Online Learning Based Congestion Control for Adaptive Multimedia Transmission

Oussama Habachi, Hsien-Po Shiang, Mihaela van der Schaar, *Fellow, IEEE*, and Yezekael Hayel

***Abstract*—The increase of Internet application requirements, such as throughput and delay, has spurred the need for transport protocols with flexible transmission control. Current TCP congestion control adopts an Additive Increase Multiplicative Decrease (AIMD) algorithm that linearly increases or exponentially decreases the congestion window based on transmission acknowledgments. In this paper, we propose an AIMD-like media-aware congestion control that determines the optimal congestion window updating policy for multimedia transmission. The media-aware congestion control problem is formulated as a Partially Observable Markov Decision Process (POMDP), which maximizes the long-term expected quality of the received multimedia application. The solution of this POMDP problem gives a policy adapted to multimedia applications' characteristics (i.e., distortion impacts and delay deadlines of multimedia packets). Note that to obtain the optimal congestion policy, the sender requires the complete statistical knowledge of both multimedia traffic and the network environment, which may not be available in practice. Hence, an online reinforcement learning in the POMDP-based solution provides a powerful tool to accurately estimate the environment and to adapt the source to network variations on the fly. Simulation results show that the proposed online learning approach can significantly improve the received video quality while maintaining the responsiveness and TCP-friendliness of the congestion control in various network scenarios.**

***Index Terms*—Congestion control, learning technology, multimedia communication.**

I. Introduction

TCP dominates today's communication protocols at the transport layer in both wireless and wired networks, due to its simple and efficient solutions for end-to-end flow control, congestion control and error control of data transmission over IP networks. However, despite the success of TCP, the existing TCP congestion control is considered unsuitable for delay-sensitive, bandwidth-intense, and loss-tolerant multimedia applications, such as real-time audio streaming and video-conferences (see [1] and [2]). There are two main reasons for this. First, TCP is error-free and trades transmission delay for reliability. In fact, packets may be lost during transport due to network congestion and errors. TCP keeps retransmitting the lost packets until they are successfully transmitted, even if this requires a large delay. The error-free restriction ignores delay deadlines of multimedia packets, i.e., the time by which they must be decoded. Note that even if multimedia packets are successfully received, they are not decodable if they are received after their respective delay deadlines. Second, TCP congestion control adopts an AIMD algorithm. This results in a fluctuating TCP throughput over time, which significantly increases the end-to-end packet transmission delay, and leads to poor performance of multimedia applications [2]. To mitigate these limitations, a plethora of research focused on smoothing the throughput of AIMD-based congestion control for multimedia transmission (see [3] and [4]). These approaches adopt various congestion window updating policies to determine how to adapt the congestion window size to the network environment. However, these approaches seldom explicitly consider the characteristics of multimedia applications, such as delay deadlines and distortion impacts. Note that although UDP is more suitable for real-time video streaming than TCP, there is a tremendous need for video over TCP due to the development of HTTP streaming that takes advantage of the web caching system. Moreover, many professional streaming systems use TCP because it can deal easily with firewalls. Hence, this motivates the need of developing improved TCP congestion control solutions for video streaming.

In this paper, we propose a media-aware POMDP-based congestion control, referred to as Learning-TCP, which exhibits an improved performance when transmitting multimedia. Unlike the current TCP congestion control protocol that only adapts the congestion window to the network congestion (e.g., the packet loss rate in TCP Reno and the RTT in TCP Vegas), the proposed congestion control algorithm also takes into account multimedia packets' distortion impacts and delay deadlines when adapting its congestion window size. Importantly, the proposed media-aware solution only changes the congestion window updating policy of the TCP protocol at the sender side, without requiring modifications to the feedback mechanisms at the receiver side.

The multimedia quality obtained by receivers is impacted by the network congestion incurred at bottleneck links. This information is only partially observable by senders and is based on feedback of network congestion signals. In order to capture the dynamics created by the network congestion and to optimize the expected long term quality of multimedia transmissions, we formulate the media-aware congestion control problem using a POMDP framework. The proposed framework allows users to evaluate the network congestion dynamics over time, and provides the optimal congestion window updating policy that maximizes the long-term discounted reward. In this paper, we con-

Manuscript received September 26, 2012; revised November 27, 2012 and December 03, 2012; accepted December 06, 2012. Date of publication January 01, 2013; date of current version February 25, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Min Dong. This work was initiated when O. Habachi and H.-P. Shiang were at UCLA, and O. Habachi, H.-P. Shiang, and M. van der Schaar acknowledge the support of the NSF CNS 0831549 grant.

O. Habachi and Y. Hayel are with the CERI/LIA, University of Avignon, 84911 Avignon, France (e-mail: oussama.habachi@univ-avignon.fr; yezekael.hayel@univ-avignon.fr).

H.-P. Shiang is with Cisco Systems Inc., San Jose, CA 95134 USA (hshiang@cisco.com).

M. van der Schaar is with the Department of Electrical Engineering, University of California, Los Angeles (UCLA), Los Angeles 90095-1594 USA (e-mail: mihaela@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2012.2237171

TABLE I
COMPARISONS OF CURRENT CONGESTION CONTROL SOLUTIONS FOR MULTIMEDIA STREAMING

Algorithm	Name of the congestion control	Type of TCP-Friendluness	Multimedia support	Distortion impact	Content dependency	Decision Type
Rejaie 1999 [12]	RAP	AIMD-based	Source rate adaptation	No	No	Myopic
Cai 2005 [3]	GAIMD	AIMD-based	Playback buffering	No	No	Myopic
Bansal 2001 [4]	Binomial Algorithm	Binomial scheme	Source rate adaptation	No	No	Myopic
Our approach	LEARNING-TCP	AIMD-like media aware	Quality-centric congestion control	Yes	Yes	Foresighted

sider that the multimedia quality is defined as the total distortion reduction of the received multimedia packets.

In practice, the sender needs to learn the network environment during transmission in order to adapt its congestion control policy. Hence, in this paper, we propose an online learning approach for solving the POMDP-based congestion control problem. Cassandra and Littman proposed the Witness algorithm in order to solve discounted finite-horizon POMDP using value iteration (see [5] and [6]). Note that Chrisman and McCallum studied, in [7] and [8], the problem of learning a POMDP model in a reinforcement learning setting. They used an extension of Q-learning [9] in order to approximate Q-functions for POMDP. They proposed Replicated Q-learning to generalize the Q-learning to vector valued states to solve POMDP models. Rumelhart explored, in [10], the reinforcement learning for POMDP models by adopting a belief-based MDP framework. However, all these reinforcement learning approaches suffer from the well-known curse of dimensionality problem, meaning that a practical POMDP problem involves an enormous state and action space, which significantly impacts the complexity and the convergence time to solve the problem. The authors of [11] proposed a TCP-friendly congestion control for multimedia transmission. Unlike our proposed congestion control, they did not propose an AIMD-like solution.

This paper presents a TCP-like window-based congestion control schemes that uses history information, in addition to the current window size and congestion feedback. A comparative study of several existing congestion control mechanisms for multimedia applications and the proposed solution is presented in Table I. In summary, the paper makes the following contributions:

A. Media-Aware Congestion Control

The proposed Learning-TCP provides a media-aware approach to adapt the AIMD-like congestion control policy to both varying network congestion and multimedia characteristics taking into account source rates, distortion impacts and delay deadlines of multimedia packets. Hence, the media-aware approach leads to a significantly improved multimedia streaming performance.

B. POMDP-Based Adaptation in the Dynamic Environment

We propose a POMDP framework to formulate the media-aware congestion control problem. It allows TCP senders to determine the congestion window size that maximizes the expected long-term quality of multimedia applications. Furthermore, , network users have only partial knowledge about the bottleneck link status. In fact, the number of packets in transmission over the bottleneck link queue depends not only on the congestion window of the user, which is known, but also on the congestion windows of all the other users, which cannot be observed. Therefore, the long term prediction and adaptation of the POMDP framework under partial observation of the system state is essential for multimedia streaming, since it can consider, predict, and exploit the dynamic nature of the multimedia traffic and the transmission environment, in order to optimize the application performance.

The POMDP solution is based on a set of updating policies composed of generic congestion control algorithms, with general increase and decrease functions like: AIMD, Inverse Increase/Additive Decrease (IIAD), Square Root inversely proportional Increase/proportional Decrease (SQRT), and Exponential Increase/Multiplicative Decrease (EIMD).

C. Online Learning for Delay-Sensitive Multimedia Applications

We unravel several structural properties of the optimal solution. Based on this, we propose a practical low-complexity online learning method to solve the POMDP-based congestion control problem on-the-fly.

The paper is organized as follows. In Section II, we model the media-aware congestion control problem that maximizes the performance of multimedia applications. Then in Section III, we formulate the problem using a POMDP based framework. Structural results and the proposed online learning method are discussed in Section IV. Section V provides simulation results and Section VI concludes the paper.

II. MEDIA-AWARE CONGESTION CONTROL PROBLEM FORMULATION

A. Network Settings

We assume that the network has a set of N end users indexed $\{1, \ldots, N\}$. Each user is composed of a sender node and a receiver node that establish an end-to-end transport layer connection. Let w_n represents the congestion window size of the user n. The network system has some bottleneck links, which results in packet losses when buffers are overloaded. Note that any user cannot observe the traffic generated by other users. In fact, an end user n can only infer the congestion status by observing feedback information from acknowledgments per RTT. For each acknowledgment, the end user n observes a congestion event $o_n \in O_n = \{success, fail\}$ (the packet being received successfully or not by the receiver). We consider a time-slotted system with a slot duration of one RTT. Moreover, we assume that the user n has a delay vector $delay_n$ of all packets in its output queue, with $delay_n(i, t+1) = delay_n(i, t) + RTT$ if the i-th packet in the queue is not transmitted during the tth RTT. When the user receives a new packet from the upper

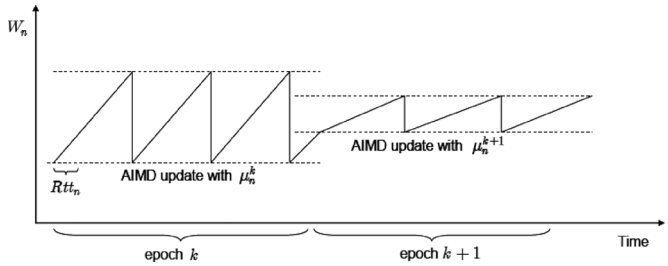

Fig. 1. Congestion window size over time with different updating policies per chunk.

layer, the delay of this packet equals 0, and is increased each RTT until the packet is transmitted successfully, or deleted by the sender. In fact, before transmitting a packet, the user verifies if $delay_n(i,t) < D_n$, where D_n is the deadline delay of the packet. If not, it drops the packet. The observed information o_n is available to the sender through transmission acknowledgments (ACK) built into the protocol.

B. Two-Level Congestion Control Adaptation

A TCP-like window-based congestion control scheme increases the congestion window after successful transmission of several packets, and decreases the congestion window upon the detection of a packet loss event. A general description regarding the congestion control window size variation is:

$$w_n \leftarrow \begin{cases} w_n + f(w_n), & \text{if } o_n = success; \\ w_n - g(w_n)w_n, & \text{if } o_n = fail. \end{cases} \quad (1)$$

Let us define $\mu_n(w_n) = [f(w_n), g(w_n)] \in \mathcal{A}$ as the updating policy that specifies the two congestion window size variation functions (we refer to $f(w_n)$ as the increasing function and $g(w_n)$ as the decreasing function), where $\mathcal{A}$ represents the set of all updating policies. Some existing examples of updating policies can be found in [3] and [4].

Unlike the existing TCP congestion control that fixes the congestion window updating policy without considering applications' characteristics, the proposed Learning-TCP uses a two-level adaptation to update the congestion window. We define a congestion control chunk as a time period, denoted by T, for user n to periodically change its congestion window updating policy. In fact, we allow a sender to update its policy at the beginning of each chunk, which it cannot change until the next chunk (see Fig. 1). Indeed, this paper focuses on how to optimally determine the updating policy, at each chunk, in order to improve the quality of multimedia applications.

C. Expected Multimedia Quality per Chunk

In this section, we discuss the objective of the proposed media-aware congestion control. Denote the application parameters as $\phi_n^k = (R_n^k, D_n^k, A_n^k)$ for user n in the kth chunk, where R_n^k represents the source rate of the multimedia application. The source rate is the average number of packets that arrives at the transmission buffer per second. For example, in a VoIP call, the source rate can be controlled and adapted to the network environment, since there are usually rate control modules implemented in VoIP software. To accurately capture the characteristics of the video packets, we adopt the sophisticated video traffic model proposed in [13], which takes into account the fact that video packets have different delay deadlines and distortion impacts. We further assume an additive distortion reduction function for multimedia applications as in [14], [15] and [13], and A_n^k is the additive distortion reduction per packet in chunk k. A_n^k can be thought of as the media quality improvement of each packet. In fact, the distortion impact A_n^k represents the amount by which the multimedia distortion is reduced if one packet from the chunk k is received at the decoder side. The distortion impact computation is similar to [16]. Note that the developed framework is also applicable in the case when the packets within one chunk have different distortion impact, where A_n^k is a set of distortion impacts of packets transmitted during the chunk k.

The following equation depicts the expected distortion reduction per RTT for an end user n:

$$E[\Omega_n^{t,k}(w_n^t, \phi_n^k)] = A_n^k(1 - p_n^k(w_n^t)) \sum_{i=1}^{\min\{w_n^t, buf_n\}} I(delay_n(i,t) \le D_n^k), \quad (2)$$

where buf_n represents the number of packet in the buffer of the user n, w_n^t is the congestion window of the user n at the tth RTT and $p_n^k(w_n^t)$ is the probability that a packet will be rejected. Note that a packet rejected because of a congestion event $p_n^k(w_n^t)$ will be retransmitted by the sender. However, the user will not retransmit a packet deleted because of the delay constraint. The average distortion reduction in the kth chunk is expressed as follows:

$$E[\Lambda_n^k(\mu_n^k, \phi_n^k)] = \frac{1}{T}\sum_{t=1}^{T} E[\Omega_n^{t,k}(w_n^t, \phi_n^k)]. \quad (3)$$

Specifically, a POMDP framework allows users to evaluate the network congestion without perfect knowledge of the overall system state. For each chunk, our proposed Learning-TCP allows an user n to select the optimal updating policy $\mu_n^{opt,k}$ that maximizes the expected distortion reduction in a chunk k, given application parameters ϕ_n^k. Thus, the proposed algorithm performs the following optimization:

$$\mu_n^{opt,k} = \arg\max_{\mu_n^k} \sum_{k=1}^{\infty} \gamma^k E[\Lambda_n^k(\mu_n^k, \phi_n^k)], \quad (4)$$

where γ is a discount factor. Note that when the application has no delay deadline, i.e., $D_n^k = \infty$, the objective function in (4) is equivalent to maximizing the exponential moving average throughput in the chunk.

During periods of severe congestion, our algorithm may not be TCP-friendly, and therefore penalizes other TCP flows. We describe, in the next section, how we adapt our algorithm to be quality-centric and TCP-friendly.

D. TCP-Friendliness

TCP is not well-suited for emerging multimedia applications because it ignores QoS requirements of the multimedia traffic. To address this issue, some approaches were proposed using end-to-end congestion control schemes [17]. Since TCP is widely used for traffic transport over the Internet, new congestion control schemes should be TCP-Friendly. Therefore, TCP-Friendly congestion control for multimedia has recently become an active research topic (see [3] and [18]). TCP-Friendliness requires that the average throughput of applications using new congestion control schemes does not exceed

that of traditional TCP-transported applications under the same circumstances (see [19]). Therefore, we examine the competitive behaviors between TCP and our proposed congestion control algorithm. It is straightforward and somehow intuitive that updating policies in $\mathcal{A}$ are not necessarily TCP-Friendly (for example, $f(w) = w$ and $g(w) = 1$). However, there exists a non-empty subset of $\mathcal{A}$, whose policies do not violate the TCP-friendliness rule.

It is well known that the TCP congestion control strategy increases by one or decreases by half the congestion window. Let us consider a scenario with a link having a capacity of r packets per RTT, shared between two flows, one TCP-transported and the other using our media-aware congestion control algorithm. The following proposition states that our Learning-TCP algorithm can be TCP-Friendly.

Proposition 1: For all updating policies μ chosen from the set $\mathcal{A}_{fr} = \{\mu(w) = [f(w), g(w)] | f(w) = \frac{3g(w)}{2-g(w)}\}$, the proposed Learning-TCP algorithm is TCP-Friendly.

Proof: See Appendix A.

III. POMDP Framework for Media-Aware Congestion Control

In the proposed framework, users have only partial knowledge about the congestion status of bottleneck links. We define the congestion factor C_g, which represents the impact of all users on the congestion status at bottleneck links. The congestion factor can be seen as a congestion level or occupation level of the bottleneck link. We denote the subscript g in order to differentiate this metric with a cost C. $\mathcal{C}$ represents the set of all possible congestion factors. Since the user cannot observe the traffic generated by other users and transmitted over the bottleneck links, the congestion factor is estimated based on history of observations and actions. Therefore, we formulate the problem with a POMDP framework. The objective function to optimize can be rewritten as follows:

$$U_n = \sum_k \gamma^k \sum_{t=1}^{T} A_n^k (1 - p_n^k(w_n^t)) \cdot \sum_{i=1}^{\min\{w_n^t, buf_n\}} I(delay_n(i,t) \leq D_n^k). \quad (5)$$

We denote by

$$u_n(X_n^k, \mu_n^k) = \sum_{t=1}^{T} A_n^k (1 - p_n^k(w_n^t)) \cdot \sum_{i=1}^{\min\{w_n^t, buf_n\}} I(delay_n(i,t) \leq D_n^k)$$

the instantaneous reward of the end user n at the kth chunk. Note that the end user tries to maximize the number of packets successfully transmitted before their delay deadlines.

A. POMDP-Based Congestion Control

Based on (5), we define a POMDP-based congestion control of user n as follows:

1) Action: The policy of a user is a set of actions that the user selects at every chunk. The action of a user at a given epoch k is to choose the appropriate congestion window updating policy. We denote by $\mu_n = \mu_n^1, \mu_n^2, \ldots$ the policy of the network user where μ_n^k is its action at the kth chunk. Note that μ_n^k represents the congestion window updating policy at chunk k.

2) State: The state is defined as $X_n^k = \{C_g, \phi_n^k\} \in \mathcal{X}_n$. The application parameters ϕ_n^k are known by user n. However, the congestion factor $C_g \in \mathcal{C}$, which is impacted by the overall traffic transmitted over the bottleneck link, cannot be directly observed by users. The user n has to infer the congestion factor based on the observed information and feedback.

At each time slot, the system has a congestion factor C_g. The user takes an action μ_n, which induces the congestion factor to transit to the state C_g' with probability $T(C_g', \mu_n, C_g)$. Having the congestion factor C_g', the user observes o_n with probability $O(o_n, C_g', \mu_n)$. The belief about the congestion factor is defined as a function $b_n^k : \mathcal{C} \to [0,1]$. The function $b_n^k(.)$ represents the probability distribution of the congestion factor at the kth chunk for user n. Denote the chosen congestion factor (i.e., inferred by the end user as the most likely of all possible congestion factors) at the kth chunk by C_g^k. The belief distribution of the congestion factor $b_n^k(C_g)$ is updated as follows:

$$b_n^k(C_g') = \frac{Prob(o_n | C_g', \mu_n^k, b) Pr(C_g' | \mu_n^k, b)}{Prob(o_n | \mu_n^k, b)}; = \frac{O(o_n, C_g', \mu_n^k) \sum_{C_g \in \mathcal{C}} T(C_g', \mu_n^k, C_g) b_n^{k-1}(C_g)}{Prob(o_n | \mu_n^k, b)}. \quad (6)$$

The denominator, $Prob(o_n | \mu_n, b)$, can be treated as a normalizing factor, independent of C_g' that causes b to sum to 1.

The probability $p_n^k(w_n)$ represents the average packet loss rate in the kth chunk when the congestion window size is w_n, which can be calculated as follows:

$$p_n^k(w_n) = \sum_{C_g \in \mathcal{C}} Prob(C_g \geq \widetilde{C}_g | w_n) b_n^k(C_g), \quad (7)$$

where $\widetilde{C}_g$ is the congestion level at the bottleneck link, which is not observable by end users. However, the average packet loss rate itself is observable by users, given a certain congestion window w_n.

3) Utility: The goal of the user n is to maximize the discounted reward defined in (5). A policy $\mu_n^{opt} = \{\mu_n^{opt,1}, \mu_n^{opt,2}, \ldots\}$ that maximizes U_n is called optimal policy and specifies for each chunk k, the optimal updating policy $\mu_n^{opt,k}$ to use. The optimal value function U_n^k satisfies the following Bellman equation:

$$U_n^k(C_g) = \max_{\mu_n^k \in \mathcal{A}} \{u_n(X_n^k, \mu_n^k) + \gamma \sum_{C_g' \in \mathcal{C}} T(C_g', \mu_n^k, C_g) U_n^{k+1}(C_g')\}. \quad (8)$$

The optimal policy at the kth chunk is expressed as follows:

$$\mu_n^{opt,k} = \arg\max_{\mu_n^k \in \mathcal{A}} \{u_n(X_n^k, \mu_n^k) + \gamma \sum_{C_g' \in \mathcal{C}} T(C_g', \mu_n^k, C_g) U_n^{k+1}(C_g')\}. \quad (9)$$

We prove, in the next section, the existence of the optimal stationary policy, and we show how to determine such policy for our POMDP problem.

B. Existence of Optimal Stationary Policies

To limit the computation complexity and propose a low-cost implementation for determining the optimal solution for POMDP-based problems, we restrict our attention to the set of

stationary policies. A policy is stationary if this policy does not depend on the chunk k. Note that we formulate our problem as an infinite horizon POMDP with expected discounted reward.

The belief set is continuous, which may lead to an explosion of the solution size and the computation complexity. Therefore, we transform the belief set to a discrete set. We use an aggregation function that maps the belief states into a discrete set of beliefs. An example of aggregation function is presented in Section IV. Moreover, for each belief, we assume that there is a finite set of actions $\mathcal{A}$. Under these assumptions, Theorem 6.2.10 of [20] can be applied and we can assume the existence of the optimal stationary policy for our POMDP problem. Therefore, we restrict our problem to the set of stationary policies. We can now omit the chunk index k, as the optimal stationary policy depends only on ϕ and C_g. The goal of this POMDP problem is therefore to find a sequence of updating policies μ_n that maximizes the expected reward. For each belief, the value function can be formulated as follows:

$$U_n(C_g) = \max_{\mu_n \in \mathcal{A}} \{u_n(X_n, \mu_n) + \gamma \sum_{C'_g \in \mathcal{C}} T(C'_g, \mu_n^k, C_g) U_n(C'_g)\}. \quad (10)$$

Specifically, a powerful result of [21] and [22] says that the optimal value function for our POMDP problem is Piecewise Linear and Convex (PWLC) in the belief. Then, every value function can be represented by a set of hyper-planes denoted Υ-vectors, Γ_k, where $U_n(C_g) = \max_{\Upsilon \in \Gamma_k} b(C_g)\Upsilon$. Γ_k is updated using the value iteration algorithm through the following sequence of operations:

$$\Gamma_{k+1}^{\mu,o_n} \leftarrow \Upsilon_\mu^{o_n}(X_n) = \frac{u_n(X_n,\mu)}{|O_n|} + \gamma \sum_{X' \in \mathcal{X}} T(X_n, \mu, X') O(o_n, C'_g, \mu) \Upsilon(X'), \forall \Upsilon \in \Gamma_k, \quad (11)$$

$$\Gamma_{k+1}^{\mu} = \oplus_{o_n} \Gamma_{k+1}^{\mu,o_n}; \quad (12)$$

$$\Gamma_{k+1} = \cup_{\mu \in \mathcal{A}} \Gamma_{k+1}^{\mu}. \quad (13)$$

Note that each Υ-vector is associated with an action that defines the best updating policies for the previous $(k-1)$ chunks. The kth horizon value function can be expressed as follows:

$$U(C_g) = \max_{\mu_n \in \mathcal{A}} \left[u_n(X_n^k, \mu_n) + \gamma \sum_{o_n} \max_{\Upsilon \in \Gamma_k^{\mu_n, o}} \sum_{C'_g \in \mathcal{C}} T(C'_g | C_g) O(o_n, C'_g, \mu) \Upsilon \right]. \quad (14)$$

Many algorithms have been proposed to implement solutions for POMDP problems by manipulating Υ-vectors using a combination of set projection and pruning operations (see [21], [5] and [23]).

The main difficulty of POMDP-based optimization is the prohibitively high computational complexity and the assumption that statistics, such as the state transition probability are known, which may be not true in practice. To overcome this obstacle, we propose an online learning method that allows the sender to determine the optimal congestion control policy on-the-fly, with a low computational complexity.

IV. Online Learning

Solving a POMDP is an extremely difficult computational problem. In this section, we show how a value function can be updated on-the-fly, with a low computation complexity, in order to solve the POMDP problem described in the previous section. In the proposed learning model, a user maintains the state-value function $Q(\mu_n, \phi, C_g)$ as a lookup table, which determines the optimal policy in the current slot. In fact, the state-value function $Q(\mu_n, \phi, C_g)$ is updated as follows:

$$Q(\mu_n^{k-1}, \phi^{k-1}, C_g^{k-1}) \leftarrow \beta_k Q(\mu_n^{k-1}, \phi^{k-1}, C_g^{k-1}) + (1-\beta_k)(U_n + \gamma Q(\mu_n^k, \phi^k, C_g^k)), \quad (15)$$

where β_k is a learning rate factor satisfying $\sum_{k=1}^{\infty} \beta_k = \infty, \sum_{k=1}^{\infty} (\beta_k)^2 < \infty$, e.g., $\beta_k = \frac{1}{k}$. At the chunk k, the user gets application parameters ϕ^k and estimates the congestion factor C_g. Then he chooses the policy μ_n^k that maximizes $Q(\mu_n^k, \phi^k, C_g)$.

The large state space $\mathcal{X}_n$ may prohibit an efficient learning solution and may increase the complexity and the convergence time of the algorithm. We propose to adopt an effective state aggregation mechanism in order to reduce the complexity and the convergence time of the learning algorithm. As an example of the aggregation function, we may quantize the congestion factor to the nearest integer.

A. Adaptive State Aggregation

We propose to use an aggregation function that maps the congestion factor space $\mathcal{C}$ into a discrete space, as we have assumed in Section III-B. This function aggregates the adjacent congestion factors $C'_g \in \tau \subset \mathcal{C}$ into a representative average congestion factor value C_g, where τ is a subset of the set of congestion factors $\mathcal{C}$. For example, $\mathcal{C} = [0,1]$ and $\tau = [\frac{i}{n}, \frac{i+1}{n}]$, $0 \le i < n$. In this paper, we propose an adaptive state aggregation method that iteratively adapts the aggregation function. Let $\Delta(C_g, U_n^k, \delta)$ represents the adaptive aggregation function, defined as follows:

$$\Delta(C_g, U_n^k, \delta) = C_g^n = \frac{C^L + C^H}{2}, \quad (16)$$

where $C^L = invU_n^k(U^{\min} + (l-1)\delta)$, $C^H = invU_n^k(U^{\min} + l\delta)$, and $(l-1)\delta \le U_n^k(C_g) - U^{\min} < l\delta$. Note that $invU_n^k$ represents the inverse function of $U_n^k(C_g)$, $U^{\min}$ denotes the minimum value of the expected utility of the user starting from the previous chunk, and δ is referred to the utility spacing that determines the aggregation function from the expected utility-to-go domain.

B. Structural Properties

In this section, we develop some structural properties of the optimal policy and corresponding value function, based on which we will then discuss approximation results of the value function. This approximation allows us to represent the value function in a compact manner. Importantly, we are able to control the computational complexity and achievable performance by using different predetermined approximation error thresholds δ.

We propose, in this section, a low-complexity online learning algorithm based on an extension of the TD-λ Algorithm [24],

described in Algorithm 1. The proposed learning method is greatly impacted by the utility spacing δ. The number of states in a chunk depends on the aggregation function $\Delta(C_g, U_n^k, \delta)$ and the size of the average congestion set in the kth chunk is $\lceil \frac{U^{k,max} - U^{k,\min}}{\delta} \rceil + 1$. We have the following theorem about the learning error:

Theorem 1: After k chunks, the learning error of the state-value iteration is bounded by: $|\bar{U}_n(C_g) - U_n(C_g)| \leq \frac{\delta(1-\prod_{i=1}^{k}\beta_i)}{1-\gamma}$, where $\bar{U}_n$ is the value function of the optimal policy.

Proof: See Appendix B.

Algorithm 1: Online learning algorithm for POMDP-based congestion control

Initialize $Q(\mu_n^k, \phi_n^k, C_g) = 0$ for all possible application parameters, congestion factor and updating policy;

Initialize ϕ, μ_n and C_g;

$U_n = 0$;

while true **do**

$\phi^{prev} = \phi$;

$\mu_n^{prev} = \mu_n$;

$C_g^{prev} = C_g$;

Get the new application parameters ϕ;

Select the policy and congestion factor such as: $(\mu_n^k, C_g) = \arg\max_{\mu_n, C_g} Q(\mu_n, \phi, C_g) b_n(C_g)$ with probability $(1-\epsilon)$, else choose a random policy and congestion factor;

$Q(\mu_n^{prev}, \phi^{prev}, C_g^{prev}) \leftarrow \beta_k Q(\mu_n^{prev}, \phi^{prev}, C_g^{prev}) + (1-\beta_k)(U_n + \gamma Q(\mu_n, \phi, C_g))$;

$U_n = 0$;

for $t = 1 \rightarrow T$ **do**

Transmit packets using the updating policy μ_n and the congestion factor C_g;

Update the congestion window based on (1);

$U_n = U_n + A_n^k \times recPkt$, where $recPkt$ is the number of packets received before their delay deadlines.

end for

Update the beliefs based on (6);

end while

At the beginning of chunk k, the user receives application parameters ϕ_n^k from the upper layer, and selects the updating policy and the congestion factor that maximize its state-value function. Then, the user transmits its packets during the chunk using the chosen policy. At the end of the chunk, the user updates the state-value function based on observation. The following lemma proves the convergence of the proposed algorithm.

Lemma 1: The proposed learning algorithm converges to the optimal value function.

Proof: See Appendix C.

C. Implementation and Complexity

Although value iteration algorithms give an exact solution of POMDP optimization problems, those algorithms require a time and space complexity that may be prohibitively expensive. In fact, to better understand the complexity of exactly solving the POMDP problem, let $|\Gamma_k|$ be the number of Υ-vectors in the kth chunk. In the worst case, the Υ-vectors size in the $(k+1)$th chunk is $|\mathcal{A}| \times |\Gamma_k|$ (see [25]), and the running time will be $|\mathcal{X}_n|^2 \times |\mathcal{A}| \times |\Gamma_k|$. It also requires solving a number of linear programs for pruning vectors.

Interestingly, the proposed algorithm has a state space of $|\mathcal{A}| \times |\mathcal{C}| \times |\Phi|$, where Φ is the set of application parameters. Moreover, our Learning-TCP algorithm has a polynomial time complexity.

Note that introducing the chunk reduces not only the state space, but also the complexity of solving the POMDP problem. Furthermore, fixing users' policies for some time (here we have the chunk time) is important from networking point of view as well. Since the user is dealing with other network users (although they are aggregate in the congestion factor), it may take time for this multiuser interaction to converge first. Note that the performance of a user's policy depends highly on the accuracy of the other users impact. If we fix the policy for some time, we know how to evaluate the resulting utility expectation and then react to it. Changing the policy every RTT may make such expectation inaccurate.

The proposed algorithm is implemented only at the transmitter side and is transparent for the receiver. We do not even require any change at routers. Furthermore, as we have proved that Learning-TCP is TCP-Friendly, any other congestion control algorithm can be implemented in parallel. For first chunks, the Learning-TCP algorithm may give suboptimal performance. However, a near-optimal result can be obtained after a sufficient number of chunks. Interestingly, we can significantly speed up the learning and avoid this problem if the state-value functions are initialized with values obtained the last time Learning-TCP was used.

V. Simulations

In this section, we present some simulation results using MATLAB-based simulations of our proposed Learning-TCP algorithm. We consider that multimedia users are transmitting video sequences at a variable bit rate of $\mathcal{R} = \{1, 1.25, 1.5, \ldots, 5.75, 6\}$ Mbps. We assume that packets can tolerate a delay of $\mathcal{D} = \{133, 266, 400, \ldots, 800\}$ ms, and we set the packet length to 1024 Bytes. Moreover, we assume that each frame has an additive distortion per packet in the set $\mathcal{A}_{distor} = \{0.05, 0.06, \ldots, 0.16\}$. We consider also a set of policies $\mathcal{A}$ composed of IIAD and SQRT policies defined as follows:

$$\text{IIAD}: f(w) = \frac{3\lambda}{2w - \lambda} \text{ and } g(w) = \frac{\lambda}{w}; \quad (17)$$

$$\text{SQRT}: f(w) = \frac{3\lambda}{2\sqrt{w+1} - \lambda} \text{ and } g(w) = \frac{\lambda}{\sqrt{w+1}}; \quad (18)$$

where $\lambda \in \{0.1, 0.2, \ldots, 0.9\}$. We consider the set of average congestion factors $\mathcal{C} = \{1, 2, .., 50\}$, and we set γ to 0.1.

A. TCP-Fairness

We focus on the fairness of our proposed Learning-TCP. Fig. 2 shows how the proposed algorithm interacts with TCP transported flows depending on QoS parameters chosen from the set $\Phi = \mathcal{R} \times \mathcal{D} \times \mathcal{A}_{distor}$. In order to study this effect, we simulate 10 connections: 5 with TCP and 5 connections

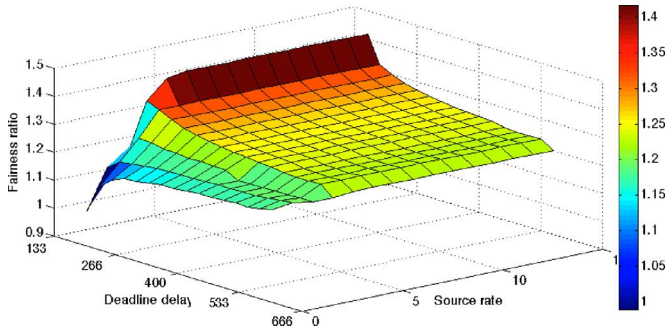

Fig. 2. Fairness ratio of LEARNING-TCP for different source rates and delay deadlines.

using the Learning-TCP algorithm, within different QoS requirements and application parameters. We illustrate, in Fig. 2, the fairness ratio depending on the delay deadline and source rate. The fairness ratio (see [4] and [12]) is defined by the ratio between the total throughput of Learning-TCP connections and total throughput of TCP connections. The closer the fairness ratio is to 1, the friendlier will the congestion control be to other TCP flows. We observe that Learning-TCP has a fairness ratio close to 1 except with hard deadline delay and high source rate. In fact, as we can see in Fig. 2, when the delay deadline is lower than 300 ms and the source rate is higher than 4 Mbps, the fairness ratio is between 1.2 and 1.45. Indeed, with hard deadline delays and high source rates, the user needs higher throughput in order to satisfy its QoS requirements.

B. Learning-TCP Algorithms and Fixed-Policy Algorithms

Now, we investigate the interactions between Learning-TCP and other multimedia congestion control algorithms. We consider a bottleneck link of capacity 100 Mbps shared between 10 users (one Learning-TCP; one TCP and 8 users using Binomial congestion control) as described in Table II. We simulate a video transmission application during 350 time slots, and we assume that users receive a new set of application parameters every chunk, $\mathrm{T} = 5$ s. In order to illustrate the impact of the delay on the congestion control algorithms, we assume that the deadline delay is 133 ms in the first chunk, and that it increases by 133 ms every chunk. A real use-case of these simulation settings is a streaming application, where the user may change the required quality at each chunk. For example, let us consider a congested network, the user decreases at the end of each chunk the required quality of the streaming, and increases the deadline delay, thereby decreases the packet loss probability. We observe, in Fig. 3, that the Learning-TCP uses different policies for each delay deadline. For hard delay deadlines, we observe that the throughput of the Learning-TCP user is higher than the throughput of other users. Fig. 4 illustrates the throughput of TCP user and Fig. 5 illustrates the throughput of Binomial congestion control users. Binomial-CC users obtain the highest average throughput (9.2 Mbps Versus 7.65 Mbps for TCP and 8.36 Mbps for Learning TCP). In fact, as we can see in Fig. 3, the Learning-TCP gives the highest throughput for hard delay deadlines. However, it is still TCP-friendlier in the average. Finally, Fig. 6 illustrates the variation of the congestion window size with the number of TCP chunk. We observe that the congestion window updating policy converges after 10 TCP chunks. Interestingly, we show in the next section, how the proposed algorithm gives better video quality when obeying the friendliness rule.

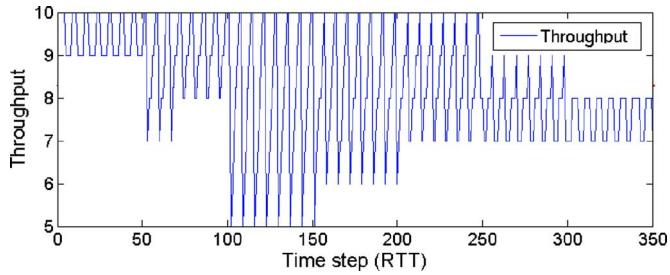

Fig. 3. Throughput of LEARNING-TCP.

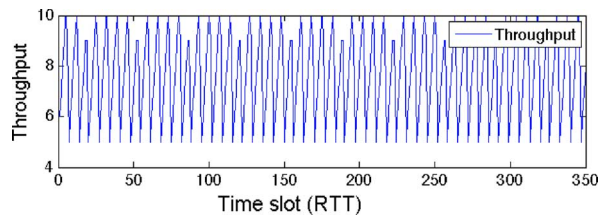

Fig. 4. Throughput of TCP.

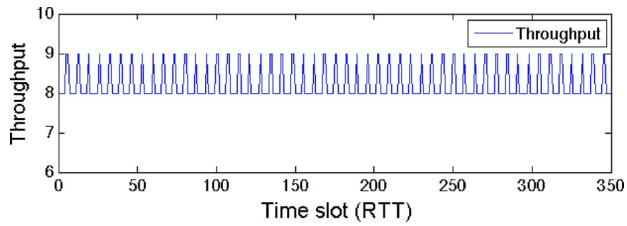

Fig. 5. Throughput of Binomial-CC.

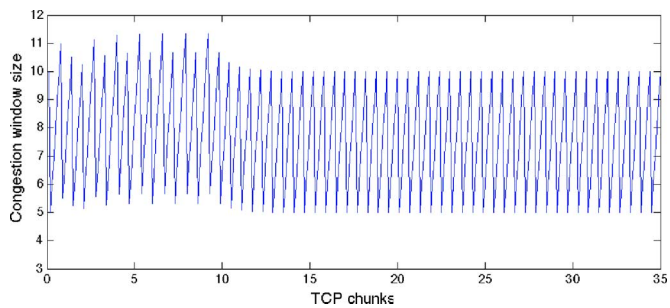

Fig. 6. Congestion window size with the number of TCP chunks.

TABLE II
USERS IN THE NETWORK

	IIAD1	IIAD2	IIAD3	IIAD4	TCP
I	$\frac{0.3}{w-0.1}$	$\frac{0.6}{w-0.2}$	$\frac{0.9}{w-0.3}$	$\frac{1.2}{w-0.4}$	1
D	$\frac{0.2}{w}$	$\frac{0.4}{w}$	$\frac{0.4}{w}$	$\frac{0.8}{w}$	0.5
	SQRT1	SQRT2	SQRT3	SQRT4	LEARNING-TCP
I	$\frac{3}{8\sqrt{w+1}-1}$	$\frac{3}{4\sqrt{w+1}-1}$	$\frac{9}{8\sqrt{w+1}-3}$	$\frac{3}{2\sqrt{w+1}-1}$	f(w)
D	$\frac{0.25}{\sqrt{w+1}}$	$\frac{0.5}{\sqrt{w+1}}$	$\frac{0.75}{\sqrt{w+1}}$	$\frac{1}{\sqrt{w+1}}$	g(w)

C. Performances of Learning-TCP Against Others Multimedia Congestion Control Algorithms

In order to evaluate the video quality (measured through the average Peak Signal to Noise Ratio (PSNR), in decibels) using

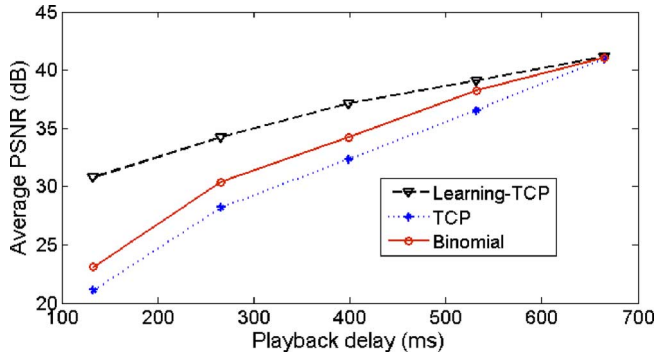

Fig. 7. Average received video quality using different congestion control for multimedia transmission.

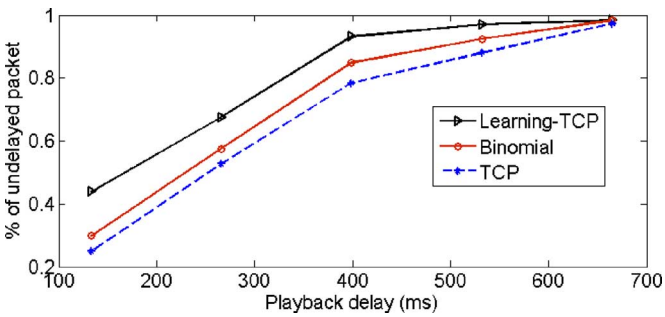

Fig. 8. The percentage of packets delivered before their delay deadlines.

different congestion control algorithms, we simulate the transmission of a video sequences with length of 50 s (CIF resolution, 50 Hz frame-rate) and compressed by an H.264/AVC codec (any codec can be used, we used this one just for illustrative purposes). We assume that users receive different values of source rate and additive distortion per packet at every chunk. The delay deadline varies between 133 ms and 800 ms. Fig. 7 illustrates the video quality obtained with different congestion control algorithms. We observe that the Learning-TCP leads to better video quality. Therefore, our proposed approach outperforms others, especially for real-time applications with hard deadline delay such as video-conferencing applications for example. In fact, as illustrated in Fig. 8, Learning-TCP users obtain the highest percentage of packets delivered before their delay deadline. Indeed, our algorithm is able to optimize the congestion window by considering the distortion impact, delay deadline and the source rate.

VI. Conclusion

We have formulated, in this paper, the media-aware congestion control problem as a POMDP that explicitly takes into consideration the multimedia streaming characteristics such as delay constraints and distortion impacts. We have considered a set of generic TCP-friendly congestion window updating functions. The optimal policy allows the sender to optimize the congestion window updating policy that maximizes the long term expected quality of multimedia applications. We have proposed an online learning method to solve the Learning-TCP on the fly. Simulation results show that the proposed congestion control algorithm outperforms conventional TCP-friendly congestion control schemes in terms of quality, especially for real-time applications with hard delay deadlines. Moreover, the proposed Learning-TCP algorithm is implemented only at the sender side, and is transparent to routers and receivers. Hence, our proposed method can easily and immediately be adopted for media-aware TCP transmission without the need of standardization and it can fairly and smoothly co-exist with existing TCP-solutions for media streaming, while providing significant performance improvements for the adopting devices.

Appendix A
Proof of Proposition 1

The proof of this proposition is a generalization of the proof of [3] and [26] made for AIMD(α, β). We extend this result for a general updating policies $f(w), g(w) : \mathcal{R} \rightarrow \mathcal{R}$.

Denote by w_{L-TCP} and w_{TCP} the congestion windows of the Learning-TCP transported flow and the TCP-Transported flow respectively. Assume that both flows have the same RTT and MSS. The effect due to different RTT and MSS is beyond the scope of this paper and is an issue in our future work. On one hand, when $w_{L-TCP} + w_{TCP} < r$, the link is in the underload region and thus, the congestion windows w_{L-TCP} and w_{TCP} evolves as follows:

$$\begin{aligned} w_{L-TCP}(t+\Delta t) &= w_{L-TCP}(t) + f(w_{L-TCP}(t))\Delta t \\ w_{TCP}(t+\Delta t) &= w_{TCP}(t) + \Delta t. \end{aligned}$$

On the other hand, when $w_{L-TCP} + w_{TCP} \geq r$, the link is overloaded and congestion occurs. We assume that both flows receive the congestion signal once congestion occurs and we denote t_i the ith time that the link is congested. Both flows decrease simultaneously their window based on the following expression:

$$\begin{aligned} w_{L-TCP}(t_i) + w_{TCP}(t_i) &= r \\ w_{L-TCP}(t_i^+) &= w_{L-TCP}(t_i) \\ &\quad - g(w_{L-TCP}(t_i))w_{L-TCP}(t_i) \\ w_{TCP}(t_i^+) &= \frac{1}{2} w_{TCP}(t_i). \end{aligned}$$

The duration between t_i and t_{i+1} is referred as the i–th cycle during which both flows increase their window. Therefore, we have:

$$\begin{aligned} &w_{L-TCP}(t_{i+1}) - w_{L-TCP}(t_i) \\ &\quad = -\frac{2g(w_{L-TCP}(t_i)) + f(w_{L-TCP})}{2(f(w_{L-TCP})+1))} w_{L-TCP}(t_i) \\ &\qquad + \frac{r f(w_{L-TCP})}{2(f(w_{L-TCP})+1))}. \end{aligned}$$

Thus, independent of the initial values of w_{L-TCP} and w_{TCP}, after a sufficient number of cycles, the congestion windows of these two flows in the overloaded region converge to:

$$\begin{aligned} w_{L-TCP}(th) &= \frac{f(w_{L-TCP})r}{2g(w_{L-TCP}) + f(w_{L-TCP})}, \\ w_{TCP}(th) &= \frac{2g(w_{L-TCP})r}{2g(w_{L-TCP}) + f(w_{L-TCP})}. \end{aligned}$$

Therefore, in the steady state, w_{L-TCP} and w_{TCP} increase and decrease periodically. Their average throughput in steady state are expressed by the following:

$$\bar{w}_{L-TCP} = \frac{(2 - g(w_{L-TCP}))f(w_{L-TCP})r}{4g(w_{L-TCP}) + 2f(w_{L-TCP})},$$
$$\bar{w}_{TCP} = \frac{3g(w_{L-TCP})r}{4g(w_{L-TCP}) + 2f(w_{L-TCP})}.$$

To guarantee the fairness between the flows, the necessary and sufficient condition is:

$$f(w) = \frac{3g(w)}{2 - g(w)}. \tag{19}$$

Appendix B
Proof of Theorem 1

Similar to [27], it can be shown that the convergence of the online learning algorithm is equivalent to the convergence of the following O.D.E.:

$$U_n = M(U_n), \tag{20}$$

where the mapping M is defined by:

$$M(U_n) = \sum_{t=1}^{T} u_n^t(C_g, \phi_n^k) + \gamma U_n.$$

A successive approximation iteration on a vector U_n simply replaces U_n with $M(U_n)$. The successive approximation method for the solution of (20) starts with a vector U_n chosen as a convex function, and sequentially computes $M(U_n), M^2(U_n), \ldots$. Since the optimal state value function is PWLC in the average congestion factor and $\gamma \in [0, 1]$, it follows from Blackwell's Sufficient Conditions that M is a sup-norm contraction mapping with modulus γ. Hence, we have:

$$\lim_{k \to \infty} M^k(U_n) = U_n^*, \tag{21}$$

where M^k is the composition of the mapping M with itself k times. The convergence rate in (21) is geometric at a rate γ. The rate of convergence can be improved using some error bounds based on the residual difference of $M(U_n)$ and U_n. Therefore, for all states i, the solution U_n^* of (20) satisfies:

$$|M(U_n)(i) - U_n^*(i)| \leq \frac{\max_{i,j}[M(U_n)(i) - U_n(j)]}{1 - \gamma} \leq \frac{\delta}{1 - \gamma}. \tag{22}$$

Thus, after a chunk, the learning error is bounded by the following upper bound:

$$|\bar{U}_n(w) - U_n(w)| \leq (1 - \beta_1)\frac{\delta}{1 - \gamma}. \tag{23}$$

After 2 chunks, the learning error is bounded by:

$$|\bar{U}_n(w) - U_n(w)| \leq \beta_2(1 - \beta_1)\frac{\delta}{1 - \gamma} + (1 - \beta_2)\frac{\delta}{1 - \gamma} \leq (1 - \beta_1\beta_2)\frac{\delta}{1 - \gamma}.$$

Finally, we obtain by induction on the number of chunks that after k chunks, the learning error is bounded as follows: $|\bar{U}_n(w) - U_n(w)| \leq \frac{\delta(1 - \prod_{i=1}^{k} \beta_i)}{1 - \gamma}$ ■

Appendix C
Proof of Lemma 1

The proof of this lemma follows from the Theorem 1 of [28]. In fact, Sarsa algorithm converges to the optimal values function whenever the following assumptions hold:

1) The state space and the action space are finite,
2) β_k satisfies $\sum_{k=1}^{\infty} \beta_k = \infty$, $\sum_{k=1}^{\infty} (\beta_k)^2 < \infty$, e.g., $\beta_k = \frac{1}{k}$,
3) The reward function is bounded.

It is straightforward that the previous assumptions hold for our problem, and therefore, the Algorithm 1 converges to the optimal values function. ■

References

[1] 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, pp. 1–22, May 2008.

[2] A. Balk, M. Gerla, D. Maggiorini, and M. Sanadidi, "Adaptive video streaming: Pre-encoded MPEG-4 with bandwidth scaling," *Comput. Netw.*, vol. 44, pp. 415–439, Mar. 2004.

[3] L. Cai, X. Shen, J. Pan, and J. W. Mark, "Performance analysis of TCP-friendly AIMD algorithm for multimedia applications," *IEEE Trans. Multimedia*, vol. 7, pp. 339–355, Apr. 2005.

[4] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithm," in *Proc. IEEE INFOCOM*, 2001, pp. 631–640.

[5] A. Cassandra, M. Littmann, and N. Zhang, "Incremental pruning: A simple, fast, exact method for partially observable Markov decision process," in *Proc. UAI*, 1997, pp. 54–61.

[6] M. L. Littman, "The Witness Algorithm: Solving partially observable Markov decision processes," Comput. Sci. Dept., Brown Univ., Providence, RI, Tech. Rep. CS-94-40, 1994.

[7] L. Chrisman, "Reinforcement learning with perceptual aliasing: The perceptual distinctions approach," in *Proc. 10th Nat. Conf. Artif. Intell.*, 1992, pp. 183–188.

[8] R. A. McCallum, "First results with utile distinction memory for reinforcement learning," presented at the Mach. Conf., Rochester, NY, USA, 1993.

[9] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Univ. of Cambridge, King's College, Cambridge, U.K., 1989.

[10] D. E. Rumelhart, "Learning internal representations by error backpropagation," *Parallel Distrib. Process.*, pp. 318–362, 1986.

[11] H.-P. Shiang and M. van der Schaar, "A quality-centric TCP-friendly congestion control for multimedia transmission," *IEEE Trans. Multimedia*, vol. 14, no. 3–2, pp. 896–909, 2012.

[12] R. Rejaie, M. Handley, and D. Estrin, "Rap: An end-to-end rate-based congestion control mechanisme for real-time stream in internet," in *Proc. INFOCOM*, 1999, pp. 1337–1345.

[13] F. Fu and M. van der Schaar, "Structural solutions for dynamic scheduling in wireless multimedia transmission," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, p. 22, May 2012.

[14] H. Shiang and M. van der Schaar, "Multi-user video streaming over multi-hop wireless networks: A distributed cross-layer approach based on priority queuing," in *IEEE J. Sel. Areas Commun.*, May 2007, vol. 25, pp. 770–785.

[15] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, pp. 390–404, Apr. 2006.

[16] M. van der Schaar and D. Turaga, "Cross-layer packetization and retransmission strategies for delay-sensitive wireless multimedia transmission," *IEEE Trans. Multimedia*, vol. 9, pp. 185–197, Jan. 2007.

[17] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Trans. Netw.*, vol. 7, pp. 458–472, Aug. 1999.

[18] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "A model based TCP-friendly rate control protocol," in *Proc. NOSSDAV*, 1999.

[19] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *ACM SIGCOMM Comput. Commun. Rev.*, Oct. 2000, vol. 30, pp. 43–56.

[20] M. L. Putterman, *Markov Decision Process Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistique. Hoboken, NJ, USA: Wiley, 2005.
[21] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov decision processes over a finite horizon," *Oper. Res.*, vol. 21, pp. 1071–1088, 1973.
[22] E. Sondik, "The optimal control of partially observable Markov processes over the infinite horizon: Discounted cost," *Oper. Res.*, vol. 26, pp. 282–304, 1978.
[23] N. L. Zhang and W. Zhang, "Speeding up the convergence of value iteration in partially observable Markov decision process," *J. Artif. Intell. Res.*, vol. 14, pp. 29–51, 2001.
[24] R. S. Sutton, "Generalization in reinforcement learning: Successful examples using sparse coarse coding," *Adv. Neural Inf. Process. Syst.*, pp. 1038–1045, 1996.
[25] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPS," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2003, pp. 1025–1030.
[26] K. W. Lee, R. Puri, T. Kim, K. Ramchandran, and V. Bharghavan, "An integrated source coding and congestion control framework for video streaming in the internet," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 747–756.
[27] D. P. Bertsekas and D. A. Castanon, "Adaptive aggregation methods for infinite horizon dynamic programming," *IEEE Trans. Autom. Control*, vol. 34, no. 6, pp. 589–598, 1989.
[28] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvari, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Mach. Learn.*, vol. 38, no. 3, pp. 287–308, 2000.

Oussama Habachi received the Engineering degree in computer sciences from the National School of Computer Sciences, Manouba, Tunisia, in 2008, the M.Sc. degree from the University of Pierre and Marie Curie, Paris, France, in 2009, and the Ph.D. degree in computer science from University of Avignon, Avignon, France, in 2012 .

He is currently a Postdoctoral Fellow at the INRIA, Nice, Sophia Antipolis, France. His research interests include performance evaluation of networks based on game theoretic, optimal control, and queueing models. More informations is available at http://www-sop.inria.fr/members/Oussama.Habachi/

Hsien-Po Shiang received the B.S. and M.S. degrees in electrical engineering from National Taiwan University in 2000 and 2002, respectively, and the Ph.D. degree in electrical engineering afrom the University of California, Los Angeles. In 2009.

During his Ph.D. study, he worked at Intel Corp., Folsom, CA, USA, in 2006, researching overlay network infrastructure over wireless mesh networks. He currently works at Cisco Systems, Inc., San Jose, CA, USA, on real-time multimedia transformation platform. He published several journal papers and conference papers on these topics. His research interests include cross-layer optimizations/adaptations, multimedia communications, and dynamic resource management for delay-sensitive applications.

Dr. Shiang has been selected as one of the eight Ph.D. students chosen for the 2007 Watson Emerging Leaders in Multimedia awarded by IBM Research, NY.

Mihaela van der Schaar (F'10) is the Chancellor's Professor of Electrical Engineering at the University of California, Los Angeles, USA. Her research interests include engineering economics and game theory, dynamic multi-user networks and system designs, online learning, multimedia networking, communication, processing, and systems, and multimedia stream mining. She holds 33 granted U.S. patents.

Dr. van der Schaar is a Distinguished Lecturer of the Communications Society for 2011–2012, the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA and a member of the Editorial Board of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. She received an NSF CAREER Award in 2004, the Best Paper Award from the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2005, the Okawa Foundation Award in 2006, the IBM Faculty Award in 2005, 2007, and 2008, the Most Cited Paper Award from *EURASIP: Image Communications Journal* in 2006, the Gamenets Conference Best Paper Award in 2011, and the 2011 IEEE Circuits and Systems Society Darlington Award Best Paper Award. She received three ISO awards for her contributions to the MPEG video compression and streaming international standardization activities. More information about her research is available at http://medianetlab.ee.ucla.edu/

Yezekael Hayel received the M.Sc. degree in computer science and applied mathematics from the University of Rennes 1, France, in 2002 and the Ph.D. degree in computer science from the University of Rennes 1 and INRIA in 2005.

Since 2006, he has been an Assistant Professor at the University of Avignon, Avignon, France. His research interests include performance evaluation of networks based on game theoretic and queueing models. His research focuses on applications in communication networks, such as wireless flexible networks, bio-inspired and self-organizing networks, economics models of the Internet, and yield management. Since he has joined the networking group of the LIA/CERI, he has participated in several projects.

Dr. Hayel was also involved in workshops and conference organization, such as WNC3 2008, GameComm 2009, and Bionetics 2009. He participates in several national (ANR) and international projects, such as the European and cefipra, with industrial companies like Orange Labs, Alcatel-Lucent, IBM, and academic partners like Supélec, CNRS, and UCLA. He has been invited to give seminal talks in institutions like INRIA, Supélec, UAM (Mexico), ALU (Shanghai). More information is available at http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/hayel/acceuil.htm