2014 IEEE INFOCOM Workshops: 2014 IEEE INFOCOM Workshop on Communication and Networking Techniques for Contemporary Video

Leveraging Users' Likes in a Video Streaming P2P Platform

L. Natali⋆, C. Barcellona⋆⋆, M.L. Merani⋆
⋆ Dipartimento di Ingegneria "Enzo Ferrari"
University of Modena and Reggio Emilia, Italy
⋆⋆ Dipartimento Energia, Ingegneria dell'Informazione e Modelli Matematici, University of Palermo, Italy
e-mail: laura.natali@unimore.it, cettina.barcellona@unimore.it, marialuisa.merani@unimore.it

***Abstract*—This paper investigates how a p2p television platform can take advantage of the presence of frequent channel viewers to grant them a more satisfying service than to less regular spectators. The idea we explore is to learn beforehand about the users' interests, in order to cluster them in groups that display different behaviors; then, the neighborhood creation strategy and video chunk scheduling algorithm of the overlay is altered, with the aim of serving frequent spectators in a privileged manner, providing them with a faster access to the selected channel without overly penalizing less habitual customers. An analytical model is developed, to capture the difference in startup delay that the proposed modifications introduce; several additional performance metrics are numerically determined, in order to thoroughly size up the performance of both groups of viewers. The obtained results show that a clear service differentiation is achieved and also quantify the effects that alternative neighborhood creation algorithms have on the amount of such gap.**

I. Rationale

In recent years commercial Internet television platforms have earned a tremendous success [1], and the p2p feature within their architecture has undoubtedly played a key role in determining their economical profits [2], [3]. In parallel, the flourishing of plenty of channels with well defined thematic strands has contributed to polarizing the users' choice towards those offerings that exactly reflect their likes.

In this paper we propose a video streaming architecture that leverages the interests of its users, with the aim of guaranteeing a better service to regular spectators than to less frequent viewers of a television channel. A p2p streaming system is examined, based upon a mesh architecture, where both the neighborhood creation algorithm and the scheduling policy adopted to distribute video chunks are modified, so as to grant a privileged service to regular users. An analytical model is newly developed, that allows to better understand the difference in playback delay incurred by the two groups, with the help of simple queueing theory tools. To this regard, it is noteworthy to observe that in literature not many works analyze p2p live video streaming systems from a rigorous theoretical point of view and even fewer employ queueing tools to point out unexplored aspects. A notable example is [4], where the authors model a multichannel P2P live video streaming system through both close and open infinite server queueing networks. [5] employs a queueing model too, but its main purpose is to evaluate the latency associated with file transfer in a p2p content sharing network. [6] puts forth a model for a p2p system, although devised for VoD services; [7] focuses on VoD too, drawing from known results about priority queueing. The current work resorts to queueing theory to model the scheduling adopted by parent peers to provide missing chunks to other nodes and quantifies the effects that the proposed policy has on the delay experienced by the two groups of users. In parallel, the performance that the two groups experience and the level of differentiation achieved are numerically assessed via a simulative study: for doing so, a soft clustering algorithm is employed on a reference data set of web-tv users, to first classify the viewers in two groups, featuring a different level of interest in a popular Italian television channel. The results indicate that for a wide range of system parameters a clear performance demarcation is achievable, and suggests the adoption of the same approach in a multichannel system.

The rest of the paper is organized as follows: section II illustrates the salient features of the p2p streaming overlay and the modifications we made to its functionalities, to reach the desired goal; section III puts forth the model we devised to capture the statistical characterization of the startup delay incurred by frequent and ordinary viewers; Section IV critically comments the numerical results we obtained and Section V concludes the paper.

II. The Interest-Aware Video Streaming System

The live streaming system we consider exhibits a p2p, mesh-based architecture. As in every pull-based solution, the streaming overlay dynamically changes: at any given time, a peer maintains partner relationships with multiple peers, seeking new neighbors to maintain a desired download rate level, replacing neighbors that leave the overlay or that display low upload contributions. The exchange of streaming data is controlled by explicit requests: peers periodically send their neighbors buffer maps, to inform them about the video chunks they own; accordingly, each peer analyzes the buffer maps it receives and on the basis of the missing chunks within its current request window it asks its neighbors for them. Chunks are requested adopting the rarest first policy: among all missing chunks, a peer first requests those that less frequently appear within the buffer maps it received, in order to guarantee a more rapid spread of rare chunks. In turn, a peer receiving video chunk requests from its neighbors answers accordingly to a specific chunk scheduling policy. In our system high interest peers receive a privileged service with respect to peers with a low interest in channel contents. As a matter of fact, when a peer receives video chunk requests from its neighbors, it first serves those coming from high interest peers and, if necessary, allows for some – moderate – queueing of the corresponding chunks waiting for transmission; requests raised by low interest peers can be satisfied only when the high interest queue is empty: when this condition is not met, the chunk requests of low interest peers are lost and do not get satisfied. Also, in order not to excessively penalize low interest users, we modify the neighborhood creation algorithm as follows: a new peer, regardless of being high or low interest, builds its neighborhood choosing peers on the basis of their available bandwidth and interest. Namely, nodes are ranked and chosen as neighbors according to the following function

$$f(B_{up}, I) = B_{up}(t) * (1 + I)\,, \tag{1}$$

where $B_{up}(t)$ is the currently available upload bandwidth and $I = 1$ if the peer belongs to C_H, the high interest cluster, $I = 0$ otherwise. This guarantees that when choosing a neighbor, a high interest node is preferred to a low interest peer if the bandwidth of the former is greater than half the bandwidth of the latter.

In what follows, the merit figures we will examine to assess the performance experienced by high and low interest peers are the following: the *startup delay*, measuring the time it takes from the moment a high/low interest peer requests the video stream until the play-out on its screen begins; the *chunk delivery ratio*, that is, the number of video chunks that meet the playback deadline over the total number of chunks that a peer belonging to either class should receive; the *playback delay*, measuring the interval between the point in time when the generic video chunk is generated at the source and the instant this chunk is rendered at a high/low interest peer.

III. Startup Delay Analysis

The focus of this section is on the startup delay experienced by the user when requesting a new channel, given by the sum of two distinct contributions: the channel setup time and the buffering delay. The former is the time required by the newly incoming peer to build its neighborhood; the latter is the time required to accumulate a given number, say S, of video chunks within the new peer's buffer. In order to capture the effects of the new scheduling we propose, we concentrate on the latter, and first provide a statistical characterization for the buffering delay τ of the system where no service differentiation is present; we then examine the overlay that leverages the knowledge of the peers' high and low interest in the television channel and determine their buffering delays, τ_H and τ_L, respectively.

A. The System with No Interest Differentiation

We adopt the following assumptions for the overlay that has no awareness of the users' interests: given N peers are steadily present in the system, the requests the i-th peer raises for missing chunks are modeled by a Poisson process of rate λ_i, $i = 1, 2, \ldots, N$; such requests are taken care of by other peers, i.e., by the overlay as a whole, that we model as a FCFS (First-Come-First-Serve) single server system: its input rate is $\lambda = \sum_{i=1}^{N} \lambda_i$ and its aggregate service rate is equal to $\mu = B_{up}/E[L]$ chunk/s, where B_{up} indicates the overall upload bandwidth available in the system and $E[L]$ the average chunk size. We further assume that the size of a video chunk is exponentially distributed and place no constraints on the system queue size, so that we can describe our system through the $M/M/1$ model.

Let us position ourselves at its output and observe that each customer leaving it corresponds to a chunk being received by a peer that previously requested it. Next, consider the time between the reception of two consecutive chunks by the same, i-th peer. The departure of the customer that corresponds to the delivery of a chunk to the i-th peer is followed by another departure: however, this does not necessarily represent the delivery of another chunk to the same node. Given peers are identical, this condition will occur with probability $p = \frac{\lambda_i}{\lambda} = 1/N$; on the contrary, with probability $(1-p)p = (1-\frac{1}{N})\frac{1}{N}$ two interdeparture times will elapse before another chunk will be received by the i-th peer; in general, with probability $(1-p)^{k-1}p$, k interdeparture times, $k = 1, 2, \ldots$, will go by before the generic i-th peer will receive another chunk. Recalling that the Laplace transform $D^*(s)$ of the probability density function (p.d.f.) of the $M/M/1$ interdeparture time is $D^*(s) = \frac{\lambda}{\lambda+s}$, it is possible to write the Laplace transform of the p.d.f. of the time d_i that elapses between the reception of two consecutive chunks by the same, i-th peer as

$$\begin{aligned} A^*(s) &= \sum_{k=1}^{\infty} D^*(s)^k \, (1-p)^{k-1} \, p = \\ &= \sum_{k=1}^{\infty} \left(\frac{\lambda}{\lambda+s}\right)^k (1-p)^{k-1} \, p\,, \end{aligned} \tag{2}$$

that after a few algebraic passages turns into

$$A^*(s) = \frac{p\lambda}{p\lambda + s} = \frac{\lambda_i}{\lambda_i + s}\,. \tag{3}$$

Let us next determine the time τ_2 required by the i-th peer to collect $S = 2$ chunks, and then generalize such result for an arbitrary value of S. Observe that τ_2 is the sum of two random variables: the first is τ_1, the time spent in system by a generic M/M/1 customer (waiting plus service time); the second is exactly d_i. In the M/M/1 system τ_1 is exponentially distributed with mean

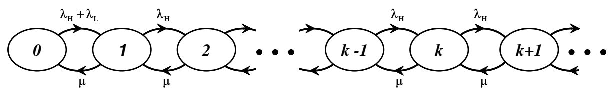

Fig. 1. State transition diagram of the loss system under consideration

$1/(\mu(1-\rho))$, $\rho = \frac{\sum_{i=1}^{N}\lambda_i}{\mu}$, $\rho < 1$ [8], so that, making use of (3), the Laplace transform of τ_2 p.d.f., $T_2^*(s)$, can be readily written as:

$$T^*_{(2)}(s) = \frac{\mu(1-\rho)}{\mu(1-\rho)+s} \cdot A^*(s) = \frac{\mu(1-\rho)}{\mu(1-\rho)+s} \cdot \left(\frac{\lambda_i}{\lambda_i + s}\right). \tag{4}$$

In general, if the condition to meet is that S chunks have to be received, then the Laplace transform $T^*(s)$ of the p.d.f. of the time τ it takes is

$$T^*(s) = \frac{\mu(1-\rho)}{\mu(1-\rho)+s} \cdot \left(\frac{\lambda_i}{\lambda_i + s}\right)^{S-1}, \tag{5}$$

its mean value being

$$E[\tau] = \frac{1}{\mu(1-\rho)} + (S-1)\frac{1}{\lambda_i} \tag{6}$$

and its variance

$$\sigma_\tau^2 = \frac{1}{\mu^2(1-\rho)^2} + (S-1)\frac{1}{\lambda_i^2}. \tag{7}$$

B. The System with High and Low Interest Users

Next, let us consider the overlay where two classes of peers are present: high class peers that have a strong interest in the content distributed within the overlay and low class peers that do not. Their aggregate rate of chunk requests is still Poisson, with rate λ_H and λ_L, respectively, $\lambda_H + \lambda_L = \lambda$; further, the size of their population is N_H and N_L, $N_H + N_L = N$. As before, the overlay is modeled as a single server queueing system with exponentially distributed service times at rate μ: however, to differentiate between classes, a customer corresponding to a chunk request raised by a high interest peer is allowed to queue up when the server is busy; on the contrary, a customer representing a chunk request originated by a low interest peer will get served only if the server is idle, whereas if the server is busy, the customer will get blocked, not being allowed to wait in queue. The state transition diagram of the new system is reported in Fig.1, that readily allows to determine the blocking probability that low interest customers will suffer, given by

$$p_0 = \frac{1 - \frac{\lambda_H}{\mu}}{1 - \frac{\lambda_H}{\mu}(1-R)}, \tag{8}$$

with $R = 1 + \frac{\lambda_L}{\lambda_H}$, as well as its steady-state probabilities:

$$p_k = \Pr[\text{k users in system}] =$$
$$= p_0 \left(\frac{\lambda_H+\lambda_L}{\mu}\right)\left(\frac{\lambda_H}{\mu}\right)^{k-1},\ k = 1, 2, \ldots,$$

We cannot take advantage of Burke's theorem in this new context; Appendix A however shows that the Laplace transform $\Phi^*(s)$ of the p.d.f. of the generic interdeparture time is given by:

$$\Phi^*(s) = \left(\frac{\mu}{\mu+s}\right)(1-p_0) + \left(\frac{\mu}{\mu+s}\right)\cdot\left(\frac{\lambda}{\lambda+s}\right)p_0. \tag{9}$$

A reasoning similar to the one applied to the system with no class differentiation leads to the conclusion that the Laplace transform of the p.d.f. for the time that elapses between the delivery of two chunks to the same, low interest customer is given by

$$A_L^*(s) = \sum_{k=1}^{\infty} \Phi^*(s)^k (1-p_L)^{k-1} p_L \tag{10}$$

where p_L indicates the probability that the departing customer corresponds to a chunk delivered to a specific, low interest peer in the real system,

$$p_L = \frac{\lambda_L p_0}{\lambda_H + \lambda_L p_0} \cdot \frac{1}{N_L}. \tag{11}$$

The Laplace transform of the p.d.f. of the time τ_L it takes a low interest customer to receive S chunks is therefore

$$T_L^*(s) = S_L^*(s) \cdot A_L^*(s)^{S-1}, \tag{12}$$

where $S_L^*(s)$ indicates the Laplace transform of the time t_{s_L} spent in system by a low interest customer, exponentially distributed with parameter μ: see Appendix B for a justification of it. Accordingly, τ_L mean value is given by

$$E[\tau_L] = \frac{1}{\mu} + (S-1)\cdot\left(\frac{1}{\mu p_L} + \frac{p_0}{\lambda p_L}\right) \tag{13}$$

and its variance by

$$\sigma_{\tau_L}^2 = \frac{1}{\mu^2} + (S-1)\cdot$$
$$\cdot\left[\left(\frac{2p_0(\mu(1-p_0)-\lambda)}{\mu p_L \lambda^2}\right) + \left(\frac{\lambda + p_0\mu}{\mu p_L \lambda}\right)^2\right]. \tag{14}$$

Following an analogous approach, the Laplace transform of the p.d.f. of the time a generic, high interest peer requires to receive S chunks is

$$T_H^*(s) = S_H^*(s) \cdot A_H^*(s)^{S-1}, \tag{15}$$

where $S_H^*(s)$ is the Laplace transform of the time t_{s_H} spent in system by a high interest customer and $A_H^*(s)$ is the Laplace transform of the p.d.f. of the time that elapses between the delivery of two chunks to the same, high interest customer: its expression is derived from (10) replacing subscript L with H and p_L with $p_H = \frac{\lambda_H}{\lambda_H + \lambda_L p_0}\frac{1}{N_H}$. The determination of $S_H^*(s)$ is reported in Appendix B and leads to the following expressions for the mean and the variance of t_{s_H}:

$$E[t_{s_H}] = \frac{c_1}{\mu} + \frac{c_2}{\mu - \lambda_H}, \tag{16}$$

with

$$c_1 = p_0 \left(1 - \frac{\lambda_H + \lambda_L}{\lambda_H}\right) \ , \ c_2 = p_0 \frac{\lambda_H + \lambda_L}{\lambda_H} \cdot \frac{\mu}{\mu - \lambda_H} \ , \quad (17)$$

and

$$\sigma^2_{t_{s_H}} = 2\frac{c_1}{\mu^2} + 2\frac{c_2}{(\mu - \lambda_H)^2} - \left[\frac{c_1}{\mu} + \frac{c_2}{(\mu - \lambda_H)}\right]^2 , \quad (18)$$

respectively. Therefore, the mean value $E[\tau_H]$ can be readily determined,

$$E[\tau_H] = E[t_{s_H}] + (S-1) \cdot \left(\frac{1}{\mu p_H} + \frac{p_0}{\lambda p_H}\right) \quad (19)$$

and so is its variance $\sigma^2_{\tau_H}$

$$\sigma^2_{\tau_H} = \sigma^2_{t_{s_H}} + (S-1) \cdot \left[\left(\frac{2p_0(\mu(1-p_0) - \lambda)}{\mu p_H \lambda^2}\right) + \left(\frac{\lambda + p_0\mu}{\mu p_H \lambda}\right)^2\right]. \quad (20)$$

To compare the statistics of the time it takes to accumulate the first S video chunks in the buffer of high and low interest peers, we will approximate the cumulative distribution function of the random variables τ_H and τ_L as Gaussian, with mean and variance determined as above, confined to $\mathcal{R}^+$ and properly normalized. The Gaussian assumption will be justified by the sufficiently large value of S examined in next Section.

IV. Numerical Results

To put at work the idea of improving the experience of users that more frequently watch a specified channel, we need some description of the users' interests: to obtain it, we refer to the database of a small Italian platform, BeConn [9], that offers a conventional, client-server web-tv service to its customers. The available log records the behavior of BeConn users for nine months. We concentrate on the most popular channel of the platform, exhibiting the longest overall viewing time and the highest number of views: $M = 113$ users are identified as its spectators. Next, users are grouped in clusters, employing Fuzzy C-Means (FCM) [10]-[11], a well-known data mining technique that we apply considering the following metrics: the average duration of the viewing session of each user and its number of views. We work with $K = 2$ clusters, that for the sake of clearness are denoted as C_H and C_L: as FCM is a soft clustering algorithm, each user i, $i = 1, 2, \ldots, M$, belongs to both clusters C_j, $j = H, L$, with a different degree, identified by the vector of its membership values $\mathbf{m_i} = [m_{iL}, m_{iH}]$, $0 \leq m_{ij} \leq 1$. The rule we adopt when implementing the algorithm is to assign the i-th element to cluster C_H if $m_{iH} \geq m_{iL}$, to cluster C_L otherwise. The fuzziness factor f, $f > 1$, is the algorithm parameter that rules how fuzzy the clustering output will be: for our data set clusters do not significantly change for different f values, and in what follows we refer to the output obtained for $f = 2$, the most commonly used value. Once the two clusters are

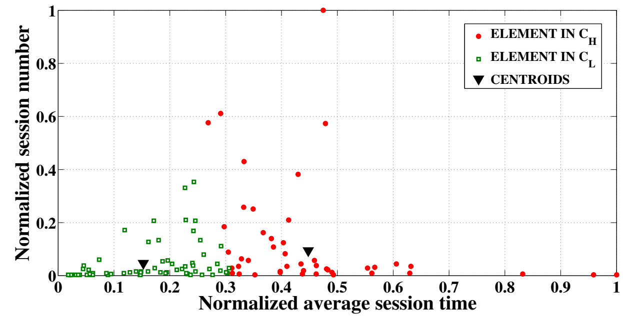

Fig. 2. Data referring to the most popular television channel and its clustering

detected, high interest users of the popular channel are identified as those belonging to the cluster C_H whose centroid, (i.e., whose representative element), exhibits a longer average session time and a higher number of views than the centroid of C_L. Fig.2 graphically illustrates the outcome of this choice: the figure reports the elements we classified, displaying for each of them the average session time and the number of viewing sessions, both normalized with respect to their maximum observed value. In order to evaluate the impact of our proposal, we first resort to a numerical approach, relying upon an event-driven p2p simulator based on the work by Zhang [12]. We modify it introducing the new video chunk scheduling algorithm, building a single channel, mesh overlay that exhibits the features described in Section III. The simulated system has an average population of $N = 500$ active peers, with heterogeneous upload and download capacities. The values and percentages we employ, drawn from the current Italian ADSL offerings, are reported in Table I. Further, the streaming rate is set to 350 kbit/s. Under these assumptions σ, the system resource index, defined as the ratio between the total upload overlay capacity and the streaming rate times the number of peers viewing the channel, is equal to 1.33, indicating that there is enough bandwidth to distribute the video content to all overlay members at the desired rate. Each node has 15 neighbors, the chunk receiving window is 20 s, the request interval for video chunks is set to 500 ms, buffer maps are sent out every 1 s and the size of each video chunk is 20 kbytes. Video chunks are fragmented and carried inside packets whose payload is 1250 bytes, transmitted over the underlying wide area network, where they suffer latencies that are captured via a suitable delay matrix. Every peer is associated to one viewer of the popular televi-

	class 1	class 2	class3
upload bandwidth (kbit/s)	384	704	1024
download bandwidth (kbit/s)	7000	7000	10000
% of peers	80	15	5

TABLE I
Users Bandwidth Profiles

sion channel of BeConn in a pseudorandom manner, therefore belonging to either cluster C_H or C_L. As M, the number of users within the available data set is lower than N, we repeat the binding several times, until all peers are classified. Both their interarrival time and their viewing session time are exponentially distributed, with average value $\frac{1}{\lambda} = 0.1$ s and $\frac{1}{\mu} = 1500$ s, respectively: all nodes exhibit the same behavior, in order to exclusively highlight the effects of the new mechanism introduced in the p2p overlay. As Section III described, video chunks for high interest users are allowed to queue up at the parent peer's site: in what follows the queue length is equal to $Q = 2$ chunks, if not otherwise specified. Figs.3 and 4 report the comparison between the performance experienced by low and high interest peers, also reporting the behavior of the nodes in the reference system with no differentiation, where video chunks do not queue up. In detail, Fig. 3 shows the delivery ratio as a function of the simulation time: we observe that the delivery ratio is very close to 1 for both categories of users, with a slight better behavior guaranteed to high interest nodes. Fig. 4 illustrates the average playback delay as a function of the simulation time: the delays in Fig. 4(a) refer to the case where the neighborhood construction algorithm is random; those in Fig. 4(b) to the case where the algorithm employs the $f(B, I)$ function defined in (1). Regardless of the adopted algorithm, a clear service differentiation is attained for the two classes: furthermore, the introduction of the $f(B, I)$ based algorithm slightly reduces the playback delay of high interest users with respect to the original system and it is also beneficial for low interest nodes. Fig. 5 reports the startup delay

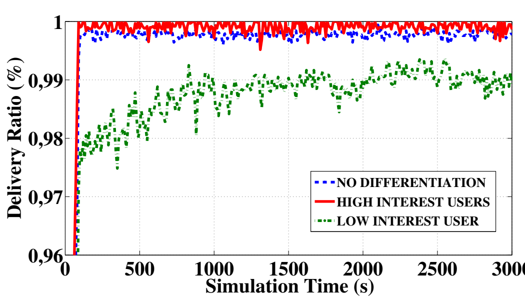

Fig. 3. Comparison between the delivery ratio of high and low interest users.

CDF: in our simulations this delay coincides with the buffering delay and it is determined computing the time that elapses from the instant when the peer requests its first video chunk up to the point in time when 10 s of video have accumulated in its buffer, an amount deemed reasonable to let the playout begin. Within the examined setting, this corresponds to the reception of $S = 21$ consecutive video chunks (to slightly relax this condition, we accept that two consecutive chunks can be missed, but no more than once). This figure reports the reference case (no differentiation between classes), as well as the curves referring to high and low interest peers, for the previously recalled algorithms of neighborhood creation. We observe that such algorithm does

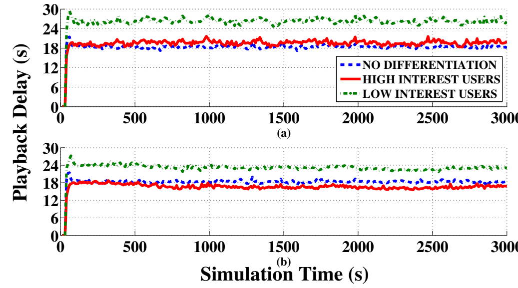

Fig. 4. Comparison between the playback delay of high and low interest users. (a): random neighborhood construction. (b): neighborhood construction ruled by the $f(B, I)$ function.

not play a significant role when determining the startup delay CDF of high interest peers: for this class of users, the most relevant improvement is introduced by the scheduling. As regards low interest peers, when the $f(B, I)$ function comes into play, a better service – a reduced startup delay – is observed. Indeed, $f(B, I)$ forces nodes to preferentially select as neighbors high interest users, i.e., peers that are more likely to own the desired content: this is beneficial for everyone in the system. Last, Fig. 6

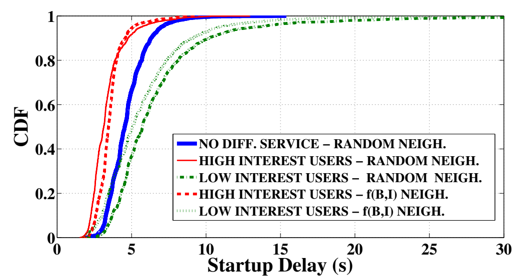

Fig. 5. Startup delay CDF.

reports the distribution function of the buffering delay of high and low interest peers, τ_H and τ_L, as analytically computed resorting to the model proposed in Subsection IV.B. With no aim for a direct comparison against the CDF values obtained by simulation, which is impracticable, given the simplicity of the model against the sophistication of the system simulative replica, it is nevertheless interesting to observe how the analysis outlines a clear service differentiation.

V. Conclusions

This paper has quantitatively investigated how a p2p television streaming platform can guarantee a more satisfying service to its frequent channel viewers than to less regular spectators. Learning about the users' interests from their behavior in the recent past, it is possible to cluster them in groups and spot habitual viewers; then, the video chunk scheduling algorithm within the overlay is altered, with the aim of granting them a higher delivery ratio, a lower playback delay and a faster access to the selected channel.

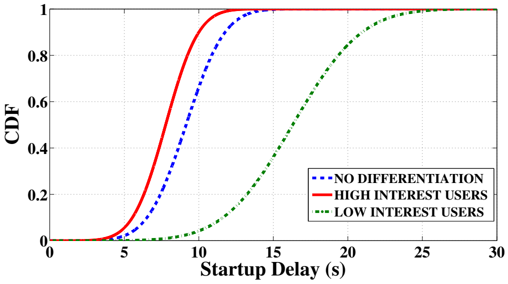

Fig. 6. Startup delay CDF: theoretical curves.

An analytical model is developed, to capture the difference in startup delay that the proposed modifications introduce; several additional metrics are numerically determined, in order to size up the performance of high and low interest viewers. The results indicate that a clear differentiation is attained and also quantify the effects that different neighborhood creation algorithms have on the amount of such gap.

APPENDIX

A. Interdeparture Time

Let us assume that at time t_0 the system of Fig.1 is non empty: conditioning upon such event, with the help of Figs. 7(a) and 7(b) we conclude that the only interdeparture time we can observe coincides with the service time of a customer. Hence we have

$$\Phi^*(s)|\text{non empty system} = \left(\frac{\mu}{\mu+s}\right). \quad (21)$$

If at time t_0 the system is empty, from Figs. 7 (c) and (d) we conclude that the interdeparture time is given by the sum of two independent random variables: regardless of the type of customer who will get into service, the first is a residual interarrival time, exponentially distributed with parameter $\lambda = \lambda_H + \lambda_L$, the second is a service time, so that

$$\Phi^*(s)|\text{empty system} = \left(\frac{\lambda}{\lambda+s}\right)\left(\frac{\mu}{\mu+s}\right). \quad (22)$$

Unconditioning, we obtain $\Phi^*(s)$ expression appearing in (9).

B. Time in System

For the system whose state diagram appears in Fig.1, consider the time spent in system by a low interest customer, t_{s_L}. It is easy to conclude that such customer, corresponding to the chunk request of a low interest user, is either immediately served or blocked, but it never incurs into queueing, so that its time in system t_{s_L} coincides with its service time.
On the contrary, high interest chunk requests can queue up: if we apply a reasoning analogous to the one proposed in [8], Chapter 5, for the M/M/1 system, we conclude that the time spent in system by a high interest customer, conditioned on the number k found in system upon its arrival, is the sum of $k+1$ exponentially distributed random variables. The Laplace transform of its p.d.f. is therefore given by:

$$S_H^*(s|k) = \left(\frac{\mu}{\mu+s}\right)^{k+1}, \quad (23)$$

so that unconditioning we have

$$S_H^*(s) = \sum_{k=0}^{\infty} S_H^*(s|k)p_k\,; \quad (24)$$

replacing p_0 and p_k expressions as provided by (8) and (9) in (24), after a few algebraic passages we get

$$S_H^*(s) = \left(\frac{\mu}{\mu+s}\right)c_1 + \left(\frac{\mu-\lambda_H}{\mu-\lambda_H+s}\right)c_2\,, \quad (25)$$

where c_1 and c_2 appear in eq(17) of Section III.

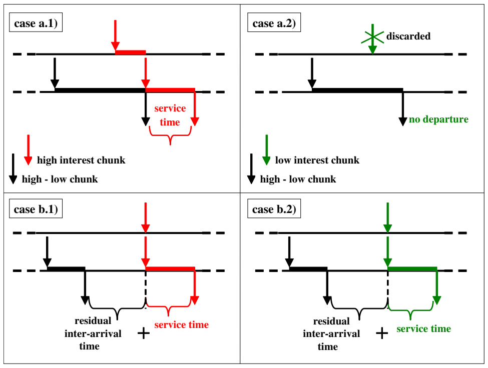

Fig. 7. Cases to consider for the interdeparture time in the system that differentiate between the two classes of users.

REFERENCES

[1] M. Merani and D. Saladino, “Live video and ip-tv,” in *Handbook of Peer-to-Peer Networking, X. Shen et al.*, Springer(US), Ed., 2010.
[2] B. Li, S. Xie, Y. Qu, G. Keung, C. Lin, J. Liu, and X. Zhang, “Inside the new coolstreaming: Principles, measurements and performance implications,” in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.
[3] PPTV, “http://www.pptv.com.”
[4] D. Wu, Y. Liu, and K. Ross, “Modeling and analysis of multichannel p2p live video systems,” *Networking, IEEE/ACM Transactions on*, vol. 18, no. 4, pp. 1248–1260, Aug. 2010.
[5] K. Ramachandran and B. Sikdar, “A queuing model for evaluating the transfer latency of peer-to-peer systems,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 3, pp. 367–378, March 2010.
[6] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello, “Push-to-peer video-on-demand system: Design and evaluation,” *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 9, pp. 1706–1716, Dec. 2007.
[7] F. Liu, B. Li, B. Li, and H. Jin, “Peer-assisted on-demand streaming: Characterizing demands and optimizing supplies,” *Computers, IEEE Transactions on*, vol. 62, no. 2, pp. 351–361, Feb 2013.
[8] L. Kleinrock, *Queueing Systems*. Wiley Interscience, 1975, vol. I: Theory.
[9] BeConn, “http://www.beconn.com.”
[10] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” *Journal of Cybernetics*, pp. 32–57, 1973.
[11] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
[12] M. Zhang, Q. Zhang, L. Sun, and S. Yang, “Understanding the power of pull-based streaming protocol,” *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 9, pp. 1678–1694, Dec. 2007.