Modeling and Design of Adaptive Video Streaming Control Systems

Giuseppe Cofano, Luca De Cicco *Member, IEEE*, and Saverio Mascolo, *Senior Member, IEEE*

***Abstract*—Adaptive video streaming systems aim at providing the best user experience given the user device and the network available bandwidth. With this purpose, a controller selecting the video bitrate (or level) from a discrete set $\mathcal{L}$ has to be designed. The control goal is to maximize the video bitrate while avoiding playback interruptions and minimizing video bitrate switches. In this paper we propose a hybrid dynamical system modeling the essential features of an important class of controllers for adaptive video streaming systems. We derive tuning rules to achieve key performance goals by sizing the control system parameters. We show how to: (i) tune the controller parameters to keep the video level switching frequency below a given target; (ii) design the video levels set $\mathcal{L}$ to obtain a performance trade-off between switching frequency and storage costs at the servers; (iii) find the minimum amount of playout buffer that should be stored to avoid rebuffering events with a given probability in case of temporary bandwidth drop. The theoretical results are validated through numerical simulation and experimental evaluation.**

***Keywords*—Adaptive video streaming, Hybrid modeling.**

I. Introduction and Background

VIDEO streaming systems allow a client to play a video that is sent by a remote server over the Internet. Video streaming platforms, such as YouTube and Netflix, are credited today as the largest contributors for the downlink bandwidth traffic in the United States and it has been predicted that such growth will lead video to globally dominate the Internet traffic in the forthcoming years [1]. As a consequence, video content providers have to deal with the twofold challenge of (i) providing a seamless multimedia experience across a heterogeneous mix of client devices (such as smart TVs, desktop PCs, smart phones) and access networks (such as wired cable/ADSL and wireless 3G/4G connections), and (ii) managing a complex delivery network in a cost effective way. From the video providers point of view, improving user engagement is the key requirement due to its direct connection to revenues. Among other subjective factors that impact user engagement, the Quality of Experience (QoE) plays a fundamental role.

Video clients employ a playout buffer to absorb the instantaneous mismatches between the video encoding bitrate and the network available bandwidth that in best-effort Internet is unpredictable and time-varying. It has been shown that playback interruptions due to video playout buffer depletion are highly detrimental for the QoE and the user engagement [2]. Intuitively, in order to avoid that the playout buffer gets empty, the video bitrate should not be higher than the available network bandwidth. This requires (i) the video content to be made elastic, i.e. its bitrate can be changed in real-time, (ii) the design of a control algorithm that dynamically selects the video bitrate. Today, the leading approach to make the content elastic, i.e. to implement adaptivity, is the *stream-switching* (or *multi-bitrate*): the server encodes the video content at different bitrates, the *video levels*, and the control algorithm selects the video level to be sent. Due to its implementation and deployment simplicity, such a technique is today employed by leading video streaming services such as Netflix, Hulu, Vudu, Livestream, and YouTube. The two main adaptive streaming standards, MPEG Dynamic Adaptive Streaming over HTTP (DASH) and HTTP Live Streaming (HLS), adopt this approach. From the architectural point of view, the leading choice is the one placing the controller at the client.

Regarding the design of the stream-switching control algorithm, the following goals have to be pursued to improve the QoE: (i) avoiding playback interruptions; (ii) maximizing video quality (level or bitrate); (iii) minimizing the start-up delay; (iv) minimizing amplitude and frequency of video level switches [3]. Two cooperating techniques are employed: (i) an algorithm to dynamically select the video level, which is required to ideally match the available bandwidth and (ii) a playout buffer controller that is used to absorb bandwidth variations and avoid playback interruptions. Playout buffer control algorithms can be designed by taking one of the following approaches: the buffer level can be controlled by acting either on the received rate (*rate-based* approach) or on the video level (*level-based* approach). It is now well-known that the mainstream *rate-based* approach leads to fundamental performance issues such as poor bandwidth utilization and unfairness in presence of concurrent flows [4], [5], whereas the *level-based* approach is able to solve the issues affecting the rate-based approach at the cost of a possible increase of video level switches at steady state.

This performance limitation is the main reason hindering the adoption of level-based controllers. The goal of this paper, which significantly extends [6], is to characterize level-based *HTTP Adaptive Streaming* (HAS) controllers and propose rules to guide the design of controllers taking such an approach. To the purpose, we provide a model of level-based controllers in the form of a hybrid dynamical system that generalizes such controllers. Then, based on this model, we derive a simple relationship between minimum switching frequency

The authors are with the Dipartimento di Ingegneria Elettrica e dell'Informazione, Politecnico di Bari, Via Orabona 4, Bari, Italy. emails: giuseppe.cofano@poliba.it, luca.decicco@poliba.it, mascolo@poliba.it

This work has been supported by the Italian Ministry of Education, Universities and Research (MIUR) through the MAIVISTO project (PAC02L1_00061) and by the "Future in Research" project no. ACYBEH5 funded by the Apulia Region, Italy.

and control system parameters to provide a configurable bound to the number of video level switches at steady state. As a new contribution to [6], we propose a methodology to tune the minimum amount of playout buffer that should always be guaranteed to bound rebuffering events to a target probability.

II. Preliminaries

The following notation is used in the rest of the paper. $\mathbb{R}_{\geq 0}$ denotes the nonnegative real numbers. Given a set $\mathcal{A}$, $\overline{\mathcal{A}}$ denotes its closure. PD denotes the set of positive-definite functions. Given a vector $x \in \mathbb{R}^n$, $|x|$ denotes the Euclidean vector norm. Given a vector $x \in \mathbb{R}^n$ and a set $\mathcal{A} \subset \mathbb{R}^n$, the distance from x to $\mathcal{A}$ is denoted $|x|_{\mathcal{A}}$ and is defined by $|x|_{\mathcal{A}} := \inf_{y \in \mathcal{A}} |x - y|$. A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to the class $\mathcal{K}_\infty$ if it continuous, zero at zero, positive when its argument is positive, strictly increasing and unbounded.

A. Hybrid dynamical systems

A hybrid system $\mathscr{H}$ can be described using four elements $(\mathcal{C}, \mathcal{D}, F, G)$ as proposed in [7]:

$$\begin{cases} \dot{x} \in F(x) & x \in \mathcal{C}, \\ x^+ \in G(x) & x \in \mathcal{D}, \end{cases}$$

where x is the state, $\mathcal{C}$ is the *flow set* where x evolves according to a continuous equation, F is a set-valued mapping describing the continuous evolution of the state, $\mathcal{D}$ is the *jump set* where jumps are enabled, G is a set-valued mapping describing the discrete evolution of the state.

Solutions to a hybrid system are given on hybrid time domains by hybrid arcs. A subset $\mathcal{E} \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a compact hybrid time domain if $\mathcal{E} = \bigcup_{j=0}^{J-1} ([t_j, t_{j+1}], j)$ for some finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \leq ... \leq t_J$. It is a hybrid time domain if for all $(T, J) \in \mathcal{E}$, $\mathcal{E} \cap ([0, T] \times \{0, 1, ..., J\})$ is a compact hybrid time domain. A *hybrid arc* η is a function defined on a hybrid time domain $\mathrm{dom}\,\eta$ mapping to $\mathbb{R}^n$ such that $\eta(t, j)$ is locally absolutely continuous in t for each j, $(t, j) \in \mathrm{dom}\eta$. A hybrid arc η is a solution to the hybrid system $\mathscr{H}$ if $\eta(0, 0) \in \overline{\mathcal{C}} \cup \mathcal{D}$ and: (S1) For all $j \in \mathbb{N}$ and almost all t such that $(t, j) \in \mathrm{dom}\eta$, $\eta(t, j) \in \mathcal{C}$, $\dot{\eta}(t, j) \in F(\eta(t, j))$; (S2) For all $(t, j) \in \mathrm{dom}\eta$ such that $(t, j+1) \in \mathrm{dom}\eta$, $\eta(t, j) \in \mathcal{D}$, $\eta(t, j + 1) \in G(\eta(t, j))$.

B. Stability definitions

We are now ready to give the definition of stability for a hybrid system:

Definition 1: [Uniform global pre-asymptotic stability (UGpAS)[7, Chap. 3]] Consider a hybrid system $\mathscr{H}$ on $\mathbb{R}^n$ and a closed set $\mathcal{A} \subset \mathbb{R}^n$. The set $\mathcal{A}$ is said to be

- *Uniformly Globally Stable* for $\mathscr{H}$ if there exists a class $\mathcal{K}_\infty$ function α such that any solution ϕ to $\mathscr{H}$ satisfies $|\phi(t, j)|_{\mathcal{A}} \leq \alpha(|\phi(0, 0)|_{\mathcal{A}})$ for all $(t, j) \in \mathrm{dom}\phi$;
- *Uniformly Globally pre-Attractive* for $\mathscr{H}$ if for each $\epsilon > 0$ and $r > 0$ there exists $T > 0$ such that, for any solution ϕ to $\mathscr{H}$ with $|\phi(0, 0)|_{\mathcal{A}} \leq r$, $(t, j) \in \mathrm{dom}\phi$ and $t + j \geq T$ implies $|\phi(t, j)|_{\mathcal{A}} \leq \varepsilon$;
- *Uniformly Globally pre-Asymptotically Stable* for $\mathscr{H}$ if it is both uniformly globally stable and uniformly globally pre-attractive.

Stability of hybrid systems can be assessed by means of Lyapunov sufficient conditions. A function $V : \mathrm{dom}V \to \mathbb{R}$ is said to be a Lyapunov function candidate for $\mathscr{H} = (\mathcal{C}, \mathcal{D}, F, G)$ if V is continuously differentiable on an open set containing $\overline{\mathcal{C}}$ and $\overline{\mathcal{C}} \cup \mathcal{D} \cup G(\mathcal{D}) \subset \mathrm{dom}V$. A basic theorem provides sufficient conditions for a generic hybrid system $\mathscr{H}$. Moreover, several relaxed or weakened Lyapunov conditions hold for special cases. The following relaxed conditions hold for the special case of *persistent flowing*:

Proposition 1: [Persistent flowing [7, Chap. 3]] Let $\mathscr{H} = (\mathcal{C}, \mathcal{D}, F, G)$ be a hybrid system and $\mathcal{A} \subset \mathbb{R}^n$ a closed set. Let V be a Lyapunov function candidate for $\mathscr{H}$ and there exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, and a continuous $\rho \in PD$ such that:

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}) \, \forall x \in \mathcal{C} \cup \mathcal{D} \cup G(\mathcal{D}) \tag{1}$$

$$\langle \nabla V(x), f \rangle \leq -\rho(|x|_{\mathcal{A}}) \, \forall x \in \mathcal{C}, \forall f \in F(x) \tag{2}$$

$$V(g) - V(x) \leq 0 \, \forall x \in \mathcal{D}, \forall g \in G(x) \tag{3}$$

If, for each $r > 0$, there exists $\gamma_r \in \mathcal{K}_\infty$, $N_r \geq 0$ such that for every solution ϕ to $\mathscr{H}$, $|\phi(0, 0)|_{\mathcal{A}} \in (0, r]$, $(t, j) \in \mathrm{dom}\phi$, $t + j \geq T$ implies $t \geq \gamma_r(T) - N_r$, then $\mathcal{A}$ is *uniformly globally pre-asymptotically stable*.

III. Adaptive Video Streaming

Early literature on video streaming, dating back to '90s and later [8], [9], was focused on systems delivering content over the User Datagram Protocol (UDP). Since the UDP lacks a congestion control mechanism, the most important issue of such systems was the design of such an algorithm at the application layer. These early systems were making the assumption, which was recently proven wrong [10], that containing the video delivery latency was a key performance index for video streaming. As such, the designed congestion control algorithms did not implement a reliable transport leading to video quality degradation due to packet losses. It is important to notice that, even though many congestion control algorithms were proposed in the literature, the common practice adopted by the industry was to employ their own proprietary algorithms to deliver video. Such a practice has led to a remarkable fragmentation of the video streaming ecosystem which has hindered its diffusion. The situation changed in 2005 when YouTube adopted the so-called *HTTP progressive download* approach, i.e., the video was downloaded as any other file through an HTTP/TCP connection using a web browser. Such an approach has been improved by the *HTTP Adaptive Streaming* (HAS)[1] by adding the possibility of adapting the video

[1]Notice that recently Google has proposed QUIC, a protocol allowing HTTP traffic to be sent over UDP sockets [11], which is used today by YouTube to deliver videos to users. Despite the different transport protocol, QUIC is semantically equivalent to HTTP over TCP. The model proposed in this paper for HAS systems is valid both in the case of HTTP over TCP and HTTP over UDP (QUIC).

bitrate to the user device and end-to-end bandwidth. The HAS approach is today the dominant technology and is employed by all video streaming platforms. Based on the above background, the challenge today is to study new control mechanisms and mathematical models for the currently used video streaming systems employing the HAS approach.

In this paper we consider HAS adaptive video streaming control systems that take the stream-switching (or multi bitrate) approach: the server encodes the video content at different bitrate levels forming a discrete set $\mathcal{L} = \{l_0, l_1, \ldots, l_{N-1}\}$ with $l_i < l_{i+1}$. Each video level is then divided logically, or physically, into segments of fixed duration. A control algorithm dynamically selects the video level to be streamed at each segment download. The overall control goal of adaptive video streaming algorithms is to maximize the users perceived Quality of Experience given the available bandwidth.

A. Plant model

In this Section we present a fluid-flow model of the playout buffer length defined as the total duration of video, measured in seconds, temporarily stored in the buffer. We denote with $q(t)$ the *playout buffer length*[2]. Given a video of total duration T_v, each video frame can be uniquely associated to a time instant $t_v \in [0, T_v]$. We define the *video encoding bitrate* as $l = dD/dt_v$, where dD is the amount of bytes required to store a portion of video of duration dt_v. Indeed, by definition, the encoding rate l is always strictly greater than zero. We denote the video level selected by the controller with $l(t) \in \mathcal{L}$. The *received rate* $r(t)$ can be defined as $r(t) = dD/dt$, i.e. the amount dD of bytes that are received in a time interval dt.

As any storage element, the playout buffer length can be modeled as an integrator

$$\dot{q}(t) = f_r(t) - d_r(t), \tag{4}$$

where $f_r(t)$ is the *filling rate* and $d_r(t)$ is the *draining rate*. If an amount of video duration dt_v is received by the client and stored in the playout buffer in a time dt, the instantaneous filling rate $f_r(t)$ is equal to dt_v/dt by definition. Thus, from $f_r(t) = (dt_v/dD) \cdot (dD/dt)$ it turns out that

$$f_r(t) = \frac{r(t)}{l(t)}. \tag{5}$$

The playout buffer is drained by the player decoder: when the video is playing, τ seconds of video are played in τ seconds, i.e. $d_r(t) = 1$; when the player is paused the draining rate is zero. Thus, the draining rate is given by

$$d_r(t) = \begin{cases} 1 & \text{playing,} \\ 0 & \text{paused.} \end{cases} \tag{6}$$

Finally, substitution of (5) and (6) in (4) yields to

$$\dot{q}(t) = \frac{r(t)}{l(t)} - d_r(t). \tag{7}$$

[2]For the rest of the paper we will use the terms "buffer" and "queue" equivalently.

B. Control approaches

In [6] we have proposed a classification for HAS control systems based on the employed actuation mechanism: the buffer level, in fact, can be controlled by acting either on the received rate (*rate-based approach*) or on the video level (*level-based approach*). The employed approach has a remarkable impact on performance.

In the following we describe the two classes of control approaches that can be used to design a video streaming system following the classification introduced in [6].

Rate-based approach. Let us consider the *rate-based* approach and for simplicity let us assume that the end-to-end available bandwidth is fixed and thus the received rate $r(t)$ is constant and equal to $\overline{r}$. These systems select the video level $l(t)$ as the maximum level $\overline{l} \in \mathcal{L}$ less than the received rate $\overline{r}$. Since $\overline{l} < \overline{r}$ it turns out that, according to (5), the filling rate would be greater than 1, i.e. the queue would always grow. Hence, with this approach, the received rate is required to be set equal to $l(t)$ to keep $q(t)$ at a fixed set point q_T. However, the client cannot arbitrarily set $r(t)$ to a desired value since, for each t, video segments are downloaded at a rate equal to $\overline{r}$. Thus, the only way to achieve, at least on average, the desired received rate $r(t)$ is to insert idle periods between the download of two consecutive video segments. In other words the client alternates between ON and OFF phases: during the ON period, the client receives at a rate $r(t) = \overline{r}$, whereas during the OFF period it stays idle, i.e. $r(t) = 0$. With this naive control approach the average received rate in an ON-OFF period can be made equal to the selected video level $l(t)$ by properly setting the OFF duration. The advantage of this approach is that, if the end-to-end bandwidth is constant, the video level is kept constant and the queue tracks the set point.

Despite its simplicity, this approach has two major drawbacks extensively studied in the literature: (i) the available bandwidth is always underutilized, since the selected video level is lower than the available bandwidth; (ii) it has been experimentally shown that the ON-OFF traffic pattern causes the video flows to obtain a bandwidth share significantly less than the fair one when competing with long-lived TCP flows [5], [4]. The first issue can significantly degrade the perceived QoE in case the distance between the levels is high. The second issue, that is known in the literature as *downward spiral effect* [4], [12], can lead to an even worse degradation of the perceived QoE.

Level-based approach. In the case of the *level-based* approach (see for instance [12]), the video segments are downloaded back to back, thus eliminating the ON-OFF traffic pattern, i.e. $r(t)$ is always equal to $\overline{r}$. In this way video flows behave as any other TCP long-lived flow and, as a consequence, full utilization and fairness with TCP long-lived flows are achieved by design. The control is done by throttling $l(t)$ in order to avoid rebuffering events. The drawback of this approach is that at steady state video level switches occur even when the available bandwidth is constant since $l(t)$ belongs to the discrete set $\mathcal{L}$ and cannot exactly match the available bandwidth.

In the following section we propose a model that captures

the essential dynamics of the closed-loop system obtained when a level-based controller is employed.

IV. Level-based Control

In this work we focus on level-based controllers since they allow to overcome the well-known underutilization and unfairness issues affecting rate-based controllers [4]. The main reason why rate-based controllers are still employed despite these issues is that level-based controllers, if not properly designed, may provoke an excessive number of video level switches even under constant available bandwidth, where the expected behavior of the controller would be to keep the video level constant. In this Section we show how to overcome such a limitation. To the purpose, Section IV-A provides a formal model of the closed-loop system obtained when using a level-based hysteresis controller. We argue that any level-based controller should reach a steady state dynamics matching the one reached by the proposed hysteresis controller. Then, Section IV-B proves the stability of the system and, based on this analysis, some key properties are derived which provide a bound to the number of video level switches at steady state. Throughout the whole Section we assume a constant available bandwidth input function.[3] Such an assumption allows us to analyze the stability of the system and to model steady state conditions that we want to investigate.

A. Hybrid Model

The control goals of a generic adaptive streaming control system are: G1) obtaining full utilization of the available bandwidth at steady state, i.e., $r(t) = B$; G2) preventing rebuffering events (occurring when the buffer gets empty), while keeping the buffer length as low as possible; G3) minimizing the video level switching amplitude and frequency at steady state. Let us briefly justify the three goals. Regarding goal G1, maximizing the video bitrate under the constraint of the available bandwidth allows to indirectly maximize the user experience [10]. Even though the design of control algorithms for video streaming directly controlling the video quality is an interesting and yet developing research area [13], [14], [15], this paper focuses on the mainstream bitrate adaptation approach currently used both in the literature and in the industry. We argue that, regardless of the taken control approach, the video bitrate cannot be neglected since it directly affects the dynamics of the playout buffer (see eq. (7)). Regarding G2, in addition to avoiding rebuffering events, the queue should be kept as low as possible to (i) avoid unnecessary segments downloads wasting network bandwidth, (ii) minimize client memory usage, and (iii) provide liveness in the case of live streaming. Regarding G3, when a video level switch from l_i to l_j occurs, the larger the distance between l_i and l_j, the higher the QoE impairment [3]. Additionally, high switching frequency should be prevented to avoid QoE degradation [3].

[3]Notice that home network scenarios represent the vast majority of video traffic [1]. In these scenarios the users share the same downlink channel of an ADSL/Cable connection. Step-like drops (increases) of the available bandwidth occur whenever a long-lived TCP flow is started (stopped) on the same channel.

With the level-based approach, the goal G1 is reached by design since segments are downloaded back to back and TCP guarantees full link utilization in the case of backlogged flows. Goals G2 and G3 are potentially in contrast since control of the playout buffer is performed by varying the video level $l(t)$. The avoidance of rebuffering events is reached at the price of the presence of level switches even at steady state, due to the fact that $l(t)$ belongs to the discrete set $\mathcal{L}$ and cannot exactly match B.

We propose a simple solution to this issue based on the employment of a deadzone. The key idea is to keep $q(t)$ in a deadzone $[q_L, q_H]$ instead of steering it to a setpoint q_T. When $q(t) \in [q_L, q_H]$, video level switches are inhibited. This design choice fully satisfies the goal G2, since the controller aims at preventing both buffer underruns (that might occur when $q(t) < q_L$) and excessive buffering (that might occur when $q(t) > q_H$). At the same time, it allows to minimize both the amplitude and the frequency of the level switches, making them predictable and tunable, as it will be shown in the following.

The proposed approach can be applied in principle to any level-based controller. In order to show its advantages, we analyze the performance of a simple controller, that acts by switching the video level $l(t)$ between the two adjacent video levels $\underline{l}$ and $\bar{l}$ such that $\underline{l} < B < \bar{l}$. In particular

$$\underline{l} = \max_{l \in \mathcal{L}} \; l \quad \text{s.t.} \quad l < B \tag{8}$$

$$\bar{l} = \min_{l \in \mathcal{L}} \; l \quad \text{s.t.} \quad l > B \tag{9}$$

The proposed controller increases the video level by selecting $l(t) = \bar{l}$ when $q(t)$ increases above the high threshold q_H, whereas it decreases the video level to $\underline{l}$ when $q(t)$ decreases below the low threshold q_L. When $q(t) \in [q_L, q_H]$, the video level is kept constant. The proposed controller represents a benchmark to compare level-based controllers making use of the deadzone $[q_L, q_H]$.

In the following we present a hybrid dynamical model $\mathscr{H}$ of the considered control system by employing the framework proposed in [7] and summarized in Section II. The formal model allows us to (i) rigorously prove that the queue length keeps within the range $[q_L, q_H]$ at steady state regardless of the initial conditions and (ii) provide some key properties of the system.

Before defining the model, we exclude some trivial cases. If $B = l_i$, i.e. the available bandwidth is exactly equal to one video level bitrate, no switching between adjacent levels at steady state occurs. However, this is a purely mathematical condition that never holds in the practice: we exclude it by imposing that $B \neq l_i$ for all $i \in \{0, \ldots, N-1\}$. Additionally, if $q(t)$ grows above a threshold $q_{\text{MAX}} \gg q_H$ due to the fact that $B > l_{N-1}$, the control algorithm reacts with a safety mechanism by employing the ON-OFF pattern to reduce the received rate and prevent the download of the entire video. Therefore, the proposed model holds only as long as $0 \leq q(t) \leq q_{\text{MAX}}$.

Let us now define the hybrid model $\mathscr{H}$. The state of the system is given by $x = \begin{bmatrix} q & l \end{bmatrix}^T \in \mathcal{X} = [0, q_{\text{MAX}}] \times \mathcal{L}$. For convenience of notation we define the following sets

$$\mathcal{C}_L = \{x \in \mathcal{X} : q < q_L\}, \mathcal{C}_H = \{x \in \mathcal{X} : q > q_H\},$$
$$\mathcal{C}_T = \{x \in \mathcal{X} : q_L \leq q \leq q_H\},$$
$$\mathcal{C}_{\text{sup}} = \{x \in \mathcal{X} : l = \bar{l}\}, \mathcal{C}_{\text{inf}} = \{x \in \mathcal{X} : l = \underline{l}\}.$$

The *flow set* $\mathcal{C}$ and *jump set* $\mathcal{D}$ are given by

$$\mathcal{C} = (\mathcal{C}_L \cap \mathcal{C}_{\text{inf}}) \cup \mathcal{C}_T \cup (\mathcal{C}_H \cap \mathcal{C}_{\text{sup}}),$$
$$\mathcal{D} = (\mathcal{C}_L \cap \mathcal{C}_{\text{sup}}) \cup (\mathcal{C}_H \cap \mathcal{C}_{\text{inf}}).$$

The flow map is defined as

$$f(x) = \begin{bmatrix} \frac{B}{l} - 1 & 0 \end{bmatrix}^T. \tag{10}$$

The jump map is given by

$$g(x) = \begin{cases} \begin{bmatrix} q & \underline{l} \end{bmatrix}^T & \text{if } x \in (\mathcal{C}_L \cap \mathcal{C}_{\text{sup}}), \\ \begin{bmatrix} q & \bar{l} \end{bmatrix}^T & \text{if } x \in (\mathcal{C}_H \cap \mathcal{C}_{\text{inf}}). \end{cases} \tag{11}$$

B. System properties

The following theorem ensures that the queue length keeps within the range $[q_L, q_H]$ regardless of the initial conditions.

Theorem 1: The set $\mathcal{A} = \mathcal{C}_T$ is uniformly globally pre-asymptotically stable (UGpAS) for the hybrid system $\mathscr{H}$.

Proof: We employ the sufficient Lyapunov condition given in Proposition 1. The distance of x from the set $\mathcal{A}$ is defined as

$$|x|_{\mathcal{A}} = \begin{cases} |x_1 - q_L| & x \in \mathcal{C}_L, \\ 0 & x \in \mathcal{C}_T, \\ |x_1 - q_H| & x \in \mathcal{C}_H. \end{cases}$$

Let us consider the following candidate Lyapunov function

$$V(x) = \frac{1}{2}|x|^2_{\mathcal{A}}$$

that satisfies condition (1). In order to prove (2) we compute:

$$\langle \nabla V(x), f \rangle = \begin{cases} (x_1 - q_L)(\frac{B}{\underline{l}} - 1) & x \in \mathcal{C}_L \\ 0 & x \in \mathcal{C}_T \\ (x_1 - q_H)(\frac{B}{\bar{l}} - 1) & x \in \mathcal{C}_H \end{cases}$$

We consider the two cases $x \in \mathcal{C}_L$ and $x \in \mathcal{C}_H$ separately. When $x \in \mathcal{C}_L$, since $B > \underline{l}$ it holds that $\frac{B}{\underline{l}} - 1 > 0$, hence $(x_1 - q_L)(\frac{B}{\underline{l}} - 1) \leq \gamma_L (x_1 - q_L)^2$, where $\gamma_L \in [-\frac{1}{q_L}(\frac{B}{\underline{l}} - 1), 0)$. When $x \in \mathcal{C}_H$, since $B < \bar{l}$ it holds that $-(1 - \frac{B}{\bar{l}}) < 0$, hence $(x_1 - q_H)(\frac{B}{\bar{l}} - 1) \leq \gamma_H (x_1 - q_H)^2$, where $\gamma_H \in [-\frac{1}{q_{MAX} - q_H}(1 - \frac{B}{\bar{l}}), 0)$. Condition (2) is satisfied. During jumps $q(t)$ does not change, hence $V(g) - V(x) = 0$ and condition (3) is verified. We still have to prove that the persistent flowing condition holds. Every solution is characterized by a transient period $\overline{T} \geq 0$, which depends on initial conditions, and is such that after each jump has to flow for at least a time $T_f > 0$ due to the fact that $\forall x \in \mathcal{X}$ the distance between $g(x)$ and $\mathcal{D}$ is strictly greater than 0. Hence, $(t, j) \in \text{dom}\phi$ implies $j \leq \frac{\overline{T}}{T_f} + \frac{t}{T_f}$, and thus from $t + j \geq T$ follows the condition of persistent flowing with $\gamma_r(s) = \frac{T_f}{1+T_f} s$ and $N_r = \frac{\overline{T}}{1+T_f}$, which concludes the proof. It is worth to notice that T_f has to be less than $\min((q_H - q_L)\frac{\bar{l}}{\bar{l} - B}, (q_H - q_L)\frac{\underline{l}}{B - \underline{l}})$. ■

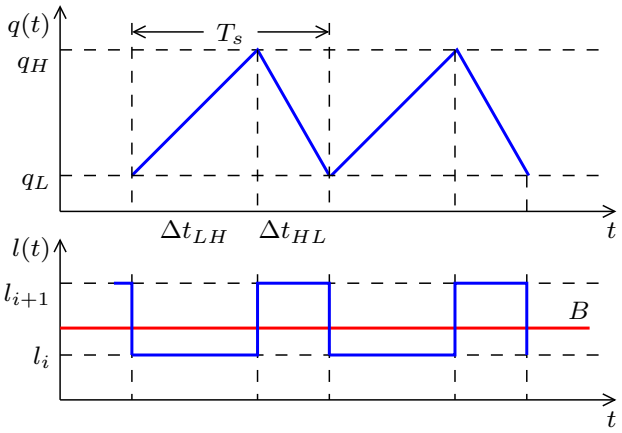


Fig. 1. Limit cycle dynamics of the playout buffer length $q(t)$ and the selected bitrate $l(t)$ at steady state

We can further characterize the dynamical behavior of the system with the following proposition.

Proposition 2: Given $B \in (l_i, l_{i+1})$, the evolution of the queue of $\mathscr{H}$ at steady state is a triangular wave with *switching period*

$$T_s = \Delta q \left(\frac{l_i}{B - l_i} + \frac{l_{i+1}}{l_{i+1} - B} \right),$$

where Δq is equal to $q_H - q_L$.

Proof: By solving (7) when $l_i < B$ is selected, we see that the queue linearly increases with constant rate $B/l_i - 1 > 0$. When q_H is reached, $l_{i+1} > B$ is selected and the queue linearly decreases with constant rate $B/l_{i+1} - 1 < 0$ until $q(t) = q_L$. Thus, l_i and l_{i+1} are alternately selected according to the jump map (11). Let us denote with Δt_{HL} (Δt_{LH}) the time elapsed to drain (fill) the queue from q_H to q_L(from q_L to q_H). We can write

$$\Delta t_{HL} = \Delta q \frac{l_{i+1}}{l_{i+1} - B}; \; \Delta t_{LH} = \Delta q \frac{l_i}{B - l_i}.$$

Thus, the period is given by

$$T_s = \Delta t_{LH} + \Delta t_{HL} = \Delta q \left(\frac{l_i}{B - l_i} + \frac{l_{i+1}}{l_{i+1} - B} \right). \tag{12}$$

Figure 1 shows the limit cycle dynamics described above. ■

Users' QoE depends on the switching period T_s, that is required to be as large as possible. We are able to provide a lower bound to T_s, which corresponds to the minimum value taken by T_s in the worst case available bandwidth B.

Proposition 3: The *minimum switching period* $\overline{T}_s$ is given by

$$\overline{T}_s = \frac{\Delta q D_i}{D_i + 2 - 2\sqrt{D_i + 1}}, \tag{13}$$

with $D_i = (l_{i+1} - l_i)/l_i$ (the video level relative distance), when $\overline{B} = \sqrt{l_i l_{i+1}}$.

Proof: By computing $\partial T_s/\partial B = 0$ we easily obtain

$$\overline{B} = \sqrt{l_i l_{i+1}}, \tag{14}$$

that is the geometric distance between the two adjacent levels. Substitution of (14) and $D_i = (l_{i+1} - l_i)/l_i$ in (12) yields, after a little algebra, to (13). ■

Remark 1: We can employ (13) to tune the distance $\Delta q = q_H - q_L$ between the queue thresholds such that a target worst case switching period is obtained. Since the function (13) is monotonically decreasing with a vertical asymptote in $D_i = 0$, the worst case switching period decreases as the relative distance of two adjacent levels increases. The knowledge of this simple relation can be used to properly design a level-based controller in order to ensure the required QoE. Motivated by this result we argue the superiority of level-based controllers over the mainstream rate-based approach that is affected by several important issues [16], [4].

Corollary 1: The minimum switching period $\overline{T}_s$ is independent of $\overline{B}$ if and only if the *relative distance* between any adjacent video level is fixed, i.e. $D_i = D \; \forall i \in \{0, \ldots, N-1\}$.

This corollary expresses a key design choice that provides a predictable performance in terms of switching period across the entire range of bandwidths in $[l_0, l_{N-1}]$. Thus, this corollary can be used to properly design the video levels set. This will be further explained in the next section.

V. Video Levels Set Design

Let us consider the problem of the optimal design of the video levels set $\mathcal{L} = \{l_0, \ldots, l_{N-1}\}$. Our goal is the minimization of both QoE impairment due to video level switching frequency and storage cost at the video provider. The video levels set $\mathcal{L}$ is assumed to be composed of N bitrates, whose number is not fixed a priori. In a typical use case the minimum video level l_0 is known a priori, whereas the maximum l_{N-1} video level has to be kept higher than a given value $\bar{l}_{N-1}$.

We formulate the following multivariate optimization problem:

$$\begin{aligned} \min_{\mathcal{L}=\{l_0,\ldots,l_{N-1}\}} \quad & J(\mathcal{L}) = C_S(S(\mathcal{L})) + \alpha C_{f_s}(\bar{f}_s(\mathcal{L})) \\ \text{subject to} \quad & N > 1,\ N \in \mathbb{N} \\ & l_0 = \bar{l}_0 \\ & l_{N-1} \geq \bar{l}_{N-1}, \end{aligned} \tag{15}$$

where $C_s(\cdot)$ and $C_{f_s}(\cdot)$ are monotonically non-decreasing functions expressing, respectively, storage cost and switching frequency cost (i.e. QoE impairment); $\bar{f}_s(\mathcal{L})$ and $S(\mathcal{L})$ are minimum switching frequency[4] and storage as functions of the video levels set; α is a positive freely adjustable weighting parameter.

First of all, we show how we can simplify this problem by reducing it to a monovariate one. By taking into account the Remark 1, we could proceed by fixing a target video level switching period $\overline{T}_s$ and the hysteresis width[5] Δq in (13) to obtain a unique value of D, which is independent of i. We assume that D is upper bounded by $D_{\max} = (\bar{l}_{N-1} - \bar{l}_0)/\bar{l}_0$. Once D and l_0 are fixed, each video level l_i, for all $i \in \{0, ..., N-1\}$ can be expressed as

$$l_i = (1 + D)^i \bar{l}_0. \tag{16}$$

Hence, the number N of levels is a function of D

$$N(D) = \left\lceil \frac{\log\left(\frac{\bar{l}_{N-1}}{\bar{l}_0}\right)}{\log(D+1)} \right\rceil + 1. \tag{17}$$

Let us now consider again (15) applying the proposed approach. The switching frequency f_s is equal to $\frac{D+2-2\sqrt{D+1}}{\Delta q D}$, which is a univariate function of D independent of the video level i. Storage for a video of duration T_v is proportional to the sum of the size of all video levels:

$$S(D) = T_v \sum_{i=0}^{N(D)-1} l_i = \frac{T_v \bar{l}_0}{D}((1 + D)^{N(D)} - 1). \tag{18}$$

We can finally formulate the resulting monovariate optimization problem:

$$\min_{D \in (0, D_{\max}]} J(D) = C_S(S(D)) + \alpha C_{f_s}(\bar{f}_s(D)).$$

Let us now assume for simplicity that both the cost functions are linear, i.e. $C_S(S(D)) = \gamma_1 S(D)$ and $C_{f_s}(\bar{f}_s(D)) = \gamma_2 \bar{f}_s(D)$. Observe that the storage cost is usually given by piecewise linear functions in the industry[6] and thus can be assumed to be linear once the operational point of the video provider is known. Regarding the switching frequency cost C_{f_s}, the relationship between the switching frequency and user engagement, which ultimately impacts video providers incomes, is still an open research issue. For simplicity it can be assumed to be linear. We obtain:

$$\min_{D \in (0, D_{\max}]} J(D) = \gamma_1 S(D) + \alpha \gamma_2 \bar{f}_s(D). \tag{19}$$

We expect that, when α is small, the storage component dominates and large values of D will be obtained, meaning that few and far apart levels should be employed. In particular, when $\alpha = 0$, the trivial solution $D = D_{max}$ (i.e. $\mathcal{L} = \{\bar{l}_0, \bar{l}_{N-1}\}$) is obtained. On the contrary, with increasing values of α the QoE component dominates and D gets smaller and smaller converging to 0.

Let us now analyze the derivatives of the two functions. The derivative of the switching frequency $\overline{f}_s = 1/\overline{T}_s$ is equal to:

$$\frac{\partial f_s}{\partial D} = \frac{1}{\Delta q} \frac{D + 2 - 2\sqrt{D+1}}{D^2 \sqrt{D+1}}. \tag{20}$$

[4]The worst-case (i.e. the maximum) over the $N-1$ minimum switching frequencies obtained by considering the $N-1$ pairs of consecutive video levels l_i and l_{i+1}.

[5]Recall that Δq cannot be made too large (see the control goal G2 in Section IV-A).

[6]For instance, see https://www.cdn77.com/pricing\#cdn-storage and https://aws.amazon.com/it/cloudfront/pricing/

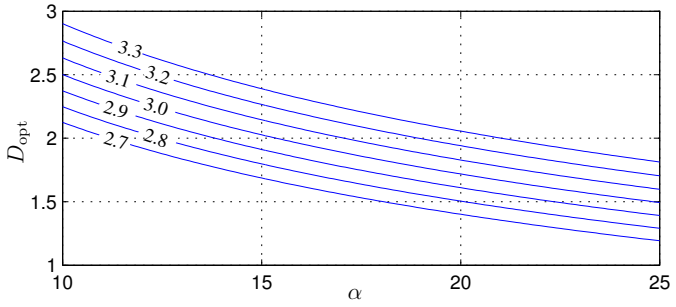


Fig. 2. Family of functions $D_{\text{opt}}(\alpha)$ when the parameter $A = \log \bar{l}_{N-1} - \log \bar{l}_0$ varies in the range [2.7, 3.1]

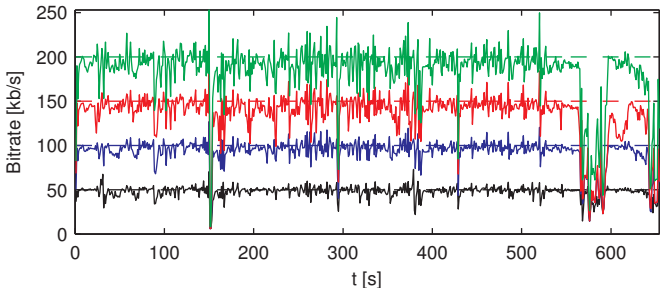


Fig. 3. Mismatch between nominal (dashed lines) and actual encoding rates (continuous lines) of the video sequence "Elephant's Dream"

It can be shown that (20) is a monotonically decreasing function that is equal to $1/(4 \cdot \Delta q)$ for $D = 0$ and asymptotically converges to 0.

The derivative of (18) turns out to be:

$$\frac{\partial S}{\partial D} = \frac{1 - (1+D)^{\frac{\log \bar{l}_{N-1} - \log \bar{l}_0}{\log(D+1)}}}{D^2} \tag{21}$$

that is monotonically increasing and goes from $-\infty$ to 0. To minimize (19) we have to solve the following equation:

$$\gamma_1 \frac{\partial S}{\partial D} + \alpha \gamma_2 \frac{\partial \bar{f}_s}{\partial D} = 0$$

It turns out that the previous equation has only one zero depending on α. Figure 2 shows the optimal value of D as a function of α for several values of $A = \log \bar{l}_{N-1} - \log \bar{l}_0$. In particular, the optimal value of D monotonically decreases converging to 0 when α goes to $+\infty$, due to the fact that with a small α the storage term dominates, whereas with a large α the switching frequency term dominates, as we previously noted.

VI. Playout Buffer Lower Threshold q_L Design

In Section IV constant available bandwidth has been assumed to model steady state conditions. In this Section we relax such an assumption to investigate robustness issues due to sudden bandwidth drops. Observe that under time-varying available bandwidth scenarios the goal of the controller is to avoid rebuffering events, whereas video level switches have to be performed in order to adapt to such bandwidth variations. We first point out in Section VI-A that the main parameter affecting robustness of the system is the lower threshold q_L. Then, in Section VI-B we propose a methodology to tune q_L in order to reach a target rebuffering ratio probability.

A. Motivation

In Section IV (Theorem 1) we have shown that the playout buffer length of an *ideal* level-based control system is always bounded in the set $[q_L, q_H]$ at steady-state provided that the available bandwidth B is greater than the lowest video level l_0. In other words this means that, if $B > l_0$ always holds, rebuffering events do not occur. However, if the available bandwidth B gets lower than l_0, the lower saturation condition is met making the system open-loop and, consequently, letting the playout buffer get empty. In such cases, a rebuffering event can be avoided only if the amount of video already stored in the playout buffer is sufficiently large to compensate this temporary mismatch. Hence, the sizing of q_L is crucial to avoid rebuffering events in such situations. A trivial way to size q_L is to set it to a very large value, an unnecessarily conservative practice that is being increasingly used today in leading video on demand (VoD) systems such as Netflix[7] [17]. However, this practice is not advisable since it makes the playout buffers very large resulting in the unnecessary pre-fetch of many video segments wasting server and network bandwidth in the case of early user abandonment [18]. Moreover, in the case of mobile connections the user traffic has usually monthly quotas and it is particularly important for the user to limit downloads of segments that may be not played due to early abandonment.

Let us analyze why the condition $B < l_0$ can occur. The video levels l_i constituting the set $\mathcal{L}$ represent nominal bitrates that are typically set to drive the encoding process and then advertised by the video streaming server to the client through the video manifest [19]. In particular, video streaming systems (f.i., Netfix and YouTube) employ the *Variable BitRate* (VBR) encoding process [20]: a target video bitrate is set and the encoder strives to produce such a bitrate on average. However, the encoder adapts the video bitrate to the video scene content. For instance, highly dynamical scenes are encoded at a higher bitrate to produce an acceptable video quality [20]. As a consequence, the actual bitrate $l_i(t)$ of a video level is time varying and an instantaneous mismatch with the nominal bitrate l_i exists in practice. To give an example, Figure 3 shows the case of the benchmark video sequence "Elephant's Dream" made available in the MPEG-DASH dataset[8]. In Figure 3 we show the first four nominal video levels (dashed line) and compare them to the actual bitrate $l_i(t)$ (continuous line) of each segment. The figure shows that wide oscillations around the nominal value are present, with a remarkable mismatch in the time interval $570\text{s} < t < 600\text{s}$ that is due to a static scene.

Even though the video levels set is designed to cover the typical user device range of bandwidths and screen resolutions, there is no way to prevent the end-to-end available bandwidth

[7]Netflix employs a playout buffer target in the range of 150 seconds.

[8]http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/ElephantsDream/

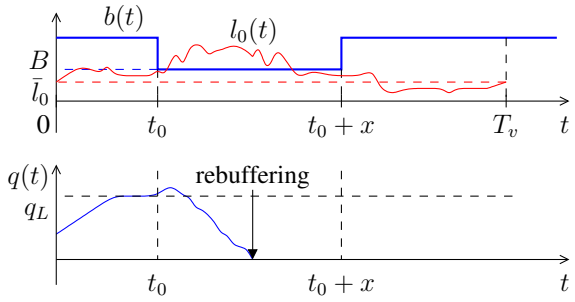


Fig. 4. Video level bitrate $l_0(t)$ and queue length $q(t)$ in the presence of a temporary bandwidth drop

to drop below the lowest video level bitrate l_0. This situation is commonplace in today wireless Internet, such as in the scenario where a user consumes the video through a smartphone with a 3G/4G connection.

B. The proposed methodology to size q_L

We consider a video of duration T_v and known[9] $l_0(t)$. Our goal is to size q_L so that, when the system is open-loop (the controller has selected l_0 and the bandwidth is less than l_0), rebuffering events are avoided with a given required probability.

A video streaming session can be seen as a Bernoulli trial having two possible outcomes: (i) no rebuffering event occurs during the session, (ii) one or more rebuffering events occur. We propose to tune q_L so that the probability $p_{\rm NR}$ of avoiding rebuffering events is higher than a given threshold $\overline{p}_{\rm NR}$. Hence, the problem we want to solve can be stated as follows:

Problem 1: Find the minimum q_L such that $p_{\rm NR} > \overline{p}_{\rm NR}$.

Remark 2: It is worth to notice that value of q_L obtained by solving Problem 1 can be advertised to the adaptive stream-switching algorithm at the start of the video playback by embedding it, for instance, in the video manifest file. Then, this value can be employed by the adaptive control system to improve its performance.

We make the following unrestrictive assumptions: (i) the system is open-loop since a generic time instant t_0 and for a certain duration x whose probability density function $f_X(x)$ is known; (ii) the system is at steady-state at t_0, i.e. $q_L \leq q(t_0) \leq q_H$.

To simplify the following discussion, we consider a constant bandwidth equal to B for $t \in [t_0, t_0 + x]$, and $q(t_0) = q_L$. Notice that if $B = \inf_{t\in[t_0,t_0+x]} b(t)$ and $q(t_0) = q_L$, by using the following procedure we obtain the most conservative setting for q_L. Figure 4 depicts an example in which a bandwidth drop occurring at time t_0 and lasting until time $t_0 + x$ provokes a rebuffering event due to the fact that, even though the nominal level $\bar{l}_0$ is less than B, during the bandwidth drop interval the actual level $l_0(t)$ is larger than B.

[9]The knowledge of $l_0(t)$ is not a restrictive assumption if we consider the case of Video on Demand (f.i. YouTube or Netflix) where $l_0(t)$ is known both at the client, through the manifest file, and at the server where the video is stored.

In order to compute $p_{\rm NR}$ we apply the law of total probability and obtain:

$$p_{\rm NR} = \int_0^\infty p_{\rm NR}(x) f_X(x) dx, \tag{22}$$

where $p_{\rm NR}(x)$ is the probability that no rebuffering occurs, known that the open-loop duration is equal to x.

In order to find $p_{\rm NR}(x)$, we compute the measure of the no rebuffering event set NR, a subset of the sample space $\mathcal{T}$

$$\mathcal{T} = \{t_0 \in [0, T_v]\}.$$

We define the no rebuffering event set as follows:

$$\mathrm{NR} = \{t_0 \in \mathcal{T} : \forall t \in [t_0, t_0 + x],\ q(t) > 0\} \tag{23}$$

In other words, $t_0 \in \mathrm{NR}$ means that the dynamics of the queue $q(t)$ that we obtain by solving (7) is always strictly positive for the whole duration of the open-loop time interval $[t_0, t_0 + x]$. Thus, to check if $t_0 \in \mathrm{NR}$ we need to compute $q(t)$ at a generic time instant $t \in [t_0, t_0 + x]$ by solving (7)

$$q(t) = q(t_0) + B \int_{t_0}^{t} \left(\frac{1}{l_0(\xi)} - 1\right) d\xi. \tag{24}$$

We notice that, since $l_0(t)$ is known and not null for all $t \in [0, T_v]$, it is always possible to compute offline $\int_{t_0}^{t} d\xi / l_0(\xi)$ for each $t_0 \in [0, T_v]$ and $t \in [t_0, T_v]$. We consider the indicator function $\mathbb{1}_{\rm NR}$ of $\mathrm{NR} \subset \mathcal{T}$, that is defined as

$$\mathbb{1}_{\rm NR}(t_0) = \begin{cases} 1 & \text{if } t_0 \in \mathrm{NR}, \\ 0 & \text{if } t_0 \notin \mathrm{NR}. \end{cases}$$

A well-known property of indicator functions states that $\mathbb{1}_{\rm NR}$ is a random variable whose expected value is equal to the probability of NR

$$E[\mathbb{1}_{\rm NR}] = \mathrm{P}(\mathrm{NR}) = p_{\rm NR}(x).$$

As expected, $p_{\rm NR}(x)$ is a function of q_L and B since $\mathbb{1}_{\rm NR}$ is based on condition $q(t) > 0$ that depends on such variables. Now, by plugging $p_{\rm NR}(x)$ into (22), we obtain $p_{\rm NR}$. Finally, by knowing $p_{\rm NR}$ we can find the minimum q_L such that $p_{\rm NR} > \overline{p}_{\rm NR}$. A way to implement the proposed technique is shown in Section VI-C.

C. Implementation

In this Section we show how the methodology proposed in Section VI-B can be implemented in the case of a DASH video streaming system. The procedure has been implemented by considering a discretization of (7) with a sampling interval equal to T_d. The video duration T_v is divided in $K = \lceil T_v / T_d \rceil$ intervals. For each discrete time interval $k \in \{1, \ldots, K\}$, $l_0(k)$ is discretized as

$$l_0(k) = \frac{1}{T_d} \int_{(k-1)T_d}^{kT_d} l_0(t) dt. \tag{25}$$

In order to compute $p_{\rm NR}(x)$ in (22) we define the following function:

Input: x, $\mathbf{l}_0$, q_L, B
Output: No rebuffering probability $p_{\rm NR}$

```
for k0:=1:K − x do
    for k:=k0:k0 + x do
        if q(k0, k, B, l0, qL) < 0 then
            buff_count ← buff_count + 1
            Break
        end if
    end for
end for
p_NR ← 1 − buff_count/(K − x)
```

Fig. 5. The pseudo-code to compute $p_{\rm NR}(x)$

$$p_{\rm NR}(x) = \text{buffProb}(x, B, \mathbf{l}_0, q_L)$$

where $\mathbf{l}_0$ is the array of segment bitrates computed using (25).

The pseudo-code of buffProb is given in Figure 5. We denote with k_0 the sampling interval at which the bandwidth drop begins. The key idea is to check for any $k_0 \in \{1, \dots, K-x\}$ if the queue gets empty, i.e. $q(k) \leq 0$, for some value of $k \in \{k_0, \dots, k_0 + x\}$. If this occurs we increase a rebuffering counter. Then, the number of rebuffering events is divided by $K - x$ to compute $E[\mathbb{1}_{\rm R}] = p_{\rm R}(x)$, that is the probability that one or more rebuffering events occur. The desired probability $p_{\rm NR}(x)$ is equal to $1 - p_{\rm R}(x)$, since we are considering a Bernoulli trial.

To conclude, we give a short discussion on how the proposed methodology, whose pseudo-code is described in Figure 5, can be used. To compute $p_{\rm NR}(x)$, we have to choose B which is the value assumed by $b(t)$ when it occurs a temporary bandwidth drop. Indeed, the choice of B involves a trade-off. Choosing a too low value of B might result in an overly conservative setting for q_L, while choosing B too high could result in having a too low q_L. Indeed, B can be tuned by leveraging statistical information on the bandwidth estimated by the clients. It is worth mentioning that, in order to size their content delivery networks, video providers continuously collect client-side measurements which include, among other metrics, the bandwidth estimated by the client. Using such datasets, B can be tuned, f.i., by taking the α percentile of the estimated bandwidth samples distribution. We believe that the proposed methodology can aid video providers in selecting the q_L parameter in a principled way so that the mentioned trade-off can be taken into account without resorting to rules of thumb based on experience. The way to select α ultimately depends on the distribution of the estimated bandwidth samples and is out of the scope of this paper.

VII. Validation

In this section we validate the hybrid model $\mathscr{H}$ (see Section IV-A) and its properties by comparing numerical simulations with experimental data. We recall that the proposed model $\mathscr{H}$ makes the fluid-flow assumption, i.e. the video segments length is infinitesimal, and assumes that the video levels encoding rates are constant. In order to improve modeling accuracy in this section we also consider a refined hybrid model $\mathscr{H}_R$ relaxing both the aforementioned assumptions. In particular, $\mathscr{H}_R$ introduces the concept of discrete video segment by allowing control decisions to be made only after a segment has been completely downloaded. This also has allowed us to employ in the simulations video levels with time-varying encoding rate. The added features make $\mathscr{H}_R$ difficult to mathematically analyze and do not add, as it will be shown in the following, a significant theoretical contribution. For this reason the formal model is not included in this paper. Both the proposed hybrid models $\mathscr{H}$ and $\mathscr{H}_R$ have been implemented with the Matlab *Hybrid Equations (HyEq) Toolbox* [21].

The proposed control algorithm has been then implemented using `TAPAS` [22], an open-source tool that allows video streaming control algorithms to be implemented and tested through experiments in a real network such as the Internet.

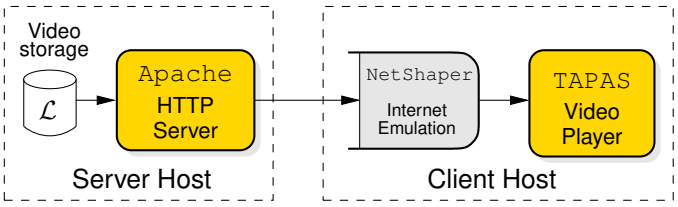


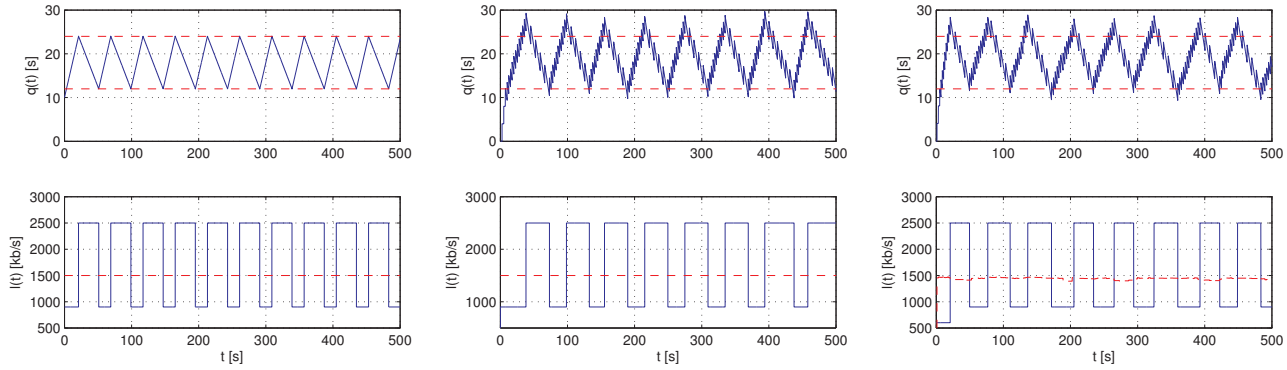
Fig. 6. Testbed employed for the experimental evaluation

Figure 6 shows the employed testbed that is composed of two hosts connected through a 1 Gbps switch: the *server host* is a workstation with a Debian Linux operating system equipped with the software `Apache`[10] that acts as HTTP server; the server host also stores the video sequence "Sintel"[11], encoded at five nominal bitrates $\mathcal{L} = \{300, 600, 900, 2500, 4000\}$kb/s (except in Section VII-B, where other bitrates are considered); the *client host* is a Ubuntu Linux machine that runs the `TAPAS` tool in order to download video segments from the HTTP server and play the video; `TAPAS` measures several variables such as the video level $l(t)$ selected by the controller and the playout buffer length $q(t)$ and stores it in log files [22]; moreover, the client host runs `NetShaper`, a tool developed by us that permits to set the bandwidth $b(t)$ and the delay of the link connecting the client to the server host to emulate an Internet connection.

A. Hybrid Model

In this section we compare the dynamics of the two proposed models $\mathscr{H}$ and $\mathscr{H}_R$, obtained via numerical simulations, with the dynamics of the real system obtained through experiments. To the purpose, Figure 7 shows the state dynamics $x(t) = [\ q(t) \quad l(t)\ ]^T$ in the case of simulations (Figure 7(a) and Figure 7(b)) and experimental runs (Figure 7(c)) when $\Delta q = 12$s and the available bandwidth B is equal to 1500kb/s. By comparing Figure 7(a) and Figure 7(c), it turns out that $\mathscr{H}$ models well the dynamics of the real control system. The only

[10] http://httpd.apache.org/
[11] http://www.sintel.org/



(a) Hybrid model $\mathscr{H}$ (b) Refined hybrid model $\mathscr{H}_R$ (c) Real system

Fig. 7. Dynamics of $q(t)$ and $l(t)$ obtained through numerical simulations and network experiments

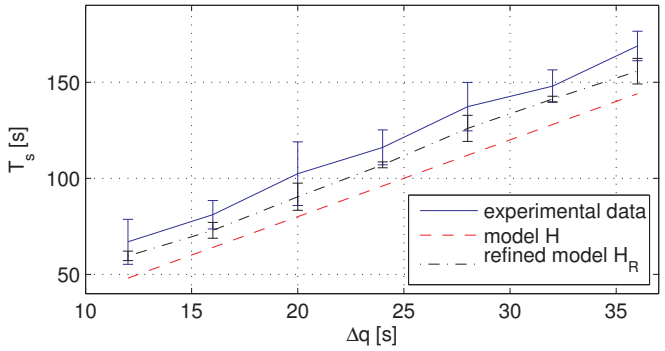


Fig. 8. The switching period T_s obtained through numerical simulations and network experiments as a function of Δq

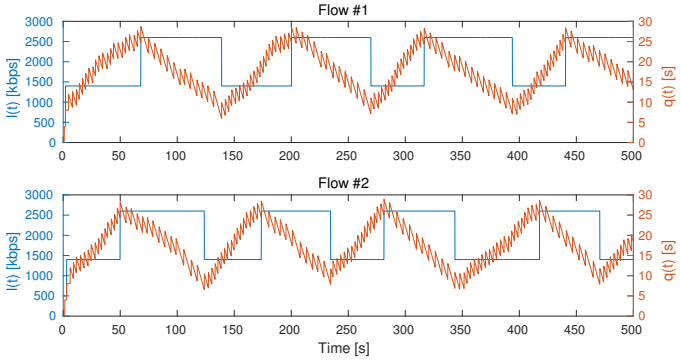


Fig. 9. Two video flows sharing a bottleneck with capacity $C = 4$ Mbps

notable difference is that in the real system the queue gets slightly below (above) the lower (higher) threshold before a video level switch is triggered, whereas $\mathscr{H}$ triggers the level switch exactly when the queue gets equal to the thresholds. This difference is due to the fact that $\mathscr{H}$ is a fluid-flow model and considers segments of infinitesimal size. On the other hand, the dynamics of the refined model $\mathscr{H}_R$ shown in Figure 7(b) accurately matches the real system dynamics.

We have then validated the average switching period T_s given by (12) both on the real system and on the refined model $\mathscr{H}_R$. Several runs, each one with a different $\Delta q \in \{12, 16, 20, 24, 28, 32, 36\}$s, have been carried out. In each run the average switching period T_s has been calculated. In Figure 8 it is shown that (12) (dashed line) fits quite accurately the average measured T_s of the real system (solid line). As expected, the average measured T_s in the case of the refined model $\mathscr{H}_R$ (the dash-dotted line in Figure 8) achieves a higher accuracy. Even though the proposed model $\mathscr{H}$ achieves lower accuracy with respect to the refined model $\mathscr{H}_R$, it has the merit of giving a closed form expression of T_s which can be considered as a worst case lower bound. It is worth to notice that such lower bound is not overly conservative.

To conclude, we experimentally validate that the findings obtained in Section IV also hold in a multiuser scenario. To the purpose we consider the case of two concurrent video streaming sessions sharing a bottleneck with a constant capacity $C = 4$ Mbps playing the same video encoded with a video level set $\mathcal{L} = \{0.24, 0.5, 0.9, 1.4, 2.6, 4.0, 5.0\}$ Mbps. Both the flows employ the same controller settings, i.e., $q_L = 12$ s and $q_H = 28$ s. Assuming ideal TCP fairness, the fair share is equal to $B = C/2 = 2$ Mbps. Under such settings, according to Proposition 2 both the flows should select a video level periodically switching between $l_3 = 1.4$ Mbps and $l_4 = 2.6$ Mbps with a switching period equal to 106 s (see (12)).

Figure 9 shows the results of the experiment. The figure shows that the limit cycle of the playout buffer length $q(t)$ and the selected video level $l(t)$ is also valid in the case of a multiuser scenario. The measured average switching period for flow #1 and #2 are respectively 125 s and 121 s, with an error of less than 20 s compared to the nominal switching period of 106 s computed using (12)). This mismatch between the computed and measured switching period is compatible with the accuracy that we have discussed above (see Figure 8). Finally, the negligible difference between the switching periods measured for the two flows is due to the slightly different

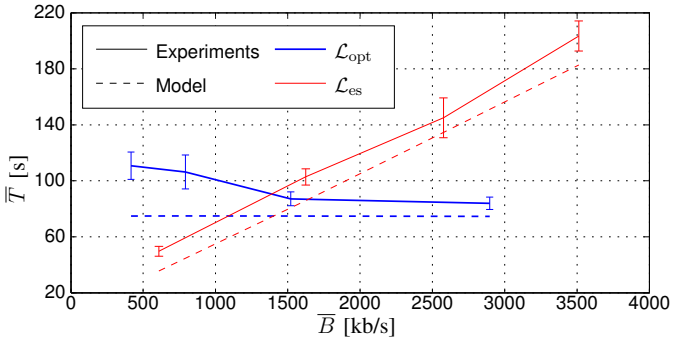


Fig. 10. Comparison between the video level switching period obtained with optimal level set $\mathcal{L}_{\rm opt} = \{300, 573, 1094, 2090, 4000\}$kb/s or with equally spaced levels $\mathcal{L}_{\rm es} = \{300, 1225, 2150, 3075, 4000\}$kb/s as a function of $\overline{B}$

bandwidth share obtained by the two videos.

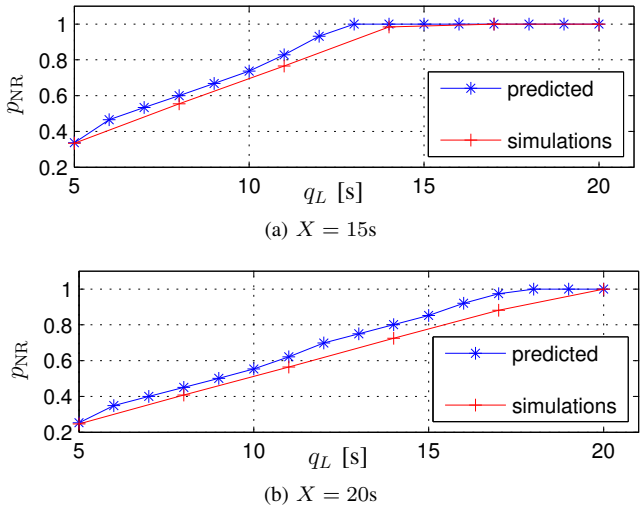
To summarize, the validation has shown that the model $\mathscr{H}$ proposed in Section IV fits with good accuracy the real system and that the refined model $\mathscr{H}_R$, though more complex, achieves a very high accuracy.

B. Design of the Video Levels

In this Section we provide the validation of the procedure proposed in Section V to design the video levels by taking care of the trade-off between level switches and storage at the server. We have experimentally compared $\overline{T}_s$ obtained when video levels are designed according to the optimal procedure proposed in Section V to the $\overline{T}_s$ obtained with equally spaced levels with interval ΔL. With the latter procedure the predicted $\overline{T}_s$ is equal to

$$\overline{T}_s(l_i) = \frac{\Delta q}{\Delta L}(\Delta L + 2l_i + 2\sqrt{\Delta L \cdot l_i + l_i^2}). \qquad (26)$$

The video levels sets have been designed in the same range $[300\,\text{kb/s}, 4000\,\text{kb/s}]$ and the cardinality of the two sets is the same. In each experiment the available bandwidth is set equal to the worst case bandwidth $\overline{B}$ for each couple of adjacent levels (l_i, l_{i+1}) according to (14). With the optimal procedure described in Section V the set $\mathcal{L}_{\rm opt} = \{300, 573, 1094, 2090, 4000\}$kb/s is obtained. The average measured worst case $\overline{T}_s$ is shown in Figure 10 (thick line) and compared to the one computed with (13) (thick dashed line). It has to be noticed that encoders are not able to produce video levels that precisely match the target levels for low levels. In fact, we have measured that the actual average bitrates of l_0 and l_1 are equal to, respectively, $342\,\text{kb/s}$ and $595\,\text{kb/s}$, with a relative error equal to 0.14 and 0.03 compared to the target levels. Despite of this, (12) nicely predicts the worst case switching period $\overline{T}_s$. With the equally spaced design procedure (thin line), the video levels set $\mathcal{L}_{\rm es} = \{300, 1225, 2150, 3075, 4000\}$kb/s is obtained and $\overline{T}_s$ grows approximately linearly, as expected from (26) (thin dashed line).



(a) $X = 15$s

(b) $X = 20$s

Fig. 11. No Rebuffering probability $p_{\rm NR}$ function of q_L

C. Design of the Playout Buffer Lower Threshold

This Section validates the design procedure to size q_L that we have proposed in Section VI-B. To the purpose, we have compared the predicted probability $p_{\rm NR}$ to the ratio of no rebuffering events obtained running simulations of the model $\mathscr{H}_R$ in the presence of a random bandwidth drop. The bandwidth drop starts in a randomly uniformly distributed time instant during the playback in both cases.

We have considered $p_{\rm NR}$, the probability of no rebuffering ratio, in the case that x is uniformly distributed with distribution $U(0, X)$. The validation has been done by comparing the value computed through the sizing procedure proposed in Section VI-B with the ratio of no rebuffering events measured through simulations obtained using the model $\mathscr{H}_R$. We have used the simulations as a term of comparison in this case, instead of network experiments, due to the very large time, in the order of weeks, they would have taken. The accuracy of the refined model $\mathscr{H}_R$ has been shown in the Section VII-A.

We have considered the probability $p_{\rm NR}$ obtained with q_L ranging in the interval $[2, 20]$s when a bandwidth drop to 50kbps occurs. The duration of the bandwidth drop is random and uniformly distributed with distribution $U(0, X)$.

The results are shown in Figure 11 (a) for the case $X = 15$s and in Figure 11 (b) $X = 20$s. In both cases the prediction is quite accurate and the measured prediction error is always lower than 0.1.

VIII. Conclusions

In this paper we have considered an important class of adaptive video streaming control systems. We have proposed a hybrid dynamical model of a threshold-based controller that can be considered as benchmark for such a class. Based on this model, we have provided tuning rules of the system parameters to minimize video level switches, which is the main drawback of employing such class of controllers. We have next shown how to design the video levels set $\mathcal{L}$ to obtain optimal trade-off between switching frequency and storage cost

requirements. Finally, we have proposed a procedure to tune the lower threshold in order to achieve the desired rebuffering probability in the presence of a temporary bandwidth drop below the minimum video level bitrate. Theoretical findings have been validated by comparing numerical simulations and experimental results, showing that the proposed model fits with good accuracy the behavior of the real system and that the derived properties are able to predict the system performance in terms of video level switching frequency and no rebuffering probability.

REFERENCES

[1] Cisco, "Cisco Visual Networking Index:Forecast and Methodology 2013-2018," 2013.

[2] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," in *Proc. of ACM SIGCOMM*, pp. 339–350, 2013.

[3] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "A case for a coordinated internet video control plane," in *Proc. of ACM SIGCOMM*, pp. 359–370, 2012.

[4] T. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: picking a video streaming rate is hard," in *Proc. of ACM IMC*, 2012.

[5] S. Akhshabi, L. Ananthakrishnan, A. C. Begen, and C. Dovrolis, "What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth?," in *Proc. of ACM NOSSDAV*, pp. 9–14, 2012.

[6] G. Cofano, L. D. Cicco, and S. Mascolo, "Characterizing adaptive video streaming control systems," in *Proc. of American Control Conference (ACC)*, pp. 2729–2734, July 2015.

[7] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[8] R. Rejaie, M. Handley, and D. Estrin, "Layered quality adaptation for internet video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2530–2543, 2000.

[9] D. McNamee, C. Krasic, K. Li, A. Goel, E. Walthinsen, D. Steere, and J. Walpole, "Control challenges in multi-level adaptive video streaming," in *proc. of IEEE CDC*, vol. 3, pp. 2228–2233, 2000.

[10] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[11] G. Carlucci, L. De Cicco, and S. Mascolo, "HTTP over UDP: An Experimental Investigation of QUIC," in *Proc. of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pp. 609–614, 2015.

[12] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: a Client-side Controller for Dynamic Adaptive Streaming over HTTP (DASH)," in *Proc. of Packet Video Workshop*, Dec. 2013.

[13] Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran, "Streaming video over http with consistent quality," in *Proc. of the ACM Multimedia Systems Conference*, pp. 248–258, 2014.

[14] F. Chiariotti, S. D'Aronco, L. Toni, and P. Frossard, "Online learning adaptation strategy for dash clients," in *Proc. of the ACM Multimedia Systems Conference*, pp. 8:1–8:12, 2016.

[15] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *Proc. of ACM SIGCOMM*, pp. 325–338, 2015.

[16] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. of ACM MMSys*, pp. 157–168, 2011.

[17] T. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proc. of ACM SIGCOMM*, pp. 187–198, 2014.

[18] A. Finamore, M. Mellia, M. Munafo, R. Torres, and S. G. Rao, "Youtube everywhere: Impact of device and infrastructure synergies on user experience," in *Proc. of ACM IMC*, pp. 345–360, 2011.

[19] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.

[20] D.-K. Kwon, M.-Y. Shen, and C.-C. J. Kuo, "Rate control for h.264 video with enhanced rate and distortion models," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 5, pp. 517–529, 2007.

[21] R. Sanfelice, D. Copp, and P. Nanez, "A toolbox for simulation of hybrid systems in Matlab/Simulink: hybrid equations (HyEQ) toolbox," in *Proc. of HSCC*, pp. 101–106, 2013.

[22] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "TAPAS: A Tool for rApid Prototyping of Adaptive Streaming Algorithms," in *Proc. of VideoNext*, pp. 1–6, 2014.

Giuseppe Cofano received the Telecommunications Engineering degree (Hons.) and the Ph.D. degree in Computer Science Engineering from Politecnico di Bari, Bari, Italy, in 2012 and 2016. Currently, he is a post-doc at Politecnico di Bari. He has held a visiting position at the University of Würzburg, Germany, in 2015. His main interests focus on the modeling and design of control algorithms for multimedia transport and adaptive video streaming.

Luca De Cicco (M' 14) received the computer science engineering degree (Hons.) and the Ph.D. degree in information engineering from Politecnico di Bari, Bari, Italy, in 2003 and 2008, respectively. Currently, he is an Assistant Professor at Politecnico di Bari since 2016. He has held visiting positions at the University of New Mexico, Albuquerque, NM, USA, in 2007; Ecole Superieure d'Electricité, Paris, France, in 2012; and the Laboratory of Information, Networking and Communication Sciences-LINCS, Paris, France, in 2013 and 2014. He is the co-author of more than 40 papers published in international journals, books, or conferences. His main interests focus on the modeling, analysis, and design of congestion control algorithms for multimedia transport, adaptive video streaming, and Session Initiation Protocol overload control.

Saverio Mascolo (SM' 14) received the Laurea degree (Hons.) in electronics engineering and the Ph.D. from Politecnico di Bari, Italy, in 1991 and 1994, respectively. Since 2001, he has been Associate Professor of Automatic Control at Politecnico di Bari. He is Full Professor since 2012. He was a Postdoctoral Researcher in 1995 and a Visiting Researcher in 1999 at the University of California, Los Angeles (UCLA) and Visiting Consultant at the University of Uppsala, Sweden, from 2002 to 2004. He has authored or co-authored more than 120 papers in international journals, books, or conferences. He has worked on congestion control in data networks (TCP and ATM), end-to-end bandwidth estimate, modeling, and control. His current research interests focus on the Future Internet, in particular, on the topic of real-time communication over the web. He has been Associate Editor of the IEEE Transactions on Automatic Control. Currently, he is Associate Editor of IEEE/ACM Transactions on Networking and of Computer Networks Journal, Elsevier.