MOS-Based Congestion Control for Conversational Services in Wireless Environments

Oussama Habachi, Yusuo Hu, Mihaela van der Schaar, Yezekael Hayel, and Feng Wu

Abstract—**Nowadays, multimedia applications and specifically streaming systems over wireless networks use the TCP transport protocol. Indeed, TCP can deal with practical issues such as firewalls and also deploys built-in retransmissions and congestion control mechanisms. We propose in this paper a Quality-centric Mean Opinion Score (MOS) based congestion control that determines an optimal congestion window updating policy for multimedia transmission. Unlike the standard congestion control algorithms, our approach defines a new Additive Increase Multiplicative Decrease (AIMD) algorithm given the multimedia application and the transmission characteristics. In order to get the optimal congestion policy in practice, the sender requires complete statistical knowledge of both multimedia traffic and the network environment, which may not be available in wireless systems. Hence, we propose in this paper, a Partially Observable Markov Decision Process (POMDP) framework in order to determine an optimal congestion control policy which maximizes the long term expected Quality of Experience (QoE) of the receiver. Moreover, the computation of an optimal policy is usually time/process consuming and as wireless devices are capacity-limited, we consider optimal solutions based on temporal difference (TD-λ) online learning algorithms. Finally, we do some practical experiments of our algorithm on a Microsoft Lync testbed with unidirectional and bidirectional communications over a wireless network. We observe that for both scenarios, our algorithm improves significantly the QoE compared to standard AIMD congestion control mechanism.**

Index Terms—**TCP, POMDP, Learning algorithms, QoE.**

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 [1][2]. However, despite the success of TCP, the existing TCP congestion control is considered unsuitable for delay-sensitive, bandwidth-intense, and loss-tolerant multimedia applications (e.g. real-time video streaming, video-conferences etc.) [2][3]. There are three 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 or bad channel conditions. TCP keeps retransmitting them until they are transmitted successfully even with a large delay. Note that even if the multimedia packets are successfully received, they are not decodable if they are received after their respective delay deadlines.
- Secondly, TCP congestion control adopts an AIMD algorithm, which linearly increases its congestion window size per Round-Trip Time (RTT) when there is no packet loss, and multiplicatively decreases the congestion window size when packet loss occurs. This results in a fluctuating TCP throughput over time, which significantly increases the end-to-end packet delay and that leads to a worse performance for multimedia applications [3].
- Finally, standard TCP congestion control is based on network performance metrics (namely Quality of Service (QoS) metrics) and not on an objective metric of the quality perceived by the user (measured through the QoE). In wireless systems where the environment has an important impact on the quality and moreover, for multimedia applications for which users are very sensitive, a QoE-based congestion control for TCP is welcome.

To mitigate these limitations, a plethora of research has been focusing on smoothing the throughput of an AIMD-based congestion control for multimedia transmission (see [4] and [5]). These approaches adopt various congestion window updating policies to determine how to adapt the congestion window size to the network congestion. However, these approaches seldom explicitly consider the characteristics of the multimedia applications when adapting their congestion window sizes. In [6], Author presented Media-TCP, a quality-centric congestion control for multimedia streaming over IP networks. They have considered the distortion impacts, delay deadlines, and interdependencies of different video packet classes.

When the bottleneck link is overloaded or the channel conditions are bad, the TCP throughput decreases and cannot satisfy the source rate of the multimedia application. This increases, generally, the jitter and the packet loss rate which could impact the user-perceived quality, which is also known as the QoE. Although the QoE is affected by some factors, such as the audio quality, devices, echo, etc., we focus, in this paper, on improving the QoE through a novel congestion control algorithm. The impact of non-networking factors could be cataloged into a protocol stack to form a conceptual relationship between QoS and QoE (see [7] and [8]).

The QoE is measured by MOS values. In a subjective test, the quality of experience is rated on a scale of 1 (bad) to 5 (excellent) by a significant number of people, and the average

Manuscript received 21 August 2011; revised 14 January 2012.

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

Y. Hu, and F. Wu are with Microsoft Research Asia, Beijing, China, 100080 (e-mail: yusuo.hu@microsoft.com, fengwu@microsoft.com).

M. van der Schaar is with the Department of E.E. , UCLA, Los Angeles, USA (e-mail: mihaela@ee.ucla.edu).

Digital Object Identifier 10.1109/JSAC.2012.120808.

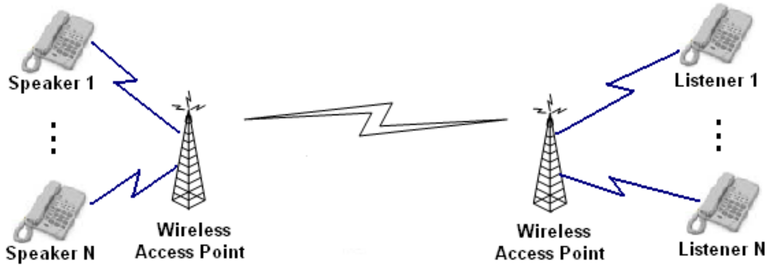

Fig. 1. The experimental model

of the scores is called a mean opinion score. Moreover, the ITU-T Recommendation P.911 [9] provides the reference for carrying out subjective measurement of audiovisual materials.

In this paper, we propose a QoE-aware POMDP-based congestion control algorithm, referred to as MOS-TCP, which exhibits an improved performance when transporting multimedia applications, specifically over a wireless path. Our algorithm is suited for networks containing wireless branches, like the model depicted in Figure 1. The goal of the MOS-TCP algorithm is to control the end-to-end congestion in order to maximize the QoE, where the packets can be lost due to congestion or randomly due to errors encountered across the wireless path. Unlike the current TCP congestion control protocol that only adapts the congestion window to the network congestion (e.g. based on network congestion signals, such as the packet loss rate in TCP Reno, or the round-trip time in TCP Vegas), the proposed congestion control algorithm adopts a two-level congestion control mechanism. Indeed, it adapts over time the congestion window size according to the source rate and the QoE feedbacks. Moreover, we consider a set of updating policies composed of generic congestion control algorithms with general increase and decrease functions, such as AIMD, Inverse Increase/Additive Decrease (IIAD), Square Root inversely proportional Increase/Square Root proportional Decrease (SQRT) and Exponential Increase/Multiplicative Decrease (EIMD).

Note that the multimedia quality obtained by the receivers is partially observable by the senders, and can be evaluated using MOS feedbacks. In order to capture the dynamics of the network congestion and optimize the quality of experience, we formulate the QoE-aware congestion control problem using a POMDP framework. The proposed POMDP framework allows the user to evaluate the network congestion variations over time and then determines an optimal threshold-based congestion window updating policy in order to maximize the long-term discounted reward. In this paper, the reward is defined by the QoE measured through the multimedia quality MOS. A comparative study of several existing congestion control algorithms for multimedia applications and our proposed solution is summarized in Table I.

In summary, we address the following contributions:

a) QoE-aware congestion control: The proposed MOS-TCP provides a QoE-aware approach to adapt the AIMD-like congestion control policy to both the varying network congestion and the multimedia characteristics. Therefore, we take into account the source rate and the packet loss rate of the multimedia packets in the transmission buffer, and their impact on the received quality based on MOS feedbacks.

b) POMDP-based adaptation in the dynamic environment: We propose a POMDP framework to formulate the QoE-aware congestion control problem in a partially observable environment like a wireless system. The framework allows the TCP senders to optimize the congestion window updating policy that maximizes the long-term expected QoE. Furthermore, the network user has a partial knowledge about the bottleneck link status. In fact, the number of packets in 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 is essential for optimizing the performances of multimedia applications.

c) Online learning for QoE-sensitive multimedia applications: Since the computation of an optimal policy is usually time/process consuming and as wireless devices are capacity-limited, we consider optimal solution based on temporal difference (TD-λ) online learning algorithm. We propose practical learning method to solve the POMDP-based congestion control problem on the fly. The proposed model-free learning algorithm is based on TD-λ reinforcement learning and is designed for QoE-sensitive multimedia applications.

The paper is organized as follows. We introduce the QoE and explain the MOS calculation in Section II. In Section III, we model the QoE-aware congestion control problem that maximizes the performance of multimedia applications. Then in Section IV, we formulate the problem using a POMDP-based framework. We present a low-complexity algorithm to solve the POMDP in Section V. Section VI provides the experimental results that validate the proposed method and Section VII concludes the paper.

II. QoE-AWARE NETWORKING AND MOS MEASUREMENT

To overcome the limitation of QoS-based optimization, QoE-based approaches have been introduced as a more effective way to optimize transmission algorithms and protocols with respect to user satisfaction. QoE metrics are defined as a set of quantitative measures to assess the perceived QoS of end users [11]. Also, a new approach, namely *QoE-aware networking*, is proposed to re-formalize the service optimization problem and to improve the user experience. Because the QoE metrics reflect the end user's experience, QoE-based approaches can improve the subjective service

TABLE I
COMPARISONS OF CURRENT CONGESTION CONTROL SOLUTIONS FOR MULTIMEDIA STREAMING

Algorithm	TCP-Friendluness	Multimedia support	Content dependency	Decision Type
RAP [10]	AIMD-based	Source rate adaptation	No	Myopic
GAIMD [4]	AIMD-based	Playback buffering	No	Myopic
Binomial Algorithm [5]	Binomial scheme	Source rate adaptation	No	Myopic
MOS-TCP	AIMD-like media aware	Quality-centric congestion control	Yes	Foresighted

quality, optimize the use of the network resources, and provide services to more users without noticeable degradation of user experience. Recently, QoE metrics have been used to optimize various types of network services. In cellular systems, the authors of [12] use a QoE-based approach to allocate downlink wireless resources among different applications. They define several QoE models for different types of applications such as file downloading, voice call and video streaming and adopt QoE-based utility maximization to improve user perceived quality. In [13], the authors apply QoE metrics to optimize IEEE 802.11 wireless LAN. They use a machine learning approach to generate real-time QoE measurements and use the QoE feedbacks to manage wireless network resources. In [14], the authors use QoE metrics for packet scheduling in multi-hop wireless networks. The packet scheduler determines the packet drop pattern that minimizes the degradation of MOS values. In P2P networks [15], scalable video coding and QoE metrics are used to optimize the performance of P2P video streaming systems. In this paper, we seek to enable QoE-awareness in a more general setting. We integrate the QoE metrics with the TCP protocol. Since TCP is a widely adopted building block in many network services, our approach is applicable to a much wider spectrum of applications.

Since QoE metrics are subjective, the standard QoE measuring process should involve human observers, e.g., when measuring VoIP quality, the MOS are often used as a subjective rating ranging from 1 (poor) to 5 (excellent). However, to enable QoE-awareness in multimedia services, it is infeasible to use subject human tests for real-time applications. Instead, some QoE online prediction methods should be used to estimate QoE from the service output. The QoE prediction methods are dependent on the types of content. A number of models have been proposed for predicting QoE with different kinds of contents including web service[16], voice services [17], audio/video content [18], etc. Instead of proposing another new approach of QoE prediction, we base our experiments on the QoE prediction results produced by an existing real system, i.e. Microsoft Lync system [19] (previous known as Office Communication Server and Office Communicator [20]). In the Lync system, the VoIP software measures a set of variables which may affect the quality of experience throughout the communication sessions. Based on the collected measurements, it can predict the subjective QoE metrics in real-time. Furthermore, the QoE metrics are normalized and represented in the standard MOS. Our considered Lync software provides several types of MOS values (NetworkMOS, ListeningMOS, conversationalMOS) in order to represent the degradation in different phases of the whole communication process (see Figure 2). The MOS prediction mechanism provides a quantitative approach to evaluate the communication quality that the end users have experienced.

- NetworkMOS is calculated purely based on the obtained network statistics (information), which include the packet loss, bit errors, packet delay, and jitter.
- ListeningMOS is not only decided by network parameters but also by the choice of audio codec and audio devices, as well as the recording conditions such as echo, background noise level, talk-over, etc. It captures the perceived quality of an audio stream at the received side. Note that both NetworkMOS and ListeningMOS are only measured for unidirectional traffic.
- ConversationalMOS is measured for both sending and receiving streams. It takes into account the round-trip delay in addition to all the above-mentioned factors.

Observing these different MOS values gives us a clear perspective on the performance of the entire communication process. A congested network, for example, will cause degradation in NetworkMOS, while a bad recording device can be identified from low ListeningMOS values.

III. QoE-AWARE CONGESTION CONTROL PROBLEM

A. Network Settings

We assume that the network has a set of N end users indexed $\{1, \cdots, N\}$. Each user is composed by a sender node and a receiver node that establish an end-to-end transport layer connection. Let $w_n \in [0, w_n^{max}]$ represents the congestion window size of user n, where w_n^{max} represents the maximum value of the congestion window size.

The network system has some bottleneck links, which result in packet losses when the buffers at the bottleneck links are overloaded. However, the congestion status at the bottleneck links is not observable by the end users. An end user n infers the congestion status by observing the acknowledgments per RTT and the quality of experience from the QoE feedbacks. In fact, the user observes these feedbacks, which depend on packet loss rate, jitter, etc., and acts in order to optimize the expected QoE. Before transmitting a packet, the user verifies if its delay is lower than the deadline delay, if not, it drops the packet and it considers the packet as lost. For each acknowledgement, the end user n observes congestion event $o_n \in \{Success, Fail\}$ (the packet being received successfully or not by the receiver) and the MOS feedback $MOS_n^k \in \mathcal{M}$ at the k-th epoch. The observed information is available to the senders through transmission acknowledgments (ACK) built into the protocol (see [1]). In fact, the sender orders transmitted packets by a sequence number and the receiver acknowledges the sequence number of the received packets. Indeed, if the receiver determines that a packet is out of order, it acknowledges the last successfully received one and therefore, the sender receives a duplicate ACK (DUP ACK). The user supposes that there is a congestion if it receives three

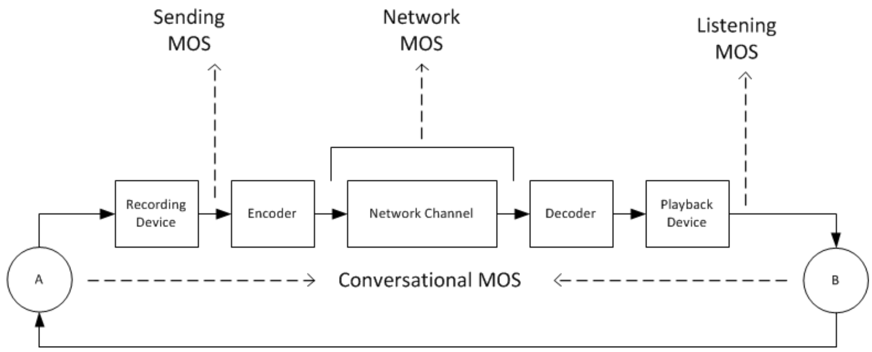

Fig. 2. Different MOS measurements in Microsoft Lync system

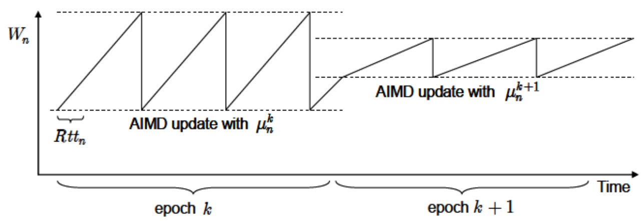

Fig. 3. Congestion window size with different update policies per epoch.

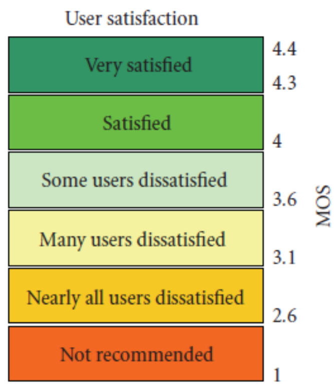

Fig. 4. Relation between MOS and user satisfaction [18].

DUP ACKs or if the timeout expires. The observed MOS is available to the sender through the QoE feedbacks.

B. Two-Level Congestion Control Adaptation

A general description regarding the congestion control window size variation can be described by:

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

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 (in this paper, we refer to $f(w_n)$ as the increasing function and $g(w_n)$ as the decreasing function), and $\mathcal{A}$ represents the set of all possible updating policies. Some existing examples of the updating policies can be found in [4] and [5].

Unlike the existing TCP congestion control that fixes the congestion window updating policy without considering the characteristics of the transmitted applications, the proposed MOS-TCP in this paper adopts a two-level adaptation mechanism to update the congestion window. We define the congestion control epoch $Epoch_n$ as T RTTs such that the user n periodically changes its window updating policy. First, we allow the sender to select its updating policy μ_n at the beginning of each epoch and cannot change it until the next epoch. Denote μ_n^k as the congestion window updating policy of user n in its k-th epoch. Then, it adapts its own congestion window size per RTT based on the updating policy μ_n^k during the k-th epoch. Figure 3 provides an illustrative example to show how the congestion window size varies over time. This paper then focuses on how to optimally determine the updating policy, at each epoch, to improve the QoE.

C. Expected multimedia quality per epoch

In this section, we discuss the objective of the proposed QoE-aware congestion control. Denote by R_n^k the source rate of a multimedia application for user n in the k-th epoch. The source rate is the average number of packets that arrives at the transmission buffer per second at the transport layer. In fact, in a VoIP call, the source rate can be controlled and adapted to the network environment, since there are usually some rate control modules implemented in VoIP software.

We propose, in this paper, a congestion control algorithm that dynamically changes the congestion window updating policy in order to maximize the QoE. Therefore, it is straightforward and somehow intuitive that each user has as objective to maximize its own QoE. As we can see in Figure 4, the MOS is correlated with the listener satisfaction. The higher MOS, the greater the listener's satisfaction. Therefore, the objective of users is to maximize the expected future MOS starting from the current slot. A similar utility function has been used in [21]. Each user tries to optimize, selfishly, the following expected total discounted MOS:

$$U_n = \sum_{k=1}^{\infty} \gamma^k u_n^k(\mu_n^k, R_n^k), \quad (2)$$

where γ is a discount factor and $u_n^k = MOS_n^k(\mu_n^k, R_n^k)$ is the received MOS by the user n at the k-th epoch, when the source rate at the k-th epoch is R_n^k and the user n uses the congestion window updating policy μ_n^k. Note that our MOS-TCP mechanism allows the user to maximize its expected

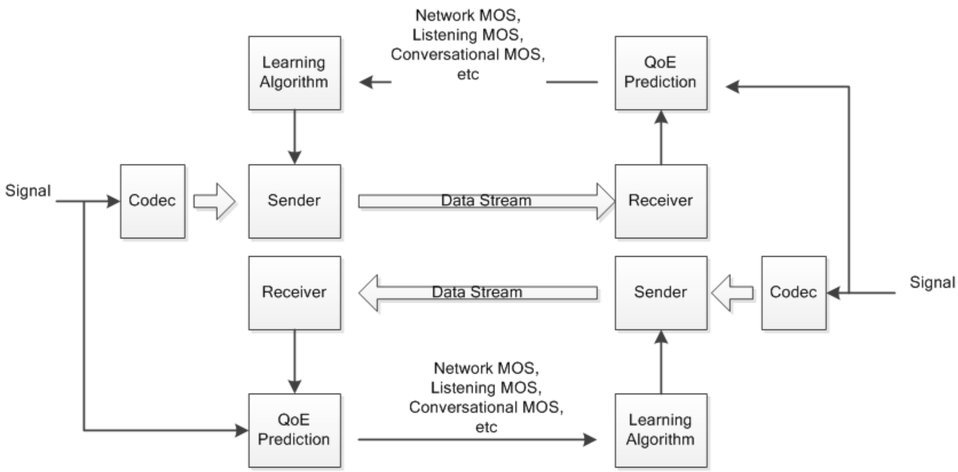

Fig. 5. MOS exchange in bidirectional conversation.

total discounted MOS by choosing the optimal congestion window updating policy μ_n^k. In fact, the QoE varies with the source rate, the congestion window updating policy and the congestion status at the bottleneck links. The latter depends not only on the user n but also on other users. We show, in the next section, how the proposed POMDP-based congestion control algorithm determines the optimal updating policy given partial knowledge of bottleneck links status.

IV. POMDP-Based Congestion Control

Our aim is to propose a congestion control algorithm that maximizes the expected QoE expressed by Equation (2). We formulate our problem using a POMDP-based framework as the global system state is not well known for users. In fact, the user has a partial knowledge about the congestion status at the bottleneck links. The latter depends on the congestion windows of all users, which is unknown by the user n. Thus, the user n has to estimate solely the impact of all the other users based on the history of observations and actions. In fact, the user n estimates the packet loss rate when it transmits data using the congestion window w_n. We present, in the next section, a POMDP-based congestion control algorithm that maximizes the QoE.

We define a POMDP-based congestion control of user n in a tuple $\{\mathcal{A}, \mathcal{X}_n, O_n, \Omega_n, P_n, U_n\}$:

1) Action: The user selects the congestion window updating policy $\mu_n = \{\mu_n^1, \mu_n^2, ...\} \in \mathcal{A}$, with μ_n^k is the updating policy of user n in the k-th epoch.

2) State: The state is defined as $X_n^k = \{p_n^k, R_n^k\} \in \mathcal{X}_n$. The source rate R_n^k is known by the user n, however, the packet loss rate p_n^k, which is impacted by other users' windows, cannot be directly observed. The user n has to infer the congestion status of the bottleneck links using observations and QoE feedbacks. The belief of the packet loss rate is defined as $b : [0, 1] \rightarrow [0, 1]$. The function $b(.)$ represents the probability distribution of the packet loss rate.

3) Observed information and observation probability: The observed information is defined by the congestion events $o_n \in O_n$. The observation probability is defined as a function $\Omega_n : \mathcal{T}_n \times O_n \rightarrow [0, 1]$. Let $\Omega_n^{k-1}(o_n = fail|w_n)$ represent the probability of packet loss when the congestion window size is observed as w_n at the $(k-1)$-th epoch. Moreover, n receives the feedback MOS_n^k at the end of each epoch.

The conventional POMDP updates the belief function per time slot (RTT), but in the proposed POMDP framework, $b_n(p_n^k)$ is updated per epoch. In fact, the belief distribution is kept the same within the epoch, which reduces the computational complexity and also the memory requirement for calculating the optimal policy.

4) State transition: The average packet loss rate p_n^k when using the congestion window updating policy μ_n^k at the k-th epoch cannot be known by n until the end of the epoch. Instead, the user estimates it based on the following expression:

$$b(p_n^{k+1}|\mu_n^{k+1}) = \frac{Prob(p_n^{k+1}|p_n^k, \mu_n^{k+1})}{\sum_p Prob(p|p_n^k, \mu_n^{k+1})}, \tag{3}$$

where $Prob(p_n^{k+1}|p_n^k, \mu_n^{k+1})$ is the probability that the packet loss rate will be p_n^{k+1} at the $(k+1)$-th epoch when choosing the policy μ_n^{k+1}, given that the packet loss rate is p_n^k.

Based on the MOS feedbacks obtained at the end of every epoch, the user chooses the updating policy that maximizes the QoE as illustrated in Figures 5 and 6.

V. MOS-based POMDP Algorithms

We propose, in this section, a POMDP-based algorithm in order to maximize the QoE for multimedia applications. Every epoch, MOS-TCP users get three feedbacks: Network-MOS, ListeningMOS and ConversationalMOS as illustrated in Figure 5. These feedbacks reflect the listener satisfaction and therefore the user has to choose the action that improves the total expected QoE. Therefore, based on these feedbacks, we propose a POMDP-based algorithm that maximizes the expected QoE. Furthermore, as solving POMDPs is an extremely difficult computational problem, we present low computation complexity online learning algorithm in order to solve the POMDP problem when the state transition probabilities are not available. Online learning algorithms are very useful in wireless systems as they require low complexity.

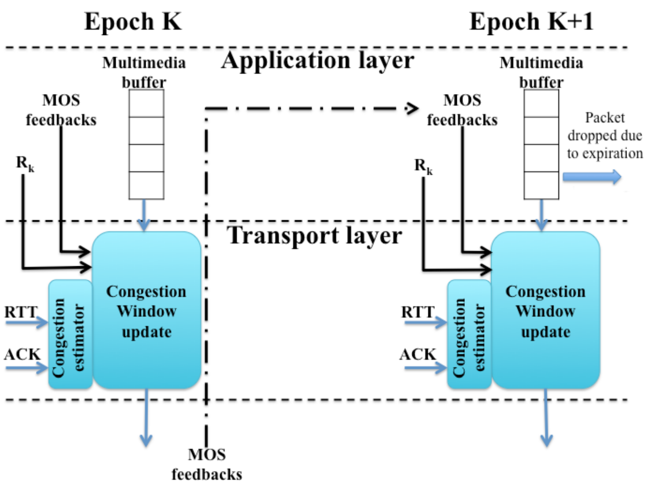

Fig. 6. System diagram of MOS-TCP in time epoch k and $k+1$.

A. Packet-loss Differentiation

The obstacle that wireless networks have to face is high BER. The fast recovery algorithm solves the single packet loss within one window. However, due to the nature of wireless networks, a fading channel may cause contiguous packet loss. Therefore, the key idea of designing a wireless TCP is to distinguish the cause of packet loss. Many schemes have been proposed in the literature. For example, TCP Veno [22] estimates the backlogged packets in the buffer of the bottleneck link. It determines the optimal throughput the network can accommodate based on the minimal RTT. The difference between the optimal throughput and the actual throughput can be used to derive the amount of backlogged packets. TCP Veno suggests that the loss is said to be random if the number of backlogged data is below a threshold, otherwise the loss is congestive. We consider the same methodology in order to distinguish between random packet loss and congestive loss.

B. The Objective function

As our objective is to avoid congestion at the bottleneck links and improve the QoE, the MOS represents a consistent feedback that gives us some information about the impact of the congestion status on the multimedia quality. The MOS feedback varies with the packet loss rate and the jitter interval, which are related to the congestion status at the bottleneck links. The higher the MOS, the better the QoE and the lower the packet loss rate and the jitter interval. Thus, our objective is to maximize the total expected received MOS. Depending on the multimedia application, the user maximizes the expected QoE using NetworkMOS, ListeningMOS or Conversational-MOS feedback. All these MOS feedbacks depend on the packet loss rate and on the jitter interval, both of which depend on the source rate and the congestion window updating policy.

C. The Optimal Policy

A policy which maximizes U_n is called an optimal policy $\mu_n^{opt} = \{\mu_n^{opt,1}, \mu_n^{opt,2}, ...\}$; it specifies for each epoch k the optimal updating policy $\mu_n^{opt,k}$. The value of an optimal policy μ_n^{opt} is defined by the optimal value function U_n^{opt}, that satisfies the following Bellman equation:

$$\begin{aligned} U_n^k(p_n^k) &= \max_{\mu_n^k \in \mathcal{A}} \{u_n^k(\mu_n^k, R_n^k) \\ &\quad + \gamma \sum_{p'} b(p_n^k) T(p'|p_n^k) J_n^{k+1}(p')\}. \end{aligned} \tag{4}$$

The optimal policy at the k-th epoch is therefore:

$$\begin{aligned} \mu_n^{opt,k} &= \arg\max_{\mu_n^k \in \mathcal{A}} \{u_n^k(\mu_n^k, R_n^k) \\ &\quad + \gamma \sum_{p'} b(p_n^k) T(p'|p_n^k) J_n^{k+1}(p')\}. \end{aligned} \tag{5}$$

D. Online learning

Solving the presented POMDP is expensive in terms of time (calculation) and space (memory) complexity, and then it is not suitable for wireless systems with small capacity multimedia devices. We present a low-complexity online learning algorithm. Our online learning is an extension of the on-policy TD-λ algorithm Sarsa [23] for POMDPs. An agent learns the longer but safer path than Off-policy learning algorithms such as Q-Learning [24], and therefore receives a higher average reward even though it does not follow the optimal path.

Each MOS-TCP user estimates the action-values $Q(\mu_n^k, R_n^k, p_n^k)$, defined as the expected future reward starting from state (R_n^k, p_n^k) and taking the action μ_n^k. The policy is chosen based on the current estimates such that they approach the optimal action-values. Moreover, the user selects the congestion window updating policy according to a ϵ-greedy policy. For a small fraction of the time, it selects randomly from the action state $\mathcal{A}$. The MOS-TCP user chooses at every epoch the optimal policy based on Algorithm 1. Interestingly, this algorithm supports the delay of MOS feedbacks as it updates the congestion window updating policy per epoch. As illustrated in Figure 6, the user gets some feedbacks at the end of each epoch, which reflects the impact of the network on the listening quality. Therefore, the user applies the online learning algorithm in order to choose the congestion window policy that maximizes the expected future MOS starting from the current slot. At the beginning of epoch k, the user receives the source rate R_n^k from the upper layer and selects the congestion window updating policy which maximizes its action-values. Then, the user transmits its packets during the epoch using the chosen policies. At the end of the epoch, the user computes the packet loss rate and updates the action-values $Q(\mu_n^k, R_n^k, p_n^k)$ based on the observed MOS feedback. Moreover, the user updates the belief probability of the packet loss rate. Depending on the MOS feedback considered in the objective function, we denote Network-CC the MOS-TCP algorithm that maximizes the NetworkMOS, Listening-CC the MOS-TCP that maximizes the ListeningMOS, and Conversational-CC the algorithm that maximizes the expected ConversationalMOS.

E. Implementation and Complexity

Although the Value Iteration algorithms give exact solutions for the POMDP optimization problems (see [25]), they need expensive time and space complexities. In fact, the sender

Algorithm 1 MOS-TCP online learning algorithm for POMDP-based congestion control

Initialize $Q(\mu_n^k, R_n^k, p_n^k) = 0$;
$k \leftarrow 1$;
Get application parameters R_n^1;
Choose arbitrarily the updating policy (μ_n^1);
while true **do**
  for $t = 1 \rightarrow T$ **do**
    Update the congestion window using policy μ_n^k ;
    Update the observation probability Ω_n^k based on congestion event observation;
  end for
  Evaluate the packet loss rate p_n^k;
  The user gets the QoE feedbacks: MOS;
  Update the beliefs based on Equation (3);
  Get application parameters R_n^{k+1};
  Choose updating policy

$$(\mu_n^{k+1}) = \arg\max_{\mu_n^{k+1}} Q(\mu_n^{k+1}, R_n^{k+1}, p_n^{k+1}) b_n(p_n^{k+1}),$$

  with probability $(1-\epsilon)$;
  Else choose a random policy in $\mathcal{A}$;

$$Q(\mu_n^k, R_n^k, p_n^k) \leftarrow Q(\mu_n^k, R_n^k, p_n^k) + \alpha[MOS + \gamma \times \sum_{p_n^{k+1}} Q(\mu_n^{k+1}, R_n^{k+1}, p_n^{k+1}) b_n(p_n^{k+1}) - Q(\mu_n^k, R_n^k, p_n^k)];$$

  $k \leftarrow k+1$;
end while

needs a large storage space and spends an exponential time when seeking for the optimal policy. As we can see in the Table II, the complexity of the exact POMDP solutions grows exponentially with the number of epoch. Importantly, our online learning algorithm can be implemented on mobile devices which do not have a sophisticated calculation nor a large memory space. In fact, the MOS-TCP algorithm needs a low time and space complexities as illustrated in the Table II. Furthermore, it supports the real-time constraints of audiovisual multimedia applications. Moreover, the proposed algorithm is implemented only on the transmitter side and is transparent to the receiver side. We do not even require any change at the routers. Interestingly, this algorithm supports the delay of MOS feedbacks as it updates the congestion window updating policy per epoch.

F. TCP-Friendliness

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 [26] and [4]). 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 [27]). We prove that the MOS-TCP user do not violate the TCP-friendliness rule if the

TABLE II
COMPARISONS OF EXACT POMDP SOLUTION AND THE PROPOSED ONLINE LEARNING ALGORITHMS

	Exact solution	MOS-TCP
Consumed Memory	$O(\lvert\mathcal{A}\rvert\lvert\mathcal{V}_{k-1}\rvert^{\lvert\mathcal{O}\rvert})$, with $\mathcal{V}_k$ is the solution of the $(k-1)$-th epoch	$O(\lvert\mathcal{A}\rvert\lvert\mathcal{X}\rvert)$
Time complexity	$O(\lvert\mathcal{X}\rvert^2\lvert\mathcal{A}\rvert\lvert\mathcal{V}_{k-1}\rvert^{\lvert\mathcal{O}\rvert})$	$O(\lvert\mathcal{A}\rvert\lvert\mathcal{X}\rvert)log(\lvert\mathcal{A}\rvert\lvert\mathcal{X}\rvert)$

following relation between $f(w)$ and $g(w)$ is satisfied:

$$f(w) = \frac{3g(w)}{2-g(w)}. \tag{6}$$

The proof is a generalization of the proof of [4] and [28] 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^{MOS} and w^{AIMD} the congestion windows of the MOS-TCP transported flow and the AIMD-Transported flow respectively. Assume that both flows have the same RTT and MSS. The effect due to different RTT and Maximum Segment Size (MSS) is beyond the scope of this paper and will be studied as part of our future research. On one hand, when $w^{MOS} + w^{AIMD} < r$, the link is in the underloaded region and thus, the congestion windows w^{MOS} and w^{AIMD} evolves as follows:

$$w^{MOS}(t+\Delta t) = w^{MOS}(t) + f(w^{MOS}(t))\Delta t \tag{7}$$
$$w^{AIMD}(t+\Delta t) = w^{AIMD}(t) + \Delta t. \tag{8}$$

On the other hand, when $w^{MOS} + w^{AIMD} \geq r$, the link is overloaded and congestion occurs. We assume that both flows receive the congestion signal simultaneously and we denote t_i the $i-th$ time that the link is congested. Both flows decrease simultaneously their window based on the following expression:

$$\begin{aligned} w^{MOS}(t_i) + w^{AIMD}(t_i) &= r \\ w^{MOS}(t_i^+) &= w^{MOS}(t_i)(1 - g(w^{MOS}(t_i))) \\ w^{AIMD}(t_i^+) &= \frac{1}{2} w^{AIMD}(t_i). \end{aligned}$$

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:

$$\begin{aligned} w^{MOS}(t_{i+1}) - w^{MOS}(t_i) &= -\frac{2g(w^{MOS}(t_i)) + f(w^{MOS})}{2(f(w^{MOS})+1))} w^{MOS}(t_i) \\ &+ \frac{rf(w^{MOS})}{2(f(w^{MOS})+1))}. \end{aligned}$$

Thus, independent of the initial values of w^{MOS} and w^{AIMD}, after a sufficient number of cycles, the congestion windows of these two flows in the overloaded region converge to:

$$w^{MOS}(th) = \frac{f(w^{MOS})r}{2g(w^{MOS}) + f(w^{MOS})}, \tag{9}$$
$$w^{AIMD}(th) = \frac{2g(w^{MOS})r}{2g(w^{MOS}) + f(w^{MOS})}. \tag{10}$$

Therefore, in the steady state, w^{MOS} and w^{AIMD} increase and decrease periodically. Their average throughput in steady state are expressed by the following:

$$\bar{w}^{MOS} = \frac{(2-g(w^{MOS}))f(w^{MOS})r}{4g(w^{MOS})+2f(w^{MOS})}, \quad (11)$$

$$\bar{w}^{AIMD} = \frac{3g(w^{MOS})r}{4g(w^{MOS})+2f(w^{MOS})} \quad (12)$$

To guarantee the TCP-Friendliness, a necessary and sufficient condition is $f(w) = \frac{3g(w)}{2-g(w)}$.

VI. Numerical Illustrations

A. Testbed experiments

Microsoft Lync is an integrated software-based communication and collaboration platform which is mainly designed for enterprise users. It provides various real-time communication features such as instant messaging, software-based voip, video/audio conferencing through the same user interface. The system includes a set of server components that can be deployed in the enterprise network. After installing the client-side component, authorized enterprise users can initiate audio/video calls with others or set up a group conference through the IP network. Furthermore, it supports communications with traditional phone through some PSTN gateway.

The system supports the standard Session Initiate Protocol (SIP) for signaling and RTP/RTCP protocols for transmitting media packets. For the two-way communications, the clients can directly connect with each other and transmit data in a peer-to-peer way. For multi-users conferencing sessions, a Multimedia Controller Unit (MCU) server can help to coordinate the session and to replicate the data packets to all the receivers. When the users are behind some Network Address Translator (NAT) or firewalls, a mediation server allows the clients to relay the data packets. The MOS prediction module in Lync is implemented at the application layer and is independent on the transport protocol. The underlying transport protocol in Lync can be either TCP, UDP, or even server-relayed tunnels (e.g. Traversal Using Relay NAT (TURN) protocol), depending on the connectivity of the Lync clients.

The proposed algorithm is implemented only at the sender side, and is transparent to the routers and the receiver. However, an end-to-end signaling mechanism need to be implemented in the application layer on both the transmitter and the received side. Note that a library-based MOS feedback mechanism can be adopted to help developers of multimedia applications to design QoE based multimedia applications without the need of run-time training and signaling.

The MOS feedbacks need to be sent from the receiver side to the sender for every epoch. The MOS prediction and the feedback are located at the application layer. Thus there is no need to modify the receiver part of the TCP code. Meanwhile, the TCP sender part can be designed to be backward compatible, i.e. the sender works in normal mode when there is no MOS feedback and will switch to the MOS-based congestion control mode only when the application layer has indicated it to do so. In this way, the MOS-based TCP clients can still interact with the old non-MOS version ones.

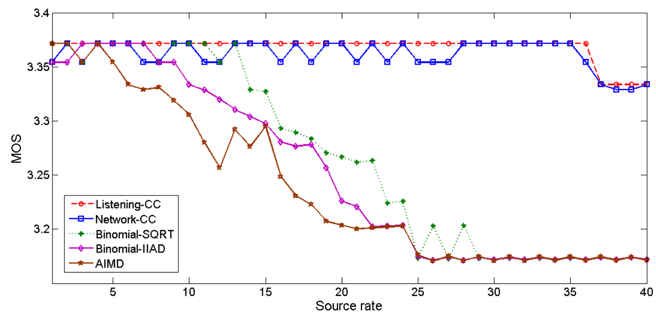

Fig. 7. ListeningMOS with different source rates in the first scenario.

In our experiments, the QoE trace is captured and anonymized from a deployed Microsoft Lync 2010 Service in the global enterprise network. The duration of the collected trace is about three months. The average length of each session is 11 minutes. From the original trace, we extract only the PC-to-PC audio streams since it reflects the voice quality over pure IP networks. The extracted part contains 1,935,110 end-to-end audio streams in total. The audio codec used by the clients is Microsoft RTAudio Speech codec with the clock rate 16KHz. We have used the Gilbert model for modeling the wireless channel conditions. This approach has been introduced in [29]. By generating synthetic traces that simulate the wireless network being tested, multiple users can access the network simultaneously and perform experiments.

We consider a set of policies $\mathcal{A}$ composed of AIMD, IIAD and SQRT, defined as follows:

$$\text{AIMD: } f(w) = \frac{3\beta}{2-\beta} \text{ and } g(w) = \beta;$$

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

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

where $\beta \in \{0.1, 0.2, ..., 0.9\}$. Note that the conventional TCP is AIMD(0.5). We compare our proposed algorithms with other congestion control algorithms for multimedia applications. We focus, especially, on AIMD and Binomial congestion control algorithms. In fact, authors of [5] has proven that the AIMD-based Binomial congestion control algorithms IIAD and SQRT are well suited for multimedia applications. We consider that the data is transmitted over a IEEE 802.11a wireless link and the playback delay is 200 ms. We have used IEEE 802.11a in our numerical study only for illustrative purposes, and any kind of wireless device can be used instead.

B. Unidirectional Communications

In this section, we focus on the unidirectional communications with a speaker and a listener in each session. We present a comparative study between Listening-CC, Network-CC and other congestion control algorithms. We compare, in different scenarios, the QoE (ListeningMOS) and we consider the following congestion control algorithms: Listening-CC,

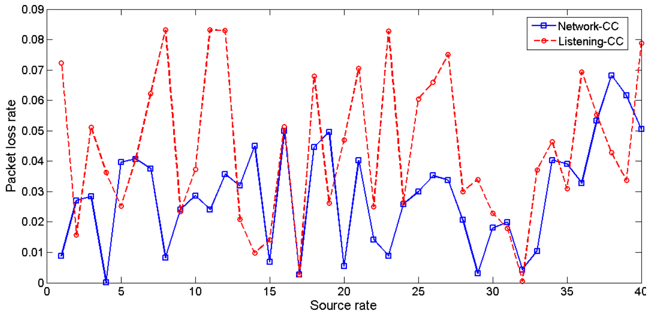

Fig. 8. Packet loss rate depending on the source rate in the first scenario.

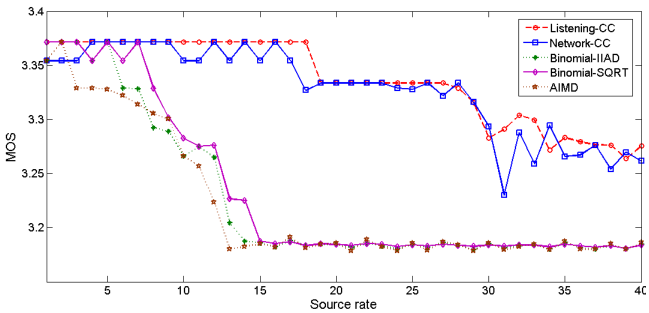

Fig. 9. ListeningMOS with different source rates in the second scenario.

Network-CC, Binomial congestion control and AIMD algorithm. We do not compare with Conversational-CC as we are considering unidirectional communications. In the first scenario, we consider two pairs of AIMD users; two pairs of SQRT users; two pairs of IIAD users; two pairs of Listening-CC users and two pairs of users with Network-CC. Each pair is composed of a transmitter (speaker) and a receiver (listener).

We run audio transmissions with different source rates and we plot in Figure 7 the obtained QoE for different type of users. We can observe that the Listening-CC and Network-CC algorithms improve significantly the QoE compared to AIMD and Binomial congestion control algorithms. Moreover, the MOS obtained with Listening-CC is slightly better than the MOS obtained by the Network-CC algorithm. Furthermore, as we can see in Figure 8, the packet loss rate for Listening-CC users is higher than Network-CC. In fact, as the NetworkMOS depends only on network factors, maximizing this MOS minimizes the packet loss rate and the jitter interval. However, Listening-CC based on ListeningMOS which depends on other factors than the network ones and therefore the users can choose the policy that maximize the ListeningMOS even with higher values of packet loss rate and jittering.

In the second scenario, we consider four pairs of AIMD users; four pairs of SQRT users; four pairs of IIAD users; four pairs of Listening-CC users and four pairs of Network-CC users. We can observe in Figure 9 that Listening-CC and Network-CC algorithms lead to better QoE than Binomial and AIMD users. Moreover, Listening-CC leads to slightly better QoE than Network-CC. Figure 10 illustrates that the packet loss rate for both Listening-CC and Network-CC algorithms is

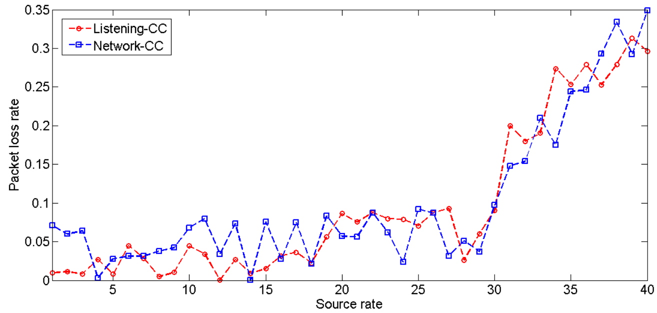

Fig. 10. Packet loss rate depending on the source rate in the second scenario.

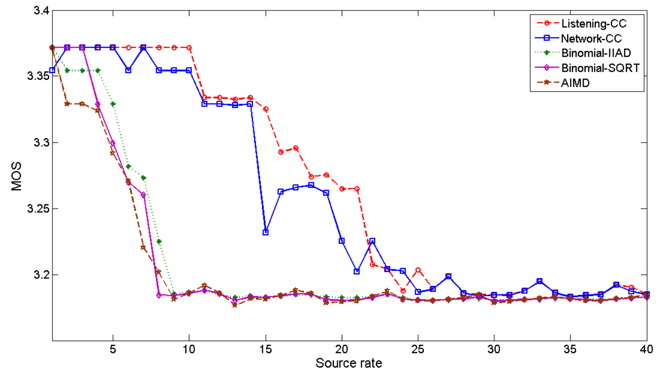

Fig. 11. ListeningMOS with different source rates in the third scenario.

increasing with the source rate as the bottleneck link become overloaded. The fluctuation of packet loss rate is due to the imperfect characteristics of the wireless link.

In the third scenario, we consider more load on the bottleneck link. We keep the same wireless link and we consider 8 pairs of AIMD users; 8 pairs of SQRT users; 8 pairs of IIAD users; 8 pairs of Listening-CC users and 8 pairs of Network-CC users. Figures 11 and 12 illustrates the ListeningMOS and the packet loss rate for different congestion control algorithms. It is clear that the MOS-TCP frameworks leads to better QoE, however, the improvement decreases with the source rate and all congestion control algorithms give the same QoE for high values of source rate. In fact, with such number of audio sessions and source rates, the wireless link is always overloaded and the source rates requested by users cannot be satisfied. Therefore, the packet loss rate increases for all the users and therefore the QoE decreases.

In summarize, both Listening-CC and Network-CC algorithms improve the QoE compared to other AIMD-based congestion control algorithms for multimedia transmission. Moreover, Listening-CC is slightly better than Network-CC algorithm, as it considers not only packet loss rate and jitter but also the impact of non-network factors.

C. Bidirectional Communications

We consider, in this section, bidirectional audio conversations. We run the three scenarios presented in Section VI-B with a bidirectional communication.

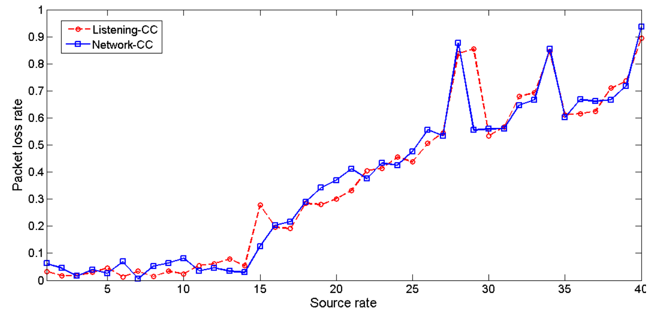

Fig. 12. Packet loss rate depending on the source rate in the third scenario.

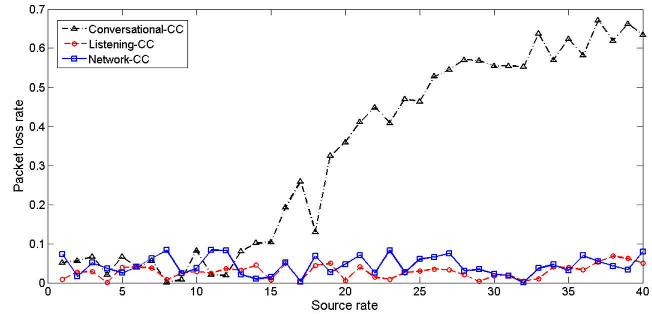

Fig. 14. Packet loss rate depending on the source rate in the first scenario.

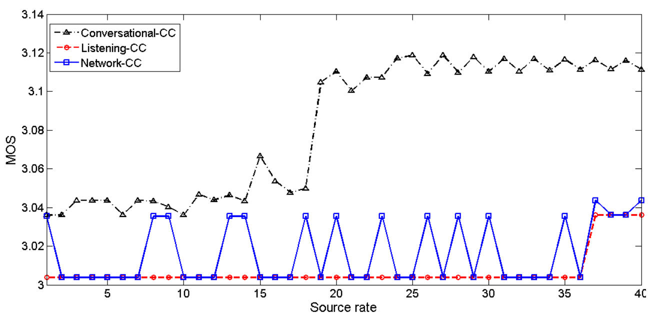

Fig. 13. ConversationalMOS with different source rates in the first scenario.

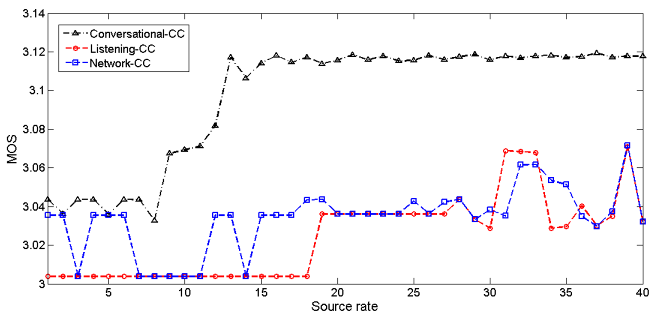

Fig. 15. ConversationalMOS with the source rates in the second scenario.

In the first scenario, we run the conversations over the wireless link with two more pairs of user using the Conversational-CC algorithm. Figure 13 illustrates the conversationalMOS for Conversational-CC and other congestion control algorithms with different values of the source rate. We can observe that the Conversational-CC leads to better QoE than Listening-CC and Network-CC algorithms. Surprisingly, the improvement of the Conversational-CC compared to Listening-CC and Network-CC algorithms is more important for higher source rate. In fact, for high values of source rate, we can observe in Figure 14 that Conversational-CC algorithm is more aggressive than other congestion control algorithms as it leads to significantly higher packet loss rate.

In the second scenario, we add four pairs of Conversational-CC users. We plot in Figures 15 and 16 the Conversational-MOS for different congestion control algorithms depending on the source rate. We can observe that the Conversational-CC algorithm outperforms other congestion control algorithms. Moreover, we remark that for some values of the source rate, the Listening-CC in better than Network-CC and for other values Network-CC is better.

We consider eight more pairs of Conversational-CC users when running the third scenario. Figure 17 shows that the Conversational-CC leads to better QoE than other congestion control algorithms. In fact, it bases on the conversationalMOS feedback which takes into consideration both sent and received audio streams and is less sensitive to the network factors, such as packet loss rate and jittering, than ListeningMOS and NetworkMOS. However, as we can see in Figure 18, when the Conversational-CC algorithm is higher than Listening-CC and Network-CC, it leads to higher packet loss rate. Moreover, when the wireless link is overloaded, all the congestion control algorithms give the same QoE. Finally, the Conversational-CC is more suitable for bidirectional communication than other congestion control algorithms.

Although the improvements in MOS do not seem to be very large (0.1-0.3) in absolute values, the relative improvements are actually significant. In the practical system (e.g., Microsoft Lync), only few users have MOS values below 3 or above 4. The dynamic range of the MOS values is about 1.0. The region between 3.0 and 4.0 is a quite sensitive interval of MOS for users. Our improvement is about 10% to 30% in the range. Because the sessions in our traces are using the same audio codec and software version, this means that the actual degradation of ListeningMOS is relatively small. However, if we focus on the NetworkMOS, the improvements are significant. As we can see in Figure 19, the improvement of MOS-TCP user is about 1 in NetworkMOS.

VII. Conclusion

We formulate, in this paper, the QoE-aware congestion control problem as a POMDP that maximizes the quality of experience for multimedia users. We consider a set of generic AIMD-like updating functions for the congestion window. The optimal policy allows the sender to optimize the congestion window updating policy that maximizes the long term expected quality of experience. We also propose an online learning method to solve the MOS-TCP on the fly. The experimental results show that the proposed algorithm outperforms other congestion control schemes in terms of

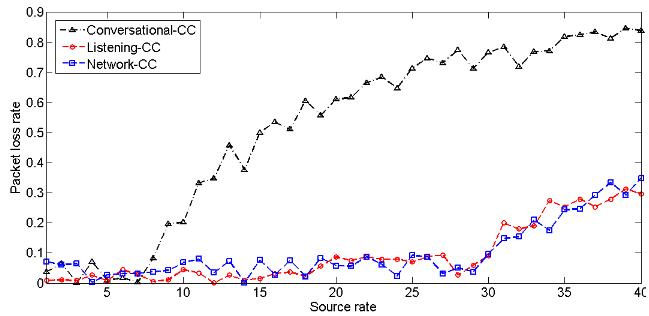

Fig. 16. Packet loss rate depending on the source rate in the second scenario.

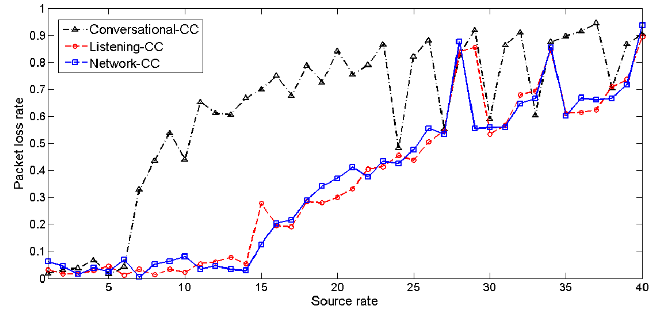

Fig. 18. Packet loss rate depending on the source rate in the third scenario.

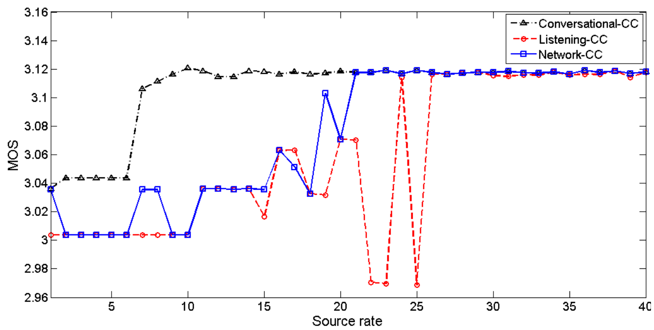

Fig. 17. ConversationalMOS with different source rates in the third scenario.

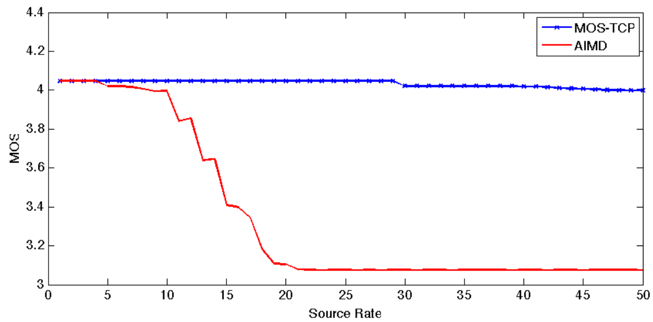

Fig. 19. NetworkMOS for MOS-TCP user and a AIMD user.

QoE. The proposed QoE-based adaptation can be straightforwardly extended to video applications. The only difference is that video or graphics based QoE feedback is needed to train the QoE-decision based engine which adapts the TCP transmission. As a part of our future work, we will extend the proposed algorithm to support a wider set of applications such as multicast applications.

ACKNOWLEDGMENT

This work was performed while Mr. Oussama Habachi and Dr. Feng Wu were visiting UCLA.

REFERENCES

[1] J. Padhye, V. Firoiu, D. F. Towsley and J. F. Kurose, *Modeling TCP Reno Performance: a simple model and its empirical validation*, IEEE/ACM Trans. Netw., vol. 8, no. 2, pp. 133-145, 2000.

[2] B. Wang, J. Kurose, P. Shenoy and D. Towsley, *Multimedia Streaming via TCP: An analytic Performance study*, ACM Trans. Multimedia Computing Communications and Applications, vol. 4, no. 2, 2008.

[3] A. Balk, M. Gerla, D. Maggiorini and M. Sanadidi, *Adaptive video streaming: pre-encoded MPEG-4 with bandwidth scaling*, Computer Networks, vol. 44, no. 4, pp. 415-439, Mar. 2004.

[4] L. Cai, X. Shen, J. Pan and J. W. Mark, *Performance Analysis of TCP-Friendly AIMD Algorithm fo Multimedia Applications*, IEEE Trans. on Multimedia, vol. 7, no. 2, pp. 339-355, Apr. 2005.

[5] D. Bansal and H. Balakrishnan, *Binomial congestion control algorithm*, in IEEE INFOCOM, 2001.

[6] H. Shiang and M. van der Schaar *Media-TCP: A Quality-Centric TCP-Friendly Congestion Control for Multimedia Transmission*, accepted for publication in IEEE Trans. on Multimedia.

[7] S. Tasaka and Y. Ishibashi, *Mutually compensatory property of multimedia QoS*, In Proc. ICC, 2002.

[8] W. Wu, et al., *Quality of experience in distributed interactive multimedia environments: Toward a theoretical framework*, In Proc. ACM Multimedia, 2009.

[9] ITU-T Recommendation P.911, *Subjective audiovisual quality assessment methods for multimedia applications*, 1998.

[10] R. Rejaie, M. Handley and D. Estrin, *RAP: An End-to-End Rate-based Congestion Control Mechanisme For Real-time Stream in Internet*, in INFOCOM, 1999.

[11] ITU-T, *Definition of Quality of Experience (QoE)*, Reference: TD 109rev2(PLEN/12).

[12] S. Thakolsri, W. Kellerer and E. Steinbach, *QoE-Based Cross-Layer Optimization of Wireless Video with Unperceivable Temporal Video Quality Fluctuation*, 2011 IEEE International Conference on Communications (ICC), vol. 5, no. 9, 2011.

[13] K. Piamrat, A. Ksentini, C. Viho and J. M. Bonnin, *QoE-Aware Admission Control for Multimedia Applications in IEEE 802.11 Wireless Networks*, IEEE 68th Vehicular Technology Conference, vol. 21, no. 24, pp. 1-5, Sept. 2008.

[14] A. B. Reis, J. Chakareski, A. Kassler and S. Sargento, *Quality of experience optimized scheduling in multi-service wireless mesh networks*, 17th IEEE International Conference on Image Processing (ICIP), vol. 26, no. 29, pp. 3233-3236, 2010.

[15] M. Michel, S. Agarwal, W. Kellerer and A. Feldmann, *Toward QoE-Aware Optimum Peer Cache Sizes for P2P Video-on-Demand Systems*, IEEE International Conference on Communications (ICC), vol. 23, no. 27, pp. 1-5, May 2010.

[16] E. Ibarrola, F. liberal, I. Taboada and R. Ortega, *Web QoE Evaluation in Multi-agent Networks: Validation of ITU-T G.1030*, Fifth International Conference on Autonomic and Autonomous Systems, pp. 289-294, 2009.

[17] ITU-T Recommendation P.562, *Telephone transmission quality, telephone installations, local line networks.*

[18] ITU-T, *The E-model, a computational model for use in transmission planning*, Feb. 2003.

[19] Microsoft Lync: http://lync.microsoft.com/

[20] Office Communications Server 2007 Quality of Experience Monitoring Server Guide, http://technet.microsoft.com/en-us/library/dd627288(office.12).aspx

[21] S. Khan, S. Duhovnikov, E. Steinbach and W. Kellerer, *MOS-based*

multiuser multiapplication cross-layer optimization for mobile multimedia communication, Advances in Multimedia, 2007.

[22] C. P. Fu, S.C. Liew, *TCP Veno: TCP enhancement for transmission over wireless access networks*. JSAC, 2003.

[23] R. S. Sutton, *Generalization in reinforcement learning: Successful examples using sparse coarse coding*, In Advances in Neural Information Processing Systems 8, pages 1038-1045, MIT Press, 1996.

[24] G. A. Rummery and M. Niranjan, *Online Q-learning using connectionist systems*, Tech. Rep. CUED/F-INFENG/TR166, Cambridge.

[25] R. D. Smallwood and E. J. Sondik, *The optimal control of partially observable Markov decision processes over a finite horizon*, Operations Research, vol. 21, pp. 1071-1088, 1973.

[26] J. Padhye, V. Firoiu, D. F. Towsley and J. F. Kurose, *A Model Based TCP-Friendly Rate Control Protocol*, In Proc. of NOSSDAV-99, 1999.

[27] S. Floyd, M. Handley, J. Padhye, J. Widmer, *Equation-Based Congestion Control for Unicast Applications*, ACM SIGCOMM Computer Communication Review, vol. 30, no. 4, pp. 43-56, Oct. 2000.

[28] 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.

[29] S. M. Ross, *Stochastic Processes*, John Wiley and Sons, 1996.

Oussama Habachi received the engineering degree in computer sciences from the National School of Computer Sciences, Manouba, Tunisia, in 2008, and the M. Sc. degree from the University of Pierre and Marie Curie, Paris, France, in 2009. He is currently working toward the Ph.D. degree in the University of Avignon, France. The topic of his P.hD. is about application of game theory in wireless networks, specifically in cognitive radio networks.

Yusuo Hu received his B.S. degree in Automation from University of Science and Technology of China in 1999. He received his Ph.D. degree in Automation from University of Science and Technology of China in 2005. He joined Microsoft Research Asia since 2005 and is now an associate researcher.

Mihaela van der Schaar is Chancellor's Professor of Electrical Engineering at University of California, Los Angeles. 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 is an IEEE Fellow, a Distinguished Lecturer of the Communications Society for 2011-2012, the Editor in Chief of 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 (2004), the Best Paper Award from IEEE Transactions on Circuits and Systems for Video Technology (2005), the Okawa Foundation Award (2006), the IBM Faculty Award (2005, 2007, 2008), the Most Cited Paper Award from EURASIP: Image Communications Journal (2006), the Gamenets Conference Best Paper Award (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, and holds 33 granted US patents. For more information about her research visit: http://medianetlab.ee.ucla.edu/

Yezekael Hayel received his M.Sc. in Computer Science and Applied Mathematics from the University of Rennes 1, in 2002. He had a Ph.D. in Computer Science from University of Rennes 1 and INRIA in 2005. Since 2006, he has been an assistant professor at the University of Avignon, France. His research interests include performance evaluation of networks based on game theoretic and queueing models. He looks at applications in communication networks like: 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, Yezekael Hayel has participated in several projects. He was also involved in workshops and conference organization (WNC3 2008, GameComm 2009 and Bionetics 2009). He participates in several national (ANR) and international projects (European, cefipra, etc) with industrial companies like Orange Labs, Alcatel-Lucent, IBM and academic partners like Suplec, CNRS and UCLA.He has been invited to give seminal talks in institutions like INRIA, Suplec, UAM (Mexico), ALU (Shanghai). More informations can be found at http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/hayel/acceuil.htm

Feng Wu received his B.S. degree in Electrical Engineering from XIDIAN University in 1992. He received his M.S. and Ph.D. degrees in Computer Science from Harbin Institute of Technology in 1996 and 1999, respectively. He is now a Senior Researcher/Research Manager in Microsoft Research. His research interests include image and video representation, media compression and communication, as well as, computer vision and graphics. He has authored or co-authored over 200 papers published in journals like IEEE Transaction on Image Processing (T-IP), IEEE Transactions on Circuits and System for Video Technology (T-CSVT), IEEE Transactions on Signal Processing (T-SP), IEEE Transactions on Visualization and Computer Graphics (T-VCG), IEEE Transactions on Multimedia (T-MM) and some other International Conferences and Forums, e.g., MOBICOM, SIGIR, INFOCOM, CVPR, DCC, and ICIP. As a co-author, he got the best paper award in IEEE T-CSVT 2009, PCM 2008 and SPIE VCIP 2007. Wu has been a senior member of IEEE. He serves as an associate editor for IEEE Transactiosn on Circuits and Systems for Video Technology and IEEE Transactions on Multimedia. He also serves as TPC chair in MMSP 2011, VCIP 2010 and PCM 2009, TPC track chair in ICME 2012, ICME 2011 and ICME 2009 and Special sessions chair in ICME 2010.