We assume that the time interval between two consecutive feedbacks is smaller than the GoP duration. We assume that the index n for the encoder parameter setting is equal to the index j of VU, meaning that the encoder can change its QP encoding parameter every VU. At the k-th received feedback, the sending rate controller sends the value q_k to the encoder controller. Let $\left\{q_k^j\right\}$ denotes the set of q values received by the encoder controller on the j-th time interval. We decide that the encoder controller selects q_{kmax}^j, the latest value of the set $\left\{q_k^j\right\}$ and calculates its corresponding QP value, i.e. QP_{kmax}^j to be used in the encoder parameter setting at the end of time interval j. This choice is motivated by the fact that the latest value of q is the more representative of the congestion state at that time. We also assume that the rate of reception of feedback packets is high enough so that any feedback associated to a congestion event is received within a delay close to the RTT. Thus, after a congestion event detected at index k, the sender can adjust its rate whit a maximum delay equal to $T_k^{reac} = T_{VU} + T_k^{buffer}$ in seconds, where T_{VU} is the VU time duration and T_k^{buffer} the delay induced by the sender rate buffer at instant k.

The size of the sender rate buffer can be kept small within a small number of packets. We then consider two situations for the time scale to react to the congestion, depending of the possible granularity of the encoder rate control, at frame level or at GoP level.

1) Encoding control at the frame level: Here, we assume that the control is done at the frame level, which means that a VU is a video frame. It shortens the delay to adjust the sending rate. The fast adaptation has a counterpart on the video encoder which should accept an update of its encoding parameter setting every frame. Moreover, it will impact the performance of the R-D control of the encoder. A video is usually segmented into several GoPs, each starting with an I frame followed by P and B frames. In general, the rate control of the encoder is realized over the duration of the sequence (for VOD sequence) or over the GoP with possibly one pass or two pass methods. When the rate control of the encoder works at the sequence level or at the GoP level, the R-D control exploits the heterogeneity in complexity of the various succeeding frames. Doing a video rate control at the frame level prevents the encoder from optimizing the output rate in such a way. It will result in a higher video rate for a given perceived video quality than when the control takes place at GoP or sequence level. In practice, an I frame is usually encoded with a lower QP than an inter-coded frame since the I frame is used as the first reference for the coding of the following inter-coding frame(s). For an optimal behavior, the proposed system must distinguish the type of frame and select a distinct QP accordingly.

2) Encoding control at the GoP level: We consider in this case that the rate control at the video encoder side is done at the granularity of the GoP, which means that a VU is a GoP. The adaptation is slower than in the first case and the delay of encoder parameter adaptation is higher.

The trade-off is thus to minimize the GoP duration to keep the congestion reaction fast enough within a few RTTs while maximizing it to improve the encoding efficiency using the rate provided by the encoder rate control algorithm. In the following, we have considered that the GoP duration is about twice the RTT.

III. Discussion on video quality metrics and Q-AIMD convergence

A. Video quality metric

R-D characteristics for video sequences are time-varying and depend on the content of the videos. Provisioning some constant transmission rate to mobile users for video delivery is in general inappropriate. If videos are encoded at a constant bitrate, the quality may fluctuate with the variations of the characteristics of the content. If a constant quality is targeted, bitrate may vary significantly. When using a video codec like H.264, video rate can be adjusted by varying *i)* the image spatial resolution of the video, *ii)* the quantization parameter (QP), or *iii)* the frame rate (fps).

R-D characteristics for video sequences can be easily modeled using different models depending on the considered quality metric, *e.g.*, Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity (SSIM) [18], etc. In this study, we focus on PSNR as one possible video metric to use in the Q-AIMD. Thus, the q value of the pseudo-code for the Q-AIMD algorithm is expressed as a PSNR value. The video encoder should then encode the video with the targeted PSNR value. Nevertheless, it may not be easy to set the control parameters of the encoder to output video with the target PSNR value, as encoders do not take PSNR as a parameter setting.

It turns out that another possible quality metric for Q-AIMD is to consider directly the QP. Indeed, for given resolution and frame rate, the value of QP is representative of the quality of the frames. The q value of the pseudo-code for the Q-AIMD algorithm is thus expressed as a QP value. The QP value can be directly set to the encoder to encode the video with the targeted QP.

Recently user surveys have been conducted investigating the impact of various influence factors on the subjective quality of digital video, especially in the context of mobile environments, see [19], [20], [21]. Subjective assessment in [20], revealed that PSNR and other metrics without consideration of spatial resolutions were not suitable to estimate the quality of videos.

In [20], a new video quality metric (VQM) is derived and modeled as following

$$VQM = \alpha PSNR + \beta M_A \left(30 - FR\right) + \frac{\delta}{\gamma + e^{-\omega x}} + \xi \quad (1)$$

where $\alpha, \beta, \delta, \omega, \xi$ are model parameters, and FR, M_A and x denote the frame rate, motion activity and the height of the spatial resolution, respectively. Model parameters are obtained using non-linear regression using DMOS+100 where DMOS is the differential mean opinion score. Experimental results in [20] showed that the proposed quality measurement modeling gives high correlation on human perception.

In order to consider the impact of the temporal and spatial resolution in our quality based congestion control algorithm, we use the model in (1) as possible metric for video quality fairness.

In the simulation section, we will compare and discuss the performance of Q-AIMD for the various quality metrics. In the rest of the paper, we noted PSNR-AIMD, QP-AIMD and VQM-AIMD the different variants of the Q-AIMD algorithm.

B. Discussion on Q-AIMD convergence

The AIMD principle implemented inside TCP is known to converge in congestion window [22]. When all flows crossing a bottleneck have the same RTT, the fairness in congestion window means the fairness in bit rate. In the absence of congestion, the competing flows crossing the same bottleneck increase their congestion window based on AIMD principle that means an increase in bit rate. Congestion occurs when the sum of bit rates exceeds the available capacity. In case of Q-AIMD, the algorithm adapts the quality value while the congestion is still caused by the sum of bit rates that exceeds the available capacity. Taking as an example case where Q-AIMD is driven by QP, from [23], [24] the relationship between rate R and QP can be modeled as follows:

$$R = a\, e^{b\, QP} \tag{2}$$

where a and b represent the characteristics of the video. Fig. 2(a) and Fig. 2(b) show the resulting state trajectory of both algorithms of two flows carrying the same video. As shown on both figures, the convergence in rate results in the convergence in QP. We reserve in a future work to complete this geometric resolution with an analytical one.

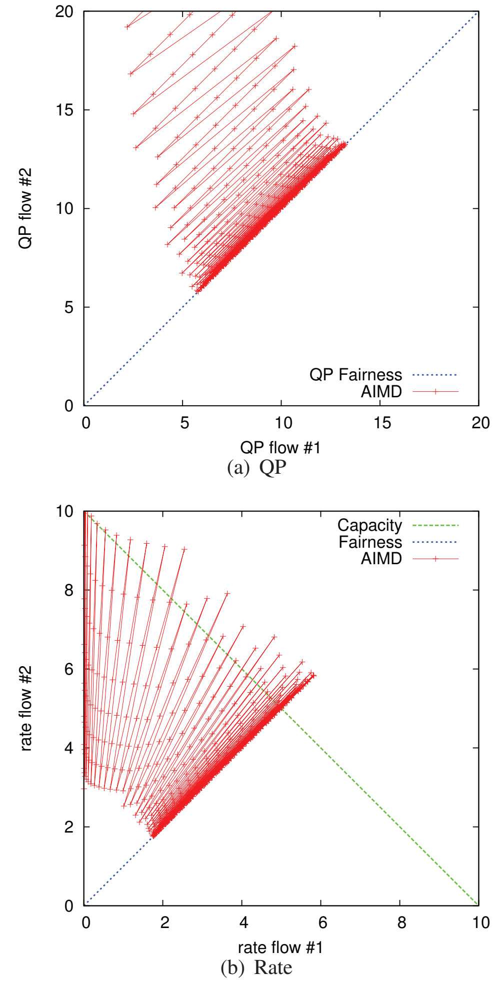


Fig. 2. State trajectory diagrams in QP and in rate for two flows

IV. Simulation results

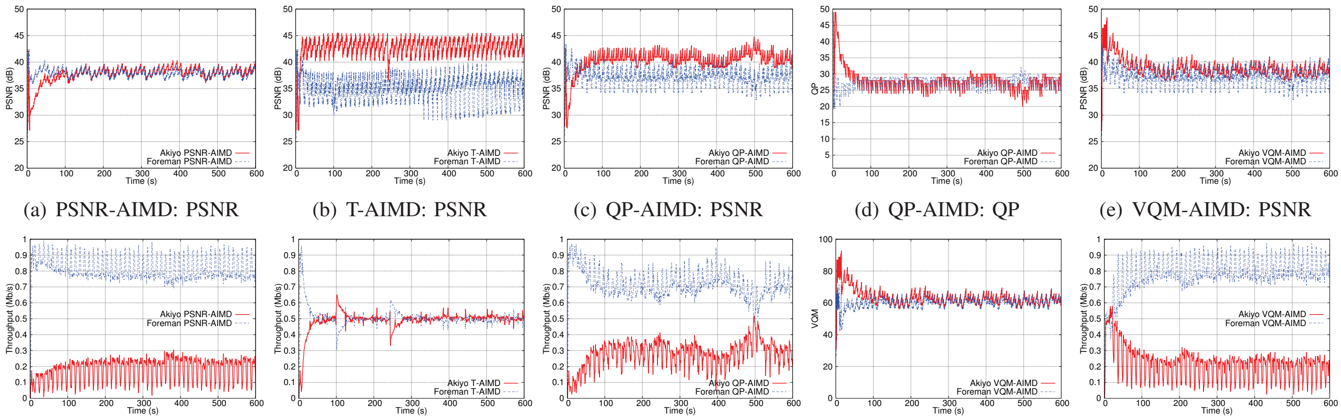
We evaluate Q-AIMD algorithm using $ns2$ simulator [25] with three variants, depending on the quality metric used for Q : PSNR-AIMD, QP-AIMD and VQM-AIMD targeting fair PSNR, QP and VQM, respectively. We compare the three variants with a reference case where the fairness target is the throughput. We call this variant T-AIMD for throughput-based AIMD. T-AIMD is similar to an unreliable AIMD congestion control as the TCP-like version of DCCP denoted DCCP/CCID#2 [26]. The rationale of using T-AIMD is to fairly compare our solution with similar assumptions. A comparison with TFRC might appear as a better choice. However, TFRC protocol does not converge as fast as TCP-like due to its smooth property [27]. As a result, the simulation obtained would be in favor of Q-AIMD and difficult to analyze.

TABLE I
Parameters of the Q-AIMD algorithm

Parameters	PSNR-AIMD	QP-AIMD	VQM-AIMD
(q_{worst},q_{best})	(30,50) dB	(50,1)	(30,100)
(α_q,β_q)	(0.15,0.85)	(-1.0,0.85)	(1.0,0.85)

The values of the parameters (α_q,β_q) in Algorithm 1 used in the experimental tests are in Table I. these values are obtained experimentally and correspond to good trade off between a fast convergence behavior and less oscillation. In fact, (α_q) should be positive when the quality metric is increasing to improve the video quality (PSNR, VQM) and negative when the quality metric is decreasing to improve the video quality (distortion, QP). Optimal values of (α_q,β_q) can be obtained using optimization system by maximizing a utility function minimizing oscillation and maximizing the convergence speed. This optimization will be addressed in future works.

The base RTT is set to 100 ms. Parameters of the Q-AIMD algorithm are given in Table I. For the QP-AIMD, QP values are rounded to the closest integer at each decreasing quality event. The video data is encoded using x.264 encoder [28] where the video frame rate is set to 25 Hz with a GoP size of 5 frames which results in the GoP duration of 200 ms. The simulation lasts 600 seconds and corresponds to 10 minutes of video transmission and all video flows start at the same time. As discussed in Section II, the video encoder adapts its



(a) PSNR-AIMD: PSNR (b) T-AIMD: PSNR (c) QP-AIMD: PSNR (d) QP-AIMD: QP (e) VQM-AIMD: PSNR

(f) PSNR-AIMD: Throughput (g) T-AIMD: Throughput (h) QP-AIMD: Throughput (i) VQM-AIMD: VQM (j) VQM-AIMD: Throughput

Fig. 3. Two videos with same spatial resolution

quality as well as bit rate at the GoP level in the simulations. In these simulations, we consider constant frame rate and only spatial resolution between different video streams is varying. The performance of the three variants of Q-AIMD is evaluated in terms of video quality and video fairness. When video flows have the same spatial resolution and the same frame rate, the VQM in (1) is proportional to the PSNR. Thus, we will evaluate visual quality in terms of PSNR when video flows have the same spatial resolution and in VQM metric when video flows have distinct spatial resolution and frame rate.

A. Two videos with same spatial resolution

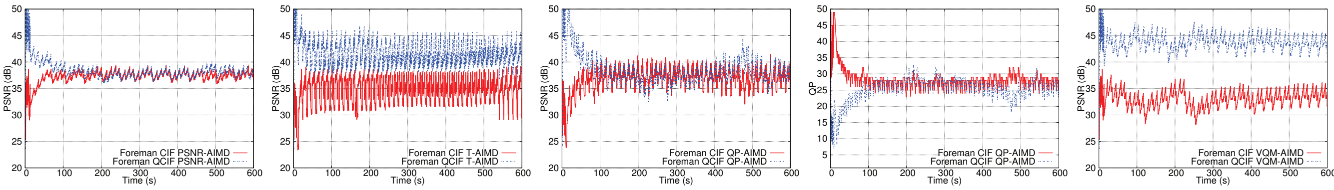
In this simulation, two CIF video sequences 'Akiyo' and 'Foreman' share the same bottleneck link of 1 Mb/s. We evaluate the visual quality of the videos in terms of PSNR value. Fig. 3(a) and 3(f) show that PSNR-AIMD achieves the fairness in PSNR between both video flows but not in throughput. Indeed, 'Akiyo' ('Foreman', respectively) achieves an average PSNR of 37.75 dB (37.93) and an average throughput of 0.18 Mb/s (0.82). On the other hand, achieving the fairness in throughput using T-AIMD is obtained at the cost of a significant gap in quality of nearly 8 dB between the video flows (Fig. 3(b), and 3(g)). In fact, 'Akiyo' and 'Foreman' with T-AIMD achieve an average PSNR of 42.76 and 34.95 dB, respectively. Similarly, QP-AIMD achieves the fairness in QP value (Fig. 3(d)) but with a slight difference in PSNR value (Fig. 3(c)). This can be explained by the fact that with the same QP, the video with less complexity and/or low motion (i.e., 'Akiyo') tends to achieve better PSNR than the one with high complexity and/or high motion (i.e., 'Foreman'). Both video flows converge in VQM value with VQM-AIMD. When video flows have the same spatial resolution and the same frame rate, the sum of three last components in Eq. (1) is the same for both flows and VQM-AIMD achieves the fairness in PSNR value.

B. Two videos with different spatial resolutions

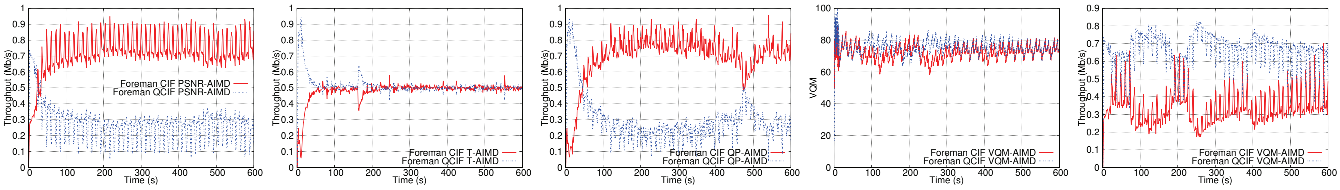
In this simulation, two videos 'Foreman' with different spatial resolutions CIF and QCIF share the same bottleneck capacity of 1 Mb/s. We evaluate the visual quality of video flows in terms of VQM value. Fig. 4(e), and 4(i) show that, in order to achieve the same visual quality VQM, smaller resolution video ('Foreman' QCIF) must have higher PSNR than the higher resolution video ('Foreman' CIF). In fact, VQM Model in Eq. (1) allows approaching the visual quality at the receiver. This model assumes that all terminals have the same spatial resolution corresponding to the highest one. Thus, for video transmitted with low resolution (i.e., QCIF) they should be up sampled at the decoder to be displayed at high resolution. This process may introduce the degradation in the quality of the decoded video contrarily to video transmitted with high resolution and so do not require up sampling process. This explain the need for higher PSNR for low resolution video compared to high resolution ones. Since PSNR-AIMD (Fig. 4(a), and 4(g)) and QP-AIMD (Fig. 4(c), 4(d), and 4(h)) do not take into account the spatial resolution, Table II shows a significant gap in VQM between the two videos. T-AIMD (Fig. 4(b), and 4(g)) achieves better VQM fairness than both PSNR-AIMD and QP-AIMD. Indeed, sharing the same bandwidth with 'Foreman' in CIF, 'Foreman' in QCIF achieves better PSNR value since it has smaller spatial resolution. This results in a closer gap in VQM than PSNR-AIMD and QP-AIMD. VQM-AIMD achieves the fairness in visual quality and reduces the quality discrepancy with respect to T-AIMD.

TABLE II
MEAN AND STANDARD DEVIATION VQM OF 2 VIDEOS

	Foreman CIF	Foreman QCIF
PSNR-AIMD	84.62±3.34	59.64±4.77
T-AIMD	77.08±7.07	68.91±6.44
QP-AIMD	82.98±6.23	59.55±8.40
VQM-AIMD	72.24±4.50	75.27±4.38



(a) PSNR-AIMD: PSNR (b) T-AIMD: PSNR (c) QP-AIMD: PSNR (d) QP-AIMD: QP (e) VQM-AIMD: PSNR



(f) PSNR-AIMD: Throughput (g) T-AIMD: Throughput (h) QP-AIMD: Throughput (i) VQM-AIMD: VQM (j) VQM-AIMD: Throughput

Fig. 4. Two videos with different spatial resolution

C. *Six videos with same spatial resolution*

The aim of this simulation is to show how the different algorithms perform when more than two video flows compete at the same bottleneck link of 3 Mb/s. Table III shows that PSNR-AIMD and VQM-AIMD achieve good quality fairness in PSNR while there is more discrepancy in visual quality between the videos with QP-AIMD. The maximum discrepancy between videos is 1.47 dB for PSNR-AIMD between 'Coastguard' and 'Mother&Daughter' and 3.16 dB for VQM-AIMD between 'Akiyo' and 'Coastguard' while QP-AIMD has a maximum discrepancy of 6.71 dB between 'Akiyo' and 'Coastguard'. Indeed, QP-AIMD algorithm aims to achieve the fairness in QP, not in PSNR. With T-AIMD, 'Coastguard', 'Foreman' and 'Silent' videos suffer from bad quality, they obtain a PSNR of 31.06, 34.74 and 34.63 dB, respectively. Furthermore, the discrepancy in PSNR is more than 10 dB between 'Akiyo', 'Mother&Daughter' and 'Coastguard'. In fact, since they are high motion and/or complex videos, they require more bandwidth to obtain the same PSNR as 'Akiyo', 'Hall' and 'Mother&Daughter' videos.

D. *Six videos with different spatial resolutions*

In this simulation, three CIF videos ('Akiyo', 'Coastguard', 'Foreman') compete with the same three videos in QCIF format at the bottleneck link of 3 Mb/s. Table IV shows that the QCIF videos suffer from a lower VQM value than the CIF videos for PSNR-AIMD and QP-AIMD since these two Q-AIMD variants achieve good fairness in their metrics. However, the VQM reflects better visual perception than the PSNR for different spatial resolutions. VQM-AIMD achieves the fairness in visual quality and is the most appropriate algorithm with respect to the two other variants. The discrepancy in the visual quality of the different flows is high in the T-AIMD in terms of both PSNR and VQM.

V. Conclusion and Future Work

A new congestion aware mechanism targeting fairness in video quality while transmitting multiple video flow is proposed in this paper. The proposed solution is called Q-AIMD, since it uses quality metric as fairness convergence criteria applied on classical AIMD algorithm. The proposed solution is evaluated with heterogeneous video contents and with different spatial resolutions. Q-AIMD is compared with classical throughput based AIMD in terms of video quality. The simulation results present an important decrease in the video quality discrepancies between the different transmitted video flows. The video quality is evaluated using different metrics to better consider the quality requirements for different spatial resolutions. In future work, we plan to analyze the convergence of the algorithm and to study the fairness against different TCP variants. Furthermore, the video quality evaluation in this paper is performed at the sender side, we expect to use the erasure codes to protect from packet losses and evaluate the video quality at the receiver side.

VI. Acknowledgments

This work was supported by the French ANR grant ANR-VERS-019-02 (ARSSO project). The authors are grateful to Abir Amri for her help in simulation tests.

References

[1] S. Gebert, R. Pries, D. Schlosser, and K. Heck, "Internet access traffic measurement and analysis," in *Proceedings of the 4th international conference on Traffic Monitoring and Analysis*, ser. TMA'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 29–42.

[2] J. B. Postel, "Transmission control protocol," RFC 793,, Tech. Rep., 1981.

[3] ——, "User datagram protocol," RFC 768, Tech. Rep., 1980.

[4] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia streaming via TCP: An analytic performance study," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 2, pp. 16:1–16:22, May 2008.

[5] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP friendly rate control TFRC: Protocol specification," RFC 3448, Tech. Rep., 2003.

[6] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," Internet Engineering Task Force, Sep. 2008.

[7] G. Hasegawa and M. Murata, "Survey on Fairness Issues in TCP Congestion Control Mechanisms," *IEICE Transactions on Communications*, vol. E84-B, no. 8, pp. 1461 – 1472, 2001.

TABLE III
MEAN AND STANDARD DEVIATION OF THE PSNR OF SIX CIF VIDEOS

	Akiyo	Coastguard	Foreman	Hall	Mother&Daughter	Silent
PSNR-AIMD	36.44±1.69	35.01±0.91	35.74±0.87	35.60±0.92	36.48±1.53	35.62±0.82
T-AIMD	42.73±1.63	31.06±1.43	34.74±2.60	36.92±1.41	41.68±1.50	34.63±1.56
QP-AIMD	40.41±1.33	33.70±1.13	35.83±1.59	37.36±0.89	39.67±1.05	35.39±1.03
VQM-AIMD	37.83±2.06	34.67±1.31	35.14±1.23	35.62±1.49	37.34±1.90	35.25±1.36

TABLE IV
MEAN AND STANDARD DEVIATION OF THE PSNR AND THE VQM OF SIX VIDEOS WITH DIFFERENT SPATIAL RESOLUTIONS CIF AND QCIF

	Akiyo CIF	Coastguard CIF	Foreman CIF	Akiyo QCIF	Coastguard QCIF	Foreman QCIF
PSNR (dB)						
PSNR-AIMD	36.95±1.97	36.07±0.99	36.61±0.92	37.54±2.48	36.55±0.98	36.69±1.92
T-AIMD	42.64±1.72	30.99±1.31	34.74±2.62	49.20±2.07	37.91±2.11	41.21±2.48
QP-AIMD	41.17±1.12	35.21±1.17	36.99±1.56	41.21±1.81	36.25±1.45	38.07±1.86
VQM-AIMD	35.63±1.97	32.76±1.21	33.82±1.72	43.96±2.12	42.13±1.69	42.65±1.62
VQM						
PSNR-AIMD	83.69±5.73	81.13±2.87	82.71±2.67	57.29±7.19	54.43±2.84	54.85±5.56
T-AIMD	100.18±5.00	66.40±3.79	77.26±7.59	91.11±6.01	58.37±6.13	67.95±7.20
QP-AIMD	95.93±3.25	78.63±3.40	83.80±4.53	67.93±5.24	53.55±4.20	58.82±5.40
VQM-AIMD	79.85±5.71	71.52±3.51	74.62±4.98	75.90±6.16	70.61±4.89	72.12±4.69

[8] R. Puri, K.-W. Lee, K. Ramchandran, and V. Bharghavan, "An integrated source transcoding and congestion control paradigm for video streaming in the internet," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 18 – 32, 2001.

[9] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource allocation for multimedia streaming over the internet," *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 339 – 355, 2001.

[10] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia streaming via TCP: An analytic performance study," in *ACM Multimedia*, 2004.

[11] P. Papadimitriou and V. Tsaoussidis, "SSVP : A congestion control scheme for real-time video streaming," *Computer networks*, vol. 51, no. 15, pp. 4377 – 4395, 2007.

[12] Y. Jung and Y. Choe, "Resource-aware and quality-fair video-streaming using multiple adaptive TCP connections," *Computers and Electrical Engineering*, vol. 36, no. 4, pp. 702 – 717, 2010.

[13] H.-P. Shiang and M. van der Schaar, "A quality-centric TCP-friendly congestion control for multimedia transmission," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 896 – 909, 2012.

[14] O. Habachi, H.-P. Shiang, M. van der Schaar, and Y. Hayel, "Online learning based congestion control for adaptive multimedia transmission," *IEEE Transactions on Signal Processing*, vol. 61, no. 6, pp. 1460 – 1469, 2013.

[15] H. Alvestrand, "Overview: Real Time Protocols for Brower-based Applications," IETF, Tech. Rep., 2013.

[16] L. Merritt and R. Vanam, "Improved Rate Control and Motion Estimation for H.264 Encoder," in *IEEE International Conference on Image Processing*, vol. 5, 2007, pp. 309 – 312.

[17] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comput. Netw. ISDN Syst.*, vol. 17, no. 1, pp. 1–14, Jun. 1989.

[18] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, pp. 600 – 612, 2004.

[19] J. Mccarthy, M. Sasse, and D. Miras, "Sharp or smooth? comparing the effects of quantization vs. frame rate for streamed video," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, 2004, pp. 535 – 542.

[20] C.-S. Kim, S.-H. Jin, D.-J. Seo, and Y.-M. Ro, "Measuring Video Quality on Full Scalability of H.264/AVC Scalable Video Coding," in *IEICE Transactions on Communications*, vol. E91-B, no. 5, 2008, pp. 1269 – 1278.

[21] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld, "Impact of frame rate and resolution on objective qoe metrics," in *International Workshop on Quality of Multimedia Experience*, 2010, pp. 29 – 34.

[22] V. Jacobson, "Congestion avoidance and control," Stanford, CA, Aug. 1988, pp. 314–329.

[23] S. Ma, W. Gao, and Y. Lu, "Rate-distortion analysis for H.264/AVC video coding and its application to rate control," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1533–1544, 2005.

[24] Y. Jingyu, D. Qionghai, X. Wenli, and D. Rong, "A rate control algorithm for MPEG-2 to H.264 real-time transcoding," *Visual Communications and Image Processing*, 2005.

[25] "The Network Simulator http://www.isi.edu/nsnam/ns/index.html."

[26] S. Floyd and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control," RFC 4341 (Proposed Standard), Mar. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4341.txt

[27] S. Floyd, E. Kohler, and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)," RFC 4342 (Proposed Standard), Mar. 2006, updated by RFC 5348. [Online]. Available: http://www.ietf.org/rfc/rfc4342.txt

[28] "x264 http://www.videolan.org/developers/x264.html."