A Game Theoretical Framework for Token-based Adaptive Video Streaming

Federico Chiariotti, Giovanni Pilon, and Leonardo Badia
Dept. of Information Engineering, University of Padova, via Gradenigo 6B, 35131 Padova, Italy
email: {chiariot, pilongio, badia}@dei.unipd.it

***Abstract*—We consider a multi-stage Bayesian game to model the interaction between an adaptive video streaming client and a congested network adopting a token-based policy for QoS provisioning. The Bayesian type of the network is its level of congestion, which is initially unknown to the client, but heavily influences its payoff. For this reason, the client may be interested in iteratively estimating such a type as the game goes on. To this end, the game is solved to find the equilibria solution, and the estimation process performed by the client is simulated. We discuss how the initial conditions can gauge the convergence speed of the estimate. We find out that, while the network type may be sometimes hard to estimate, especially in low congestion scenarios, nevertheless the equilibrium action of the client is still very close to the ideal best response with full knowledge of the network type. We extend this result to the ability of the client to even correctly estimate the prior distribution of the network type from multi-stage streaming games.**

***Index Terms*—Multimedia communication; Streaming media; Quality of service; Game theory; Bayesian games.**

I. Introduction

VIDEO streaming services are rapidly becoming the main source of traffic on the Internet, as well as an essential service for mobile and computer users. According to [1], video traffic was 60% of total consumer traffic in 2013, and a further increase to up to 75% is expected for the next few years.

Multimedia streaming is usually described as having strict Quality of Service (QoS) requirements on both error rate and delay, while having an extremely high bit rate. These requirements get even more taxing for high definition video, and can become a heavy burden for congested wireless networks, in which the available data rates are limited [2], [3].

A possible solution to this problem is dynamic video code rate adaptation: the Quality of Experience (QoE) loss when viewing a compressed video is both milder and more controllable than the one caused by random packet errors or high network delay. The streaming client can then request different qualities basing on its assessment of the network conditions, fully exploiting the available bandwidth and maximizing user QoE [4], [5]. However, when video streaming start, the network conditions are not known by the client, thus making it difficult to adapt the request, and the resulting "barganing" of resources with multimedia users [6] can be inefficient.

In this paper, we propose to study such an interaction within a game theoretical framework. In the recent scientific literature, game theory has often been applied to networking; in many cases, the focus is on guaranteeing QoS and fairness among multiple users accessing the same constrained resource. For example, [7] considers multiple video streams in an orthogonal frequency division multi-access (OFDMA) downlink; game theory is used to accommodate the request of the users and fairly allocate bandwidth among them. Along similar lines, papers [8] and [9] adopt a Nash bargaining solution, for a general network and for the specific case of an OFDMA downlink, respectively, once again to maximize fairness between multiple video users. Finally, in [10], with a slightly different approach, game theory is applied to video compression, optimizing the bit rate control via an estimation of the coding complexity of frames; but game theory is invoked again to obtain fair coordination of multiple users. Remarkably, in these contributions, a perfect information assumption is often made, which means to deal with users that are fully aware of the game's payoffs, i.e., they perfectly know the network conditions.

Differently from these previous works, our model does not investigate multiple access management, which we assume to be left to the network controller. Instead, we consider the individual allocation of resources to a single user, and therefore our game involves just the user and the network management as two players. Also, the novelty of our approach is that we consider that the user is unaware of the network condition, and we resort to Bayesian games to capture the imperfect information of this game.

In more detail, we consider a wireless network implementing QoS provisions through a token-bucket method [7] as the practical mechanism to coordinate the users. A strategic user of the network, identified as player 1 in our framework, is requesting the allocation of some resources for video transmission, driven by a utility function describing its perceived QoE. The network as a whole is the other agent in the game, player 2, and reacts to this request according to its own utility function, influenced by the congestion status [2], which is known to player 2 (but not to player 1) and therefore represents the *type* of this player.

The result is a multistage Bayesian game, in which player 1 requests video packets with different rates and chooses token payments dynamically, gradually adapting to the information it receives about network congestion. An advantage of the game theoretical framework is that there is no explicit signaling between the network and the video streaming client, but the latter infers the congestion from the utility it gets after each stage. Remarkably, we are able to show that the estimation

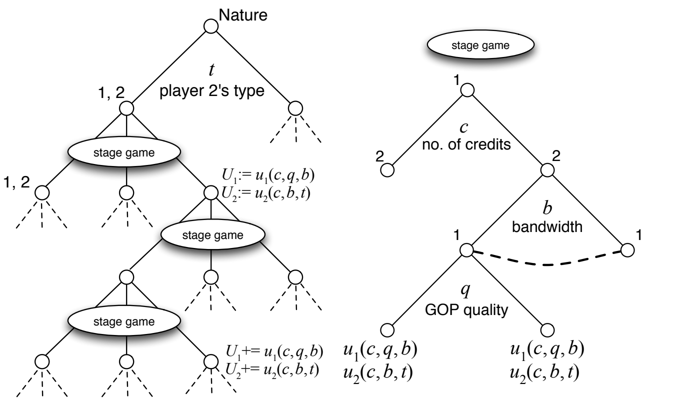

Fig. 1. Structure of the game. The stage game is reported to the right.

of player 2's type is not always successful (it is especially difficult when the network load is low); however, player 1's action almost always converges quite rapidly to the ideal allocation request. Indeed, when the type of player 2 is correctly estimated, this is automatic, whereas in the case of incorrect estimation but low network load, the allocation is often successful. Thus, our game theoretical setup proves that an iteratively updated mechanism of user requests can be suitable to achieve an efficient multimedia allocation in wireless networks.

The rest of this paper is organized as follows. In Section II we introduce our game-theoretical characterization of the network allocation problem for video streaming. We further discuss the solution of the game, and how it can be reached through iteration across game repetitions; finally, we extend the analysis to the type estimation of the "network" player. Section III reports some numerical results for the convergence of the multi-stage game to the correct estimate of the types and the optimal action. Finally, we conclude in Section IV.

II. Game-theoretical model

We consider a video streaming scenario as a multistage two-player game; players 1 and 2 are the streaming client and the network, respectively. Each stage game represents the quality setup for the transmission of a Group Of Pictures (GOP); the client can choose the video quality dynamically for each GOP. In addition, the network implements a token-bucket scheme [7] to ensure fairness among the users. With such a mechanism, the user can receive high quality GOPs, but it needs to pay a higher price (more tokens) for it.

The perceived QoE of a GOP, and therefore the entire game development, depend on the network congestion [**?**], [2]. This is represented as a *type*, denoted as $t \in [0, 1]$ (where 0 means no congestion and 1 is the highest congestion level), of player 2. Thus, the scenario is framed as a Bayesian game [12]. While the actual value of t is known only to player 2, its prior pdf, written as $f_0(t)$, is common knowledge for both players; we assume it is a Gaussian truncated between 0 and 1, with average μ_t and variance σ_t^2.

Each stage game is organized as follows (see Fig. 1): first, player 1 chooses a price for the channel use, paying an integer number of tokens c between 0 and 10. Then, the two players

TABLE I
Parameters of the utility function.

q	k_0	k_1	k_2	k_3
H	50	1	5	16
M	50	2/3	8	14
L	50	1/3	12	14

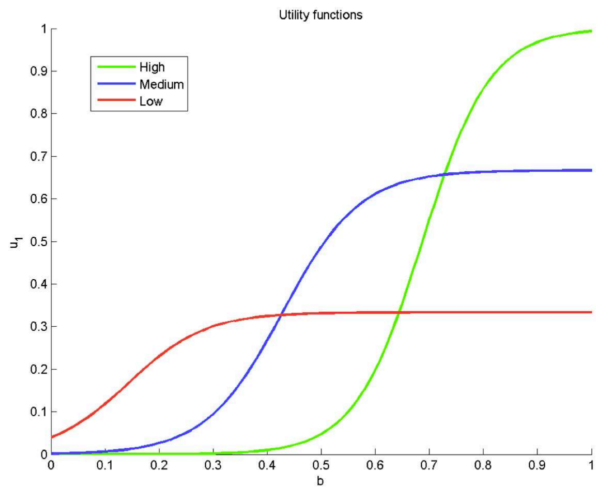

Fig. 2. Function u_1 for the three quality values (with no noise, $c = 5$)

move simultaneously, as denoted by the dashed line: player 2 chooses how much bandwidth b to allocate to the client, based on its own type and player 1's choice of c. At the same time, player 1 chooses the quality q for the next GOP, with three possible choices: high (H), medium (M), and low (L); we assume that these correspond to three available compression levels of the GOP through scalable encoding [4]. After each stage game, stage utilities are computed and cumulated with the previously accrued utilities.

The variables involved are summarized as follows

$$\begin{aligned} f_0(t) \sim N_{[0,1]}(\mu_t, \sigma_t^2) \qquad & c \in \{0, 1, \ldots, 10\} \\ q \in \{H, M, L\} \qquad & b \in [0, 1] \end{aligned} \tag{1}$$

The stage utilities are instead computed as follows

$$u_1(c, q, b) = \frac{k_1(q)}{1 + e^{-k_2(q) + k_3(q) b}} - \left(\frac{c-5}{k_0}\right) + w \tag{2}$$

$$u_2(c, b, t) = 1 - \left(1 + \frac{c-5}{k_0} - t - b\right)^2 \tag{3}$$

where $w \sim N(0, \sigma_w^2)$ is a superimposed noise term that keeps into account video quality fluctuations. Parameters k_1, k_2, and k_3 represent the dependence of u_2 on the chosen quality q, while k_0 is actually a fixed term. The values chosen for our numerical setup are reported in Table I.

The client utility described in (2) and shown in Fig. 2 is a sigmoid function; video Quality of Experience (QoE) is often modeled with sigmoid functions in the literature [13]. The perceived QoE may vary due to random parameters due to the channel characteristics (such as fading in wireless channels) or of the video itself (such as the current GOP's dynamic content), which is the reason for the noise term w.

As (3) shows, player 2's utility function does not depend on q: we assume that the network is large enough not to

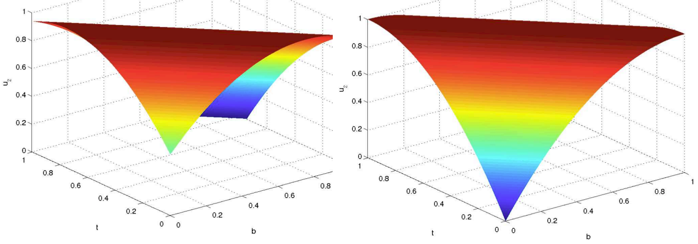

Fig. 3. Function u_2 for all possible moves b, $c = 0$ (left) and $c = 5$ (right)

be perturbed by a single video stream. As the default GOP cost is assumed to be 5 tokens, if player 1 pays that amount player 2 simply allocates the available bandwidth to the client, whose amount depends on the network congestion, i.e., on t; in other words, b is simply set to $1 - t$. If player 1 is willing to pay $c > 5$ tokens, the network temporarily redistributes the load to provide a larger bandwidth, while if it pays fewer tokens than 5, it will receive less resource accordingly. The fact that c cannot be increased indefinitely is reflected in its direct influence on u_1; the higher c, the lower u_1. Fig. 3 shows the network utility as a function of b and t for two different values of c.

A. Solution of a game stage

Within a given stage, the behavior of player 2 only depends on c and t, which are known to it when it makes its move: we can then derive a closed-form equation for the best move s_2^*, corresponding to the selection of the most suitable b. If the best move is out of the interval $[0, 1]$, player 2 will play the closest admissible move.

$$s_2^*(c, t) = \underset{b}{\text{argmax}}\, u_2(c, b, t) = 1 - t + \frac{c - 5}{k_0} \quad (4)$$

The best move of player 1 is instead

$$s_1^*(t) = \underset{c,q}{\text{argmax}}\, u_1\big(c, q, s_2^*(c, t)\big)\ , \quad (5)$$

where maximization is more complicated, as it involves both choices of c and q. However, since c and q have finitely many values, the solution can simply be found by enumeration.

Now, as strategy s_1^* is a best response to s_2^*, and knowing that s_2^* is a best response to any strategy by player 1, a hypothetical perfect information game would have (s_1^*, s_2^*) as a subgame-perfect equilibrium (SPE) for each stage [12]. The uncertainty on the type of player 2 makes 1's move sub-optimal: in the first stage, player 1 has no information on player 2's type, and it chooses to play the Bayesian Nash Equilibrium (BNE):

$$s_1^* = \underset{c,q}{\text{argmax}} \int_0^1 u_1^*\big(c, q, s_2^*(c, \theta)\big) f_0(\theta) \mathrm{d}\theta \quad (6)$$

Each stage of the game can also be regarded a signaling game, as player 2 reveals some information about its type t with its move. The estimate $\hat{t}_i$ is not perfect because of the noise, but s_2^* would be a separating strategy in a noise-free scenario as $u_2(c, b, t)$ is fully invertible.

Thus, player 1 can estimate b and t as

$$\hat{b} = \frac{k_2(q) + \log\left(\dfrac{k_1(q)}{u_1 + (c-5)/k_0} - 1\right)}{k_3(q)} \quad (7)$$

$$\hat{t} = 1 + \frac{c - 5}{k_0} - \hat{b} \quad (8)$$

where it is worthwhile noting that $\hat{b} = \hat{b}(q, u_1)$, i.e., the estimate of b (and consequently also that of t) depends on u_1, i.e., the partial utility perceived by player 1 in the present stage. After a sequence of n further refined estimates of the type $\hat{\mathbf{t}}_n = (\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_n)$, we can write an updated estimate of the prior after n estimations as $f_n(t)$, for which it holds

$$f_n(t) = P p(\hat{\mathbf{t}}_n | t) f_0(t) \quad (9)$$

$$f_n(t) = P \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left[\frac{-\xi}{2\sigma_w^2}\right] f_{n-1}(t) \quad (10)$$

$$\text{with } \xi = \Big(u_1\big(c, q, s_2^*(c, t)\big) - u_1\big(c, q, s_2^*(c, \hat{t}_n)\big)\Big)^2$$

where $p(\hat{\mathbf{t}}_n | t)$ is the probability of getting $\hat{\mathbf{t}}_n$ conditioned on the true type t. From (9) we can obtain the recursive formula (10), which can be applied in practice; this is how player 1 can update its beliefs about the pdf of t to find the optimum a posteriori distribution. Note that the normalization constant P in (9) and (10) serves to scale the pdf so that $\int_0^1 f_n(t)dt = 1$. Therefore, the BNE strategy for stage n is

$$s_1^* = \underset{c,q}{\text{argmax}} \int_0^1 u_1^*\big(c, q, s_2^*(c, t)\big) f_n(t) dt \quad (11)$$

We expect that player 1's beliefs about player 2's type will converge to the real value if the value of σ_w^2 is small; it is also possible to estimate the noise variance σ_w^2 along with the type, but the convergence will be slower. Also, note that it may not be important that $\hat{t}$ converges exactly to the real type of player 2 as long as the estimate still allows player 1 to play through (11) a strategy that is close enough to the real best move, see (5). We will see that this is actually the case in many situations where t is incorrectly estimated, but s_1^* is still adequate.

B. Multi-stage game and prior estimation

After deriving the optimal behavior for the two players in a video streaming game, we can expand the model to perform a prior estimation: if player 1 knows that $f_0(t)$ is a truncated Gaussian distribution but not its parameters μ_t and σ_t^2, it can repeat the video streaming game several times in independent network conditions to obtain several reliable values of t drawn from the distribution.

After getting n estimates of the network type t as $\hat{\mathbf{t}}_n = (\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_n)$, we can derive the prior parameters by using the mean and variance unbiased estimators [14]

$$\hat{m_t} = \frac{1}{n} \sum_{i=1}^{n} \hat{t}_i \quad (12)$$

$$\hat{v_t}^2 = \frac{1}{n-1} \sum_{i=1}^{n} |\hat{t}_i - \hat{\mu}_i|^2 \quad (13)$$

so that we can estimate the parameters of the truncated Gaussian distribution as $\hat{\mu}_n$ and $\hat{\sigma}_n$ computed as

$$\hat{\mu}_n = \hat{m}_t + \frac{\phi\left(-\frac{\hat{m}_t}{\hat{v}_t}\right) - \phi\left(\frac{1-\hat{m}_t}{\hat{v}_t}\right)}{\Phi\left(\frac{1-\hat{m}_t}{\hat{v}_t}\right) - \Phi\left(-\frac{\hat{m}_t}{\hat{v}_t}\right)}\hat{v}_t \tag{14}$$

$$\hat{\sigma}_n^2 = \hat{v}_t^2 \left(1 + \frac{-\frac{\hat{m}_t}{\hat{v}_t}\phi\left(-\frac{\hat{m}_t}{\hat{v}_t}\right) - \frac{1-\hat{m}_t}{\hat{v}_t}\phi\left(\frac{1-\hat{m}_t}{\hat{v}_t}\right)}{\Phi\left(\frac{1-\hat{m}_t}{\hat{v}_t}\right) - \Phi\left(-\frac{\hat{m}_t}{\hat{v}_t}\right)}\right) - \hat{\mu_n}^2 \tag{15}$$

where $\phi(x)$ is a standard Gaussian pdf and $\Phi(x)$ is a standard Gaussian cdf. The complexity of the formulas makes it hard in some cases to accurately gauge the mean and variance of the truncated Gaussian, as small variations in the sample mean and variance, as well as errors in the estimates, may cause noticeable errors in the resulting mean and variance.

III. Numerical Results

We evaluate the proposed game theoretical framework via independent simulations implemented in Matlab. The type of player 2 was quantized with a 0.01 step in order to have repeatable results and reduce computation times; the formulas only need slight adaptations for this discrete case. In the following, we report the results for the proposed iterative procedure, focusing on its convergence of both the estimation of player 2's type and the best action of player 1. Moreover, we show how the same procedure can be used to gauge the estimation of the prior distribution.

A. Performance metrics

As getting a certain estimate of player 2's type is extremely hard in a noisy scenario, and meaningless in the continuous case, we had to define meaningful performance metrics; we measured the speed of the convergence of the estimate as well as the quality of the moves of player 1.

We can define the metric Δu as the distance between the maximum utility and the one resulting from the actual move (without noise):

$$\Delta u = u_1\big(s_1^*(t), s_2^*(c^*, t)\big) - u_1\big(c, q, s_2^*(c, t)\big) \tag{16}$$

We can see the evolution of Δu to fathom how fast the moves of player 1 converge to the optimum, and if the payoff difference is significant.

Player 1 does not estimate player 2's type directly, but it maintains the best possible estimate of $f_0(t)$, using the information from the previous rounds in order to keep playing the Bayesian best response in each stage. We define three convergence criteria to measure the number of stages it takes for the estimate to be accurate enough; all three consider a region around the true value, and consider the estimate accurate if the probability of being inside that region is larger than a chosen value. Formally, convergence is declared to be reached whenever

$$\text{Prob}[t - t_{\text{th}} \leq t_e \leq t + t_{\text{th}}] \geq P_0 \tag{17}$$

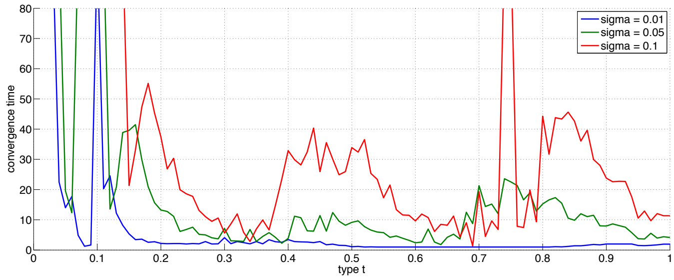

Fig. 4. Average convergence time (criterion 1) for all types

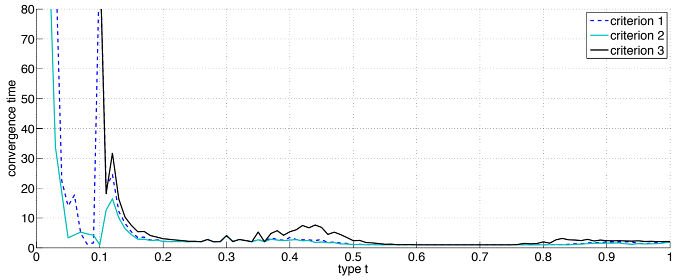

Fig. 5. Average convergence time for all criteria, $\sigma_w = 0.01$

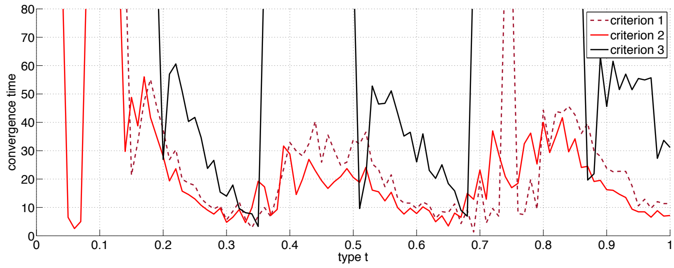

Fig. 6. Average convergence time for all criteria, $\sigma_w = 0.1$

We set $t_{\text{th}} = 0.05$ and $P_0 = 0.9$ for criterion 1, $t_{\text{th}} = 0.05$ and $P_0 = 0.8$ for criterion 2 and $t_{\text{th}} = 0.02$ and $P_0 = 0.9$ for criterion 3. Criterion 3 is the strictest, as it requires that the estimated type fall inside a narrow region with very high probability, while both the other criteria use a larger acceptable region.

B. Type estimate convergence

We ran two simulation campaigns to evaluate the convergence time of the type estimate: in the former, we ran the game several times for every type, plotting the average convergence time for each type, while in the latter we used a Monte Carlo approach to plot a histogram of convergence times.

Fig. 4 shows that, as the noise on player 1's utility increases, it becomes more and more difficult for it to ascertain player 2's type. The effect of the noise is particularly strong when $t < 0.2$, i.e., the network is free of congestion; the flatness of the utility function in this area makes it hard to distinguish between types even with low levels of noise. The convergence times increase significantly already when $\sigma_w = 0.1$, while for $\sigma_w = 0.01$ convergence is reached in less than 10 stages for almost all steps.

In Figs. 5 and 6 we can see that convergence times increase when using a stricter convergence rule such as criterion 3); for $\sigma = 0.01$ the overlap is almost perfect, with the type estimate

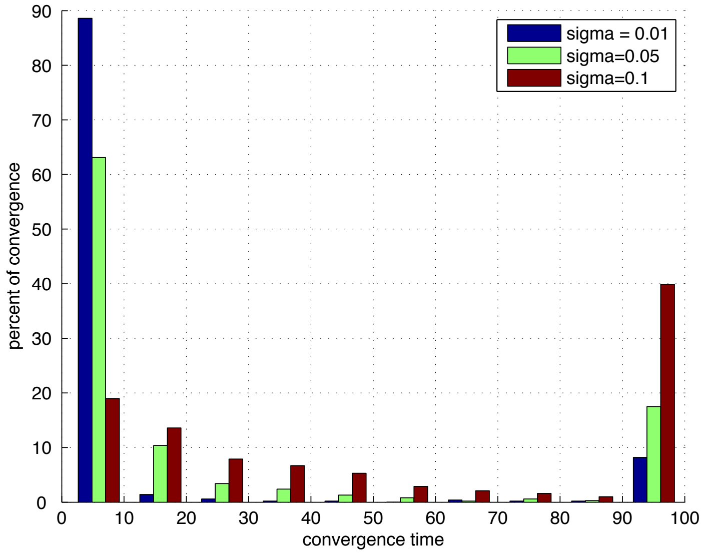

Fig. 7. Histogram of convergence times in the Monte Carlo simulation

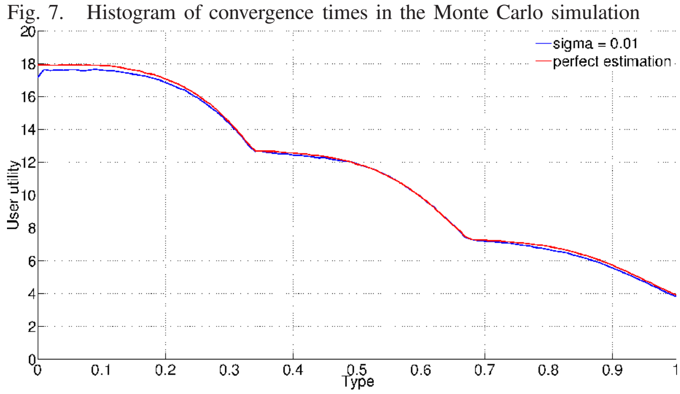

Fig. 8. Average utility over 20 stages with no noise, compared with $\sigma = 0.01$

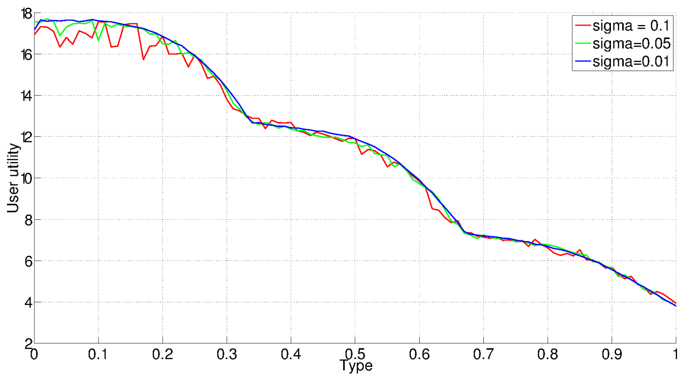

Fig. 9. Average utility over 20 stages for several noise levels

converging in very few stages with all criteria. This means that, for low noise levels, the estimate for $t > 0.1$ is extremely accurate after less than 10 stages.

The results of the Monte Carlo simulation, shown in Fig. 7 (which considers criterion 1), substantially confirm the analysis: convergence is extremely fast when the noise level is low, but slows steadily as noise variance increases. The error probability for low types may pose a problem in getting accurate estimates, but this is due to the structure of the utility functions; nevertheless, what is important is whether

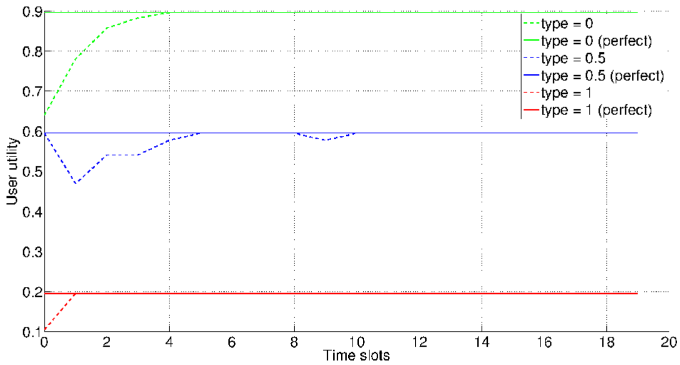

Fig. 10. Utility over time for $\sigma_w = 0.1$

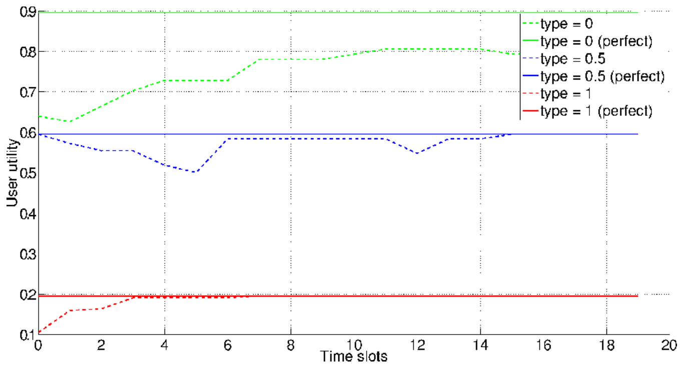

Fig. 11. Utility over time for $\sigma_w = 0.3$

the strategy of player 1 quickly converges to its best move, even when its estimate does not converge to the correct type. If the client really needs to estimate the network type for reasons beyond the simple evaluation of its optimal move, it can actually think of a more exploratory strategy to ascertain whether the network is congested or not.

C. Convergence to the optimal action

Although the type estimation is not always perfect, convergence to the optimal action is extremely fast: Figs. 8 and 9 show that, after 20 stages, the average utility for noise levels up to $\sigma_w = 0.1$ matches the perfect one almost exactly.

The convergence to the optimal action is extremely fast, as Fig. 10 shows; even with a non-negligible level of noise, which makes it hard to estimate the correct type when the reward function is flatter, the client takes the perfect action every time taking only 3 or 4 turns to adjust. We also tested the client in a very difficult situation, raising the noise level to $\sigma_w = 0.3$; the results are displayed in Fig. 11. Such a noise level makes it almost impossible to correctly estimate the type, but the client manages to get close to the optimum in all cases, and even achieve it in less than 10 stages when the network type is 1.

D. Prior estimation

We try using repeated estimates in independent conditions to get an estimate of the prior distribution. We use a low noise level ($\sigma_w = 0.01$) and terminate the estimation process after a finite number of stages (as opposed to the previously defined three criteria), choosing the type with the highest

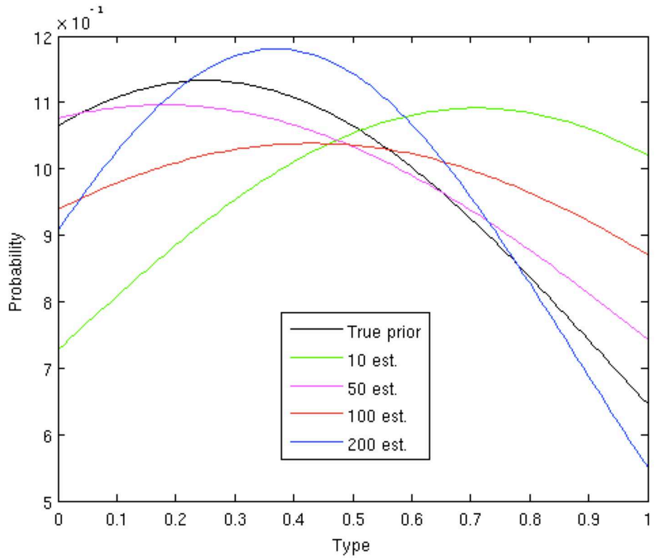

Fig. 12. Prior for $\mu_t = 0.25$, $\sigma_t^2 = 0.5$

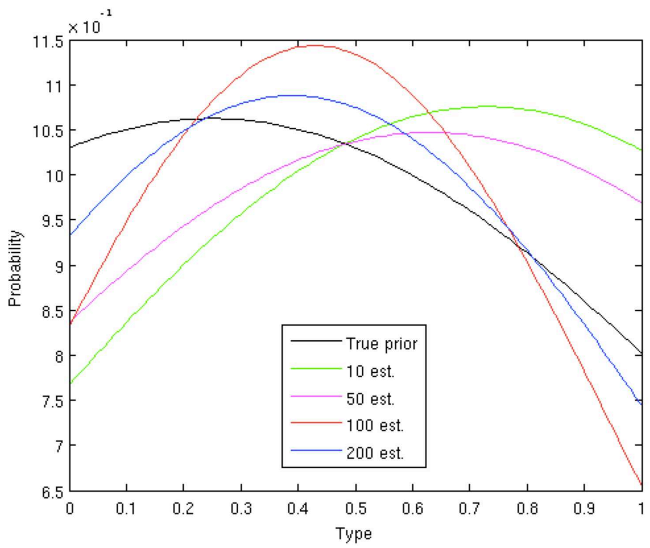

Fig. 13. Prior estimation for $\mu_t = 0.25$, $\sigma_t^2 = 1$

estimated probability; in light of the previous results, we want good estimates to avoid large errors in the sample mean and variance.

The related results are reported in Figs. 12 and 13, for a fixed value of $\mu_t = 0.25$ and $\sigma_t^2 = 0.5$ or $\sigma_t^2 = 1$, respectively. The estimates gave satisfactory results for a range of distributions; the results are even closer to the actual prior for smaller variances, and the larger error in estimating the mean with high variance (Fig. 13) is compensated by the flatter nature of the distribution.

IV. Conclusions and Future Work

In this work, we proposed a game theoretical investigation of interaction between multimedia users and the network management, especially focusing on the uncertainty of the users about the network congestion status. We analyzed the game from several standpoints, to understand the dynamic evolution of the interaction and the estimation process itself. The simulation results show that the convergence to the best action is fast and efficient for reasonable noise levels; while considerably more difficult, the prior estimation also proved satisfactory.

The results showed a strict dependence between convergence speed and the slope of the utility function; a nearly flat reward function results in non-convergent type estimation, as noise overwhelms the signaling (although the client still converges to the optimal action). A possible future line of study could be an analysis of the influence of utility functions on convergence, changing the slopes and using linear functions as a benchmark. Another interesting approach would be to consider a more realistic evaluation of QoE from QoS metrics [11] so as to use real video QoE metrics as the payoff function of the client. We might also compare our prior estimation technique with other, more refined, statistical techniques that can be developed to reduce convergence times and errors in the estimates.

Finally, a more ambitious improvement might consider multiple users; a multitude of smart players contending for network resources would be a challenging model, which could show us if our network model is a good approximation of reality, as well as providing new data.

References

[1] Cisco report “Cisco visual networking index: forecast and methodology: 2013–2018,” Cisco, 2014.

[2] H. Nam, K. H. Kim, D. Calin, and H. Schulzrinne, “Towards dynamic network condition-aware video server selection algorithms over wireless networks,” *Proc. IEEE ISCC*, June 2014.

[3] L. Badia, N. Baldo, M. Levorato, and M. Zorzi, A Markov framework for error control techniques based on selective retransmission in video transmission over wireless channels, *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 488500, Apr. 2010.

[4] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H. 264/AVC standard, *IEEE Trans. Circ. Syst. for Video Techn.*, vol. 17, no. 9, pp. 11031120, 2007.

[5] H. Sohn, H. Yoo, W. D. Neve, C. S. Kim, and Y. M. Ro, “Full-reference video quality metric for fully scalable and mobile SVC content,” *IEEE Trans. on Broadcasting*, vol. 56, no. 3, 2010.

[6] L. Zhengye, W. Zhenyu, L. Pei, L. Hang, W. Yao, “Layer bargaining: multicast layered video over wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 28 , no. 3, pp. 445 - 455, 2010.

[7] M. Iturralde, A. Wei, T. Ali Yahiya, and A. L. Beylot, “Resource allocation for real time services using cooperative game theory and a virtual token mechanism in LTE networks,” *Proc. IEEE CCNC*, pp. 879–883, Jan. 2012.

[8] M. Liu, H. Li, and W. Li, “Smoothing rate control for multiple video streams using game theory,” *Proc. IEEE ISCAS*, pp. 2913–2917, May 2011.

[9] N. Khan, M. G. Martini, and Z. Bharucha, “Quality-aware fair downlink scheduling for scalable video transmission over LTE systems,” *Proc. IEEE SPAWC*, pp. 334-338, Jun. 2012.

[10] I. Ahmad, and J. Luo, “On using game theory to optimize the rate control in video coding,” *IEEE Trans. Circ. Sys. for Vid. Techn.*, vol. 16, no. 2, pp. 209–219, Feb. 2006.

[11] P. Orosz, T. Skopko, Z. Nagy, P. Varga, and L. Gyimothi, “A case study on correlating video QoS and QoE,” *Proc. IEEE NOMS*, May 2014.

[12] M. J. Osborne and A. Rubinstein. *A course in game theory*. MIT press, 1994.

[13] J. W. Lee, R. R. Mazumdar, and N. B. Shroff, “Non-convex optimization and rate control for multi-class services in the Internet,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, pp. 827–840, Apr. 2005.

[14] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. vol. 2, Chapman & Hall/CRC, 2014.