

Adaptive Streaming in P2P Live Video Systems: A Distributed Rate Control Approach

MARIA LUISA MERANI and LAURA NATALI, University of Modena and Reggio Emilia, Italy

Dynamic Adaptive Streaming over HTTP (DASH) is a recently proposed standard that offers different versions of the same media content to adapt the delivery process over the Internet to dynamic bandwidth fluctuations and different user device capabilities. The peer-to-peer (P2P) paradigm for video streaming allows us to leverage the cooperation among peers, guaranteeing the service of video requests with increased scalability and reduced cost. We propose to combine these two approaches in a P2P-DASH architecture, exploiting the potentiality of both. The new platform is made of several swarms and a different DASH representation is streamed within each of them; unlike client-server DASH architectures, where each client autonomously selects which version to download according to current network conditions and to its device resources, we put forth a new rate control strategy implemented at peer site to maintain a good viewing quality to the local user and to simultaneously guarantee the successful operation of the P2P swarms. The effectiveness of the solution is demonstrated through simulation and it indicates that the P2P-DASH platform is able to provide its users with very good performance, much more satisfying than in a conventional P2P environment where DASH is not employed. Through a comparison with a reference DASH system modeled via the Integer Linear Programming (ILP) approach, the new system is shown to outperform such reference architecture. To further validate the proposal, in terms of both robustness and scalability, system behavior is investigated in the critical condition of a flash crowd, showing that the strong upsurge of new users can be successfully revealed and gradually accommodated.

CCS Concepts: • **Information systems** → **Multimedia streaming**; • **Computer systems organization** → **Peer-to-peer architectures**; • **Networks** → **Network simulations**

Additional Key Words and Phrases: DASH, peer-to-peer video streaming, flash-crowd, integer linear programming

ACM Reference Format:

Maria Luisa Merani and Laura Natali. 2016. Adaptive streaming in P2P live video systems: A distributed rate control approach. ACM Trans. Multimedia Comput. Commun. Appl. 12, 3, Article 46 (May 2016), 23 pages.

DOI: <http://dx.doi.org/10.1145/2912123>

1. INTRODUCTION

As several studies related to the use of the Internet indicate [Cisco WP 2015; European Commission 2012], we are witnessing an explosive increase of video content; IP video flows currently represent more than 50% of all Internet traffic and are expected to grow further at a very swift pace. End-users consuming video typically rely upon different devices, ranging from smartphones to tablets and PCs, and also undergo different connectivity conditions on the basis of their location: at home, in office, on the road. To counteract fluctuating network conditions and also to cope with heterogeneous user requests, a new delivery streaming framework termed dynamic adaptive streaming has

Authors' addresses: M. L. Merani and L. Natali, Dipartimento di Ingegneria "Enzo Ferrari," University of Modena and Reggio Emilia, Modena, Italy; emails: {marialuisa.merani, laura.natali}@unimore.it.

been recently introduced. Several proprietary solutions have first flourished, ranging from Smooth Streaming by Microsoft [2015] and Zambelli [2009] to HTTP Live Streaming by Apple [2015] and Dynamic HTTP Streaming by Adobe [2015]. They have in turn triggered the release of the Dynamic Adaptive Streaming over HTTP (DASH) standard [DASH-IF 2014; Sodagar 2011], which is currently the most widespread, internationally recognized approach: in DASH, the “adaptive” term refers to the capability of the technique to modify the features of the transmitted video in order to adapt it to varying network conditions and different user requests.

This work puts forth a peer-to-peer (P2P) system that exploits DASH technology for video streaming: in doing so, it embraces the future vision of the Internet provided in Babaoglu and Marzolla [2014], where network users are seen as collectively forming the “human grid,” whose distributed resources are at the basis of any service provisioning. A multi-overlay architecture is therefore proposed where the role of the servers in the distribution process is minimal; moreover, there are as many overlays as the number of available DASH representations, and an entirely distributed control strategy rules the transition of the peers from one overlay to another. Such control logic is locally implemented at the peer’s site and it strives to preserve both the quality that the single peer experiences and the good functioning of all overlays. In order to jointly achieve these goals, it relies on local and global parameters: The former reflect the peer’s status and the latter indicate the status of the different overlays. System behavior is explored in some relevant user scenarios and the findings are summarized as follows:

- the new distributed control allows the pure P2P platform to successfully reach steady state, providing the majority of its users with the desired video representation;
- in the most unfriendly scenario that has been investigated, the performance that the peers experience is significantly better than the one they would undergo in a DASH-unaware P2P system;
- when compared to the performance that a reference, idealized system would achieve, in the most favorable setting the current proposal attains a very close number of satisfied users, that is, users that stream the video at the desired bit rate, whereas, in the worst case, the number of satisfied users is slightly suboptimal but has much better streaming quality;
- the critical upsurge of a flash crowd is promptly revealed by the control algorithm and is satisfactorily handled;
- combining the DASH feature with P2P offers a significant advantage in terms of switching delay that the peers experience when requesting different video alternatives.

The rest of the article is organized as detailed below: Section 2 critically provides the state of the art and covers the related work; Section 3 illustrates the proposed architecture and the distributed rate control algorithm, while Section 4 describes the integer linear programming approach that contributes to validate the proposal. In Section 5, a numerical picture of the performance attained by the new platform is offered and Section 6 draws the conclusions.

2. STATE OF THE ART

The majority of the works on DASH, the new adaptive streaming standard, focuses on client-server architectures, and, in this setting, different algorithms to dynamically choose the most appropriate DASH representation have been proposed. Among some recently published contributions, the study in Zhang et al. [2014] investigates the adoption of DASH over multiple content distribution servers. In the examined architecture, the rate adaptation logic is applied to blocks of video fragments, downloaded in parallel from different servers; a novel proportional-derivative controller is further introduced to adapt the video bit rate, and system performance is studied through a theoretical

approach and some laboratory experiments. In a similar manner, in a P2P architecture, each user in parallel requests missing video content to several neighbors, that is, each of them acts as a concurrent server; in this context, however, the pool of peers that provide useful content varies over time, often unpredictably.

Various bandwidth-based solutions have also been introduced in order to perform rate control in DASH. In Liu et al. [2011], the authors suggest an adaptation algorithm that relies on the smoothed HTTP throughput to determine if the current media bit rate matches the end-to-end network bandwidth. In Li et al. [2014], the proposed strategy relies on the TCP client throughput to determine the available bandwidth in the presence of congestion, whereas it constantly probes the network and adapts to its new conditions otherwise. The status of the playback buffer is considered in Mok et al. [2012], which strives to preserve the minimum buffer length that avoids interruptions and minimizes video quality variations during the playback; so does Suh et al. [2014], whose authors investigate a more conservative, QoE-driven scheme. More generally, current network conditions are monitored to efficiently—and hopefully fairly—leverage network resources, whereas the status of the client's buffer is examined to guarantee a smooth *playout* and avoid video stalls, the focal point being the client's streaming quality. Yet, the algorithms that govern the video streaming requests are steered only by the user's perspective: That is, the client's objective is to optimize its viewing experience, regardless of the impact that its decisions have on the video quality perceived by other users. To the authors' knowledge, in the client-server strategies proposed in the literature the user is not directly interested in the status of the whole system, nor does it possess the concept of cooperation or partnership with other users.

The present proposal adopts an opposite approach: In its decision-making process, the peer takes into account both its local point of view and an overall perspective; as a matter of fact, in a P2P architecture, selfish behaviors do not produce advantages either to the single node or to the whole system. Hence, a generic peer cannot neglect the status of the nodes within its swarm to attain a satisfying video quality. Differing from previous works, we therefore propose a new rate control algorithm that strives to guarantee a good performance to the single user without losing the overall system sight: Accordingly, a node selects the most proper DASH representation, taking into account some local, indirect video quality measures and also some parameters that reflect the health status of the P2P system.

Some valid contributions in the literature have put forth the joint adoption of P2P and DASH. Lederer et al. [2012] proposes a standard compliant solution (pDASH) that integrates peer-assisted streaming in conventional DASH client-server systems, and shows that the P2P component permits us to significantly reduce the server load; furthermore, the same solution is investigated in a mobile environment in Klusch et al. [2014]. Tian et al. [2013] applies DASH to a particular P2P architecture where the server capacity is infinite and exploits game theory to rule the switching process among different representations. Differing from Lederer et al. [2012] and Tian et al. [2013], we do not examine a Video on Demand (VoD) system: Both works correctly investigate platforms whose population size is very limited, as is the case for VoD, and interpret P2P as a secondary, although beneficial, feature of the video distribution architecture. On the contrary, our study delves the feasibility and the achievable performance of a system that delivers a *live streaming channel* to a *large floor of users*, with *very little contribution from the server*.

3. P2P-DASH

3.1. DASH Essentials

Dynamic Adaptive Streaming over HTTP is a standard that serves the purpose of delivering multimedia content through the network via conventional web servers. In

DASH, the different constituent components of the multimedia content, that is, video, audio, and text, are made available to the final consumer in different versions, displaying different characteristics: As an example, these might be alternative encoded versions at different bit rates of the audio and video components or different language versions of the audio component in the case of pre-recorded material. In DASH, the multimedia content is organized into a sequence of consecutive temporal periods and it is accessed as a collection of media segments: A manifest file, the Multimedia Presentation Description (MPD), details the alternatives—also termed *representations*—that are available for each component in every period and provides the URLs at which the corresponding segments can be accessed. The MPD is initially passed to the DASH client, which next proceeds to issue HTTP get requests to retrieve the segments of the desired representations; during the streaming session, the client can dynamically adjust the client-server communication, moving from one representation to another. The DASH standard does not indicate which adaptation algorithm to employ, nor does it detail the fetching mechanism that the client adopts; these are left entirely open.

3.2. P2P System Architecture

In the proposed architecture, P2P is the significant feature that rules the distribution of multimedia content: Users are therefore organized in a virtual network, termed overlay, which is built at the application layer, and are actively involved in the video diffusion process. They relay portions of the content they already own in their local buffer to other users—their peers—through the virtual links of the overlay. We consider the case where a single video channel has to be streamed to a population of users whose size is N and assume that K DASH representations of the video are instantaneously available. Unlike VoD client-server architectures, where K takes on relatively high values, in the examined scenario, K is intentionally confined to lower values, not to excessively complicate the design and implementation of a live system that has to respect tighter latency constraints. As a matter of fact, we envision a platform for delivering near-real time streaming events, such as sport competitions, where the server is required to instantaneously produce the different DASH video versions: Working with relatively small values of K relieves the burden placed on the server.

Different DASH representations of the same video content may differ for resolution, frame rate, color depth, and quality, and each of these parameters influences the final video bit rate on the basis of the chosen codec. DASH is codec agnostic, meaning that it can be equally employed with different video codecs, such as H.264/AVC, H.265/HEVC, and MPEG-2 Vide, and since it is out of our scope to evaluate codec performance, we simply assume that the bit rate is what distinguishes different DASH representations.

Each representation is distributed within a separate P2P overlay: The j th representation exhibits a streaming rate r_j and is delivered within the j th swarm, $1 \leq j \leq K$; without loss of generality, we set $r_j < r_{j+1}$, $\forall j$, $j = 1, 2, \dots, K - 1$. As Figure 1 indicates, the server is responsible for initially igniting the diffusion process: It delivers each representation to the corresponding overlay as a sequence of video chunks, relatively small fragments of the encoded video, identified by a unique sequence number. Every peer belongs to one overlay at a time and contributes to sustain the process with its own upload bandwidth, further spreading the video chunks that it receives to other peers belonging to the same overlay. Peers have heterogeneous upload and download capacities and therefore supply content to other mates in a diversified manner.

The examined P2P overlays are mesh based, and within each of them every peer exchanges video chunks with M neighbors randomly chosen within a list of potential partners provided by the channel server at joining time. In order to maintain an adequate download rate, a peer immediately replaces a neighbor that left the system or a neighbor that does not supply a sufficient video content, choosing a new one within

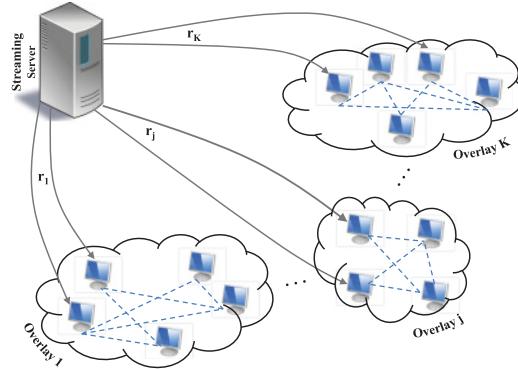
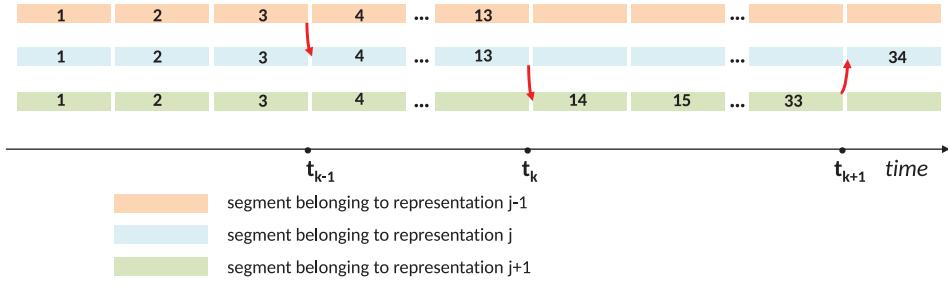


Fig. 1. System architecture.

an updated list of potential partners. Every overlay implements a pull-based protocol: The peer periodically informs its neighbors about the video chunks that it holds in its streaming buffer by forwarding its buffer maps. Moreover, the peer regularly asks its neighbors for the chunks that are missing within its current request window W . Such a window identifies the range where chunks that can be requested by the node fall: Its right edge coincides with the highest chunk sequence number that the node is aware of from the inspection of the buffer maps it received, its left edge falls X seconds behind, where X indicates the window duration. In a live streaming system, a newly joining peer has to hook the ongoing video stream as soon as possible. Therefore, the right edge of its request window has to chase the newest chunk sequence number in the swarm, and then the window has to progressively slide forward whenever the peer learns through new buffer maps that a chunk with a higher sequence number is available. When a chunk that is yet to be received falls outside the current request window, it will not be claimed any longer as outdated.

The DASH representation that a peer desires is the representation featuring the streaming rate the peer aims at viewing the video content at: We observe that it depends on the combined limits set by the user's access network and by the display characteristics of its terminal. The user's preferences might also contribute to determine it, as is the case when different charging rates are applied for distinct video qualities.

A peer requesting the video channel for the first time begins watching representation 1, which features the lowest bit rate, r_1 , and, hence, the lowest quality: This choice allows us to start the playout in a reasonably short time, and positively influences the overall quality that the user experiences [Zambelli 2009]. After joining overlay 1, if the peer's desired representation is not the video alternative at the lowest rate, the peer attempts to move upward and possibly succeeds; however, during the flow of the streaming, the peer might also have to move downward to a lower-quality representation, depending on current system conditions, and then it will dynamically attempt to advance again. These actions translate into migrations from one overlay to another. We require that a peer exclusively moves from its current overlay to an adjacent one, that is, from overlay j to $j + 1$ or $j - 1$ if $2 \leq j \leq K - 1$, from overlay j to overlay $j + 1$ if $K = 1$, and from K to $K - 1$ if $j = K$; this guarantees the minimal gap between two consecutive representation playouts, as recommended in Yitong et al. [2013] and Zink et al. [2003] to confine the amplitude of bit-rate variations and its negative effects on the perceived video quality. When a node migrates from one overlay to another, it takes some time before the peer becomes operative in the new swarm; first it must



t_{k-1}, t_k, t_{k+1} time instants when transitions occur

Fig. 2. Migrating to different media representations in ideal conditions.

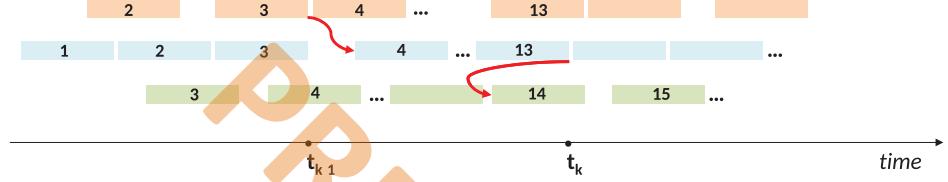


Fig. 3. Migrating in real conditions.

disconnect its previous neighbors, join the new overlay, and ask the tracker for a new list of potential partners in order to construct the new neighborhood. At this point, it waits to receive some buffer maps, needing to know which video chunks are available from its partners, and then it issues the first chunk request for the new representation.

Furthermore, we assume that every DASH segment is made of the same number n of video chunks and that the chunk duration t_{chunk} is the same in every overlay, which translates into a different amount of media bits being placed in every chunk. Indicating by L_j such size for the chunks that are distributed within overlay j , we have $L_j = t_{chunk} \cdot r_j$; as Figure 2 shows for the ideal case of perfectly synchronized overlays, this choice allows a smooth transition between different representations. In the realistic circumstance of a time lag between different overlays as portrayed in Figure 3, the constant duration of the chunks still preserves the smoothness of the transitions, as long as the user's buffer can absorb time shifts among different representations.

3.3. The Proposed Algorithm

In a conventional client-server scenario, DASH provides the user with the functionalities needed to perform adaptive streaming and leaves open the selection of the algorithm that rules the switching among the different available representations. The client is assigned the task to check its current conditions via the monitoring of various parameters and to ask the server for a different representation if needed. As previously observed, proposals in the literature rely on the observation of several indicators and put forth different criteria to govern the switching. However, it is exclusively the local perspective that plays a role in the design of the control algorithm: Within the current proposal, although the primary aim of the peer remains to experience a satisfying streaming quality and to efficiently deploy its network resources, the user's decisions can no longer be taken in isolation, as the peer has to be fair to all other peers within the system. We therefore require that every peer employs two distinct types of status indicators, *local* and *global*, to decide if, when, and where to migrate; whereas

the former indicators indirectly supply information about the peer's video quality and are locally measured by the peer itself, the latter indicators provide a clue about the current health of the overlays and are periodically handed out by the server to all peers that rely on both to enforce the adaptive rate selection algorithm.

Among local indicators, we utilize:

- the Delivery Ratio (*DR*), defined as the ratio between the number of video chunks that meet the playback deadline over the total number of chunks that a peer should receive, measured with a periodicity of \tilde{t} s. Such a ratio indicates the throughput that the peer experiences and indirectly signals the quality of the received video in the very recent past;
- the Request Window State (*RWS*), defined as the ratio between the number of downloaded video chunks within the current request window W and the size of such window measured in number of video chunks, $0 \leq RWS \leq 1$. This indicator provides an indirect forecast of video quality in the near future, as its value reflects the imminent status of the playout buffer at the peer's site.

Among global indicators, we select those among the status pointers determined by the server that can be distributed to peers with very little effort, namely:

- the instantaneous resource index [Wu et al. 2009] of the j th overlay, $\sigma_j(t)$, $j = 1, 2, \dots, K$, defined as

$$\sigma_j(t) = \frac{C_{Sj} + \sum_{i \in N_j(t)} c_i}{|N_j(t)| \cdot r_j}, \quad (1)$$

where C_{Sj} is the capacity that the server commits to the j th overlay to distribute the representation with rate r_j , c_i is the upload capacity of the i th peer belonging to overlay j , $N_j(t)$ is the set of active nodes within such overlay at time t , and $|N_j(t)|$ is the set cardinality;

- the instantaneous efficiency of the j th overlay, defined as

$$E_j(t) = \frac{U_{Sj}(t) + \sum_{i \in N_j(t)} u_i(t)}{|N_j(t)| \cdot r_j}, \quad (2)$$

where $U_{Sj}(t)$ is the actual upload rate that the server provides to overlay j at time t and $u_i(t)$ is the actual upload rate at time t of the i th user within the same overlay.

When the instantaneous resource index $\sigma_j(t)$ takes on a value greater than or equal to 1, in principle, the j th overlay can successfully guarantee the video delivery to all of its members, whereas when its value falls below 1, the overlay operates in a critical regime; hence, σ_j is a high-level indicator of the overlay health. However, the instantaneous efficiency $E_j(t)$ provides a more accurate picture, as it captures some system behavior that would otherwise go unseen if σ_j only were examined. To depict a scenario where this happens, it suffices to examine a flash crowd of viewers that abruptly enters the system, wishing to stream the video: Newly incoming peers initially act as free riders, as they momentarily have no video chunks to share. We must conclude that the value of the instantaneous efficiency drops, revealing a potentially critical operating condition that the resource index cannot seize.

Let us next consider the generic peer i within overlay j and indicate by $r_d(i)$ the streaming rate of its desired representation; the steps that the algorithm enforced by the peer goes through every Δt seconds are listed below, while Table I collects all the employed variables:

- (i) peer i checks its current streaming rate r_j against $r_d(i)$ and if $r_j < r_d(i)$, that is, if the peer is not satisfied, it first verifies whether it can leave its current overlay or it

Table I. Variables Employed by the Algorithm

r_j := streaming rate within the j -th overlay
$r_d(i)$:= i -th peer desired streaming rate
σ_j := j -th overlay resource index
c_i := i -th peer upload capacity
$E_j(t)$:= j -th overlay efficiency at time t
$DR_i^{(t)}$:= i -th peer Delivery Ratio at time t
$RWS_i^{(t)}$:= i -th peer Request Window Status at time t

has to defer its departure. This last circumstance occurs if overlay j is not in good health and node i upload capacity is beneficial to it, that is, if σ_j is lower than 1 and if the peer upload capacity c_i is greater than the streaming rate r_j ; in this case, it is convenient that the peer does not move upward to overlay $j + 1$. If on the contrary nothing prevents the departure, peer i further verifies if its future contribution to the target overlay $j + 1$ will be positive, that is, if its upload capacity c_i is greater than the streaming rate r_{j+1} . If so, then the peer migrates, as it will be beneficial to overlay $j + 1$. If not, then the node further verifies that overlay $j + 1$ has abundant overall upload capacity and that the video diffusion process within the overlay is taking place in an efficient and noncritical manner, hence the target overlay is able to accommodate a new peer, regardless of it being a relatively “poor” contributor. So, the peer moves to overlay $j + 1$ if $\sigma_{j+1} > 1$ and $E_{j+1}(t) > E_{thres}$, where E_{thres} is a properly set threshold.

- (ii) peer i also verifies its local status quality indicators, updating the weighted moving average of its delivery ratio and request window state in the following manner:

$$DR_i^{(t)} = w_D \cdot DR_i^{(t)} + (1 - w_D) \cdot DR_i^{(t-\Delta t)} \quad (3)$$

and

$$RWS_i^{(t)} = w_W \cdot RWS_i^{(t)} + (1 - w_W) \cdot RWS_i^{(t-\Delta t)}, \quad (4)$$

respectively, where w_D and w_W are the weight coefficients. If $DR_i^{(t)}$ and also $RWS_i^{(t)}$ are below their predefined thresholds, RWS_{thres} and DR_{thres} respectively, then the viewing quality is not deemed satisfying and the peer scales down to a lower rate representation, hence it moves to overlay $j - 1$.

The pseudo code describing the algorithm locally implemented by the peer is reported in Algorithm 1.

Note that in the rate selection algorithm that is proposed to steer the peer’s movements among overlays, the upgrade decision is conservatively influenced by the peer’s local metrics—its current rate r_j and its desired streaming rate $r_d(i)$ —and also by the global indicators of the current and destination overlays, whereas the downgrade process is exclusively governed by local indicators. The rationale behind this choice is that global indicators point to the overall status of the system and therefore let the peer learn if the migration from its current swarm to the overlay distributing a higher bit rate representation can be safely performed. On the other hand, through local indicators the peer indirectly measures its streaming quality and decides if it is time to scale down to a lower DASH representation, in order to preserve a good viewing experience.

4. INTEGER LINEAR PROGRAMMING MODEL

To demonstrate that the proposed architecture and the rate control rule that it adopts are able to cope with the expectations of the users in terms of desired streaming rate, we formulate an Integer Linear Programming (ILP) problem modeling a simplified, static system that will represent an initial term of comparison. More accurately, we

ALGORITHM 1: Rate Switching Control Algorithm

```

Node  $i$  in overlay  $j$  every  $\Delta t$  seconds
;verifies its satisfaction
if ( $r_j < r_d(i)$ ) then
    ;verifies the current overlay status
    if (( $\sigma_j < 1$ ) and ( $c_i >= r_j$ )) then
        do not migrate to overlay  $j + 1$ ;
    else
        ;verifies the destination overlay status
        if (( $c_i > r_{j+1}$ ) or ( $\sigma_{j+1} > 1$  and  $E_{j+1} > E_{thres}$ )) then
            migrate to overlay  $j + 1$ ; exit;
        end
    end
end
;verifies its viewing quality
if (( $DR_i^{(t)} < DR_{thres}$ ) and ( $RWS_i^{(t)} < RWS_{thres}$ )) then
    migrate to overlay  $j - 1$ ; exit;
else
    stay in overlay  $j$ ; exit;
end

```

examine K overlays/representations and a population of N users with heterogeneous streaming rate requirements, and state that the system goal is to satisfy as many users as possible providing them with the streaming rate they desire, subject to the only constraint that in every overlay the resource index takes on a value greater than one. To mathematically state such problem, we define user i satisfaction within overlay j as:

$$s(i, j) = \begin{cases} 1, & \text{if } r_d(i) = r_j \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

and state that the goal is to maximize the satisfaction function \bar{S} , defined as

$$\bar{S} = \sum_{i=1}^N \sum_{j=1}^K s(i, j) \cdot x_{ij}, \quad (6)$$

subject to the following constraints:

$$\sum_{j=1}^K x_{ij} = 1, \quad \forall i, i = 1, 2, \dots, N, \quad (7)$$

$$r_d(i) - r_j \geq 0, \quad \text{if } x_{ij} = 1, \quad \forall i, i = 1, 2, \dots, N, \quad (8)$$

and

$$C_{Sj} + \sum_{i=1}^N c_i \cdot x_{ij} - r_j \cdot \sum_{i=1}^N x_{ij} \geq 0, \quad \forall j, j = 1, 2, \dots, K \quad (9)$$

where x_{ij} is a binary variable that indicates if peer i is assigned to overlay j , that is,

$$x_{ij} = \begin{cases} 1, & \text{if peer } i \text{ belongs to overlay } j \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

The set of constraints in (7) guarantees that every peer belongs to one overlay only. The set of constraints in (8) indicates that the i -th peer can only belong to an overlay

that distributes the video at a rate r_j lower than or equal to its desired rate $r_d(i)$. The set of constraints in (9) is equivalent to the requirement that in every overlay the resource index σ_j is greater than 1. To verify last statement, it suffices to observe that $r_j \cdot \sum_{i=1}^N x_{ij} = r_j \cdot |N_j|$: dividing both members of (9) by it easily demonstrates the equivalence.

The solution to this problem answers the following question: If a genie were to allocate as many users as possible in the way they desire, exclusively respecting the resource index constraint in every overlay, then what is the maximum value of satisfaction that could be attained? Note, however, that two important remarks apply:

- the ILP model represents a simplified P2P-DASH system operating in a static setting, where no notion of quality is present and where peers do not dynamically enter and leave the system; as such, it provides an upper, optimistic bound to the actual global satisfaction that any newly proposed system exhibits in a real scenario;
- There might be more distributions of the peers within the overlays that all achieve the same, maximum value of satisfaction. It is therefore useful to verify whether we have a unique distribution or not: If the former circumstance occurs, then we can consider the actual values it provides and compare against them the distribution determined by a newly proposed control.

In order to check whether we have a unique problem solution and consequently one single distribution for the online nodes in each overlay, we proceed in the following manner: we numerically solve the problem and determine the achieved value of satisfaction S , then proceed to a new formulation, where we add to the previous constraints in (7) and (9) the following:

$$\sum_{j=1}^K \sum_{i=1}^N s_{ij} \cdot x_{ij} \geq S, \quad (11)$$

and replace the objective function in (6) with a new one, namely

$$S' = \sum_{i=1}^N \sum_{j=1}^K s(i, j) \cdot w_{ij}, \quad (12)$$

the w_{ij} 's being uniformly randomly picked values in $[0, 1]$. If the distribution of peers within the different overlays that fulfills the new maximization problem does not vary with respect to the original solution, then the distribution is unique and can be profitably employed for comparison purposes. We anticipate here that applying this procedure to the examined setting revealed that the solution we determined is indeed unique.

5. NUMERICAL RESULTS

5.1. Simulation Setup

To evaluate the performance of the system under investigation and of the algorithm that it employs, we implemented an event-driven simulator based on the source code available in Zhang et al. [2007]. We built a replica of a multi-overlay system, where every swarm streams the same video at a different bit rate. The system has an average population of $N = 2000$ active peers, that dynamically enter and leave; nodes populate the system within the first 20 s and in this interval their interarrival times are exponentially distributed with an average of 0.1 s; after the first 20 s, the interarrival times are modulated so as to keep nearly constant the number of peers in the system. Session times are exponentially distributed, with an average of 1500 s. Nodes

Table II. User Capacity Profiles and Percentages

	Class 1	Class 2	Class 3	Class 4
Upload capacity (kbit/s)	704	1024	1500	10000
Download capacity (kbit/s)	2048	8192	10000	50000
% of peers	20	21	42	17

belong to four different classes whose upload and download capacity values stem from the current European Internet connection offerings [Swisscom 2014; Orange 2014; T. Net 2014]; the percentages of users belonging to the different classes are drawn from the Akamai European average connection speed report [Akamai 2014]; the employed values are reported in Table II. $K = 4$ video representations are available at rates 700, 1500, 2500, and 3500 kbit/s; such values have been chosen having in mind the typical streaming rates of Internet Standard Definition video (SD) and High Definition video (HD). The streaming server allocates to each overlay only a small amount of its upload capacity, equal to four times the rate of the video, that is, $C_{S_j} = 4 \cdot r_j$, indicating that our focus is on a pure P2P system. Moreover, the size of the current request window that every peer works with is $X = 20$ s, the chunk duration is $t_{chunk} = 200$ ms, the delivery ratio is locally computed every $\tilde{t} = 5$ s and the threshold values we employed are the following: $DR_{thres} = 0.5$, $RW_{thres} = 0.3$ and $E_{thres} = 0.9$. In next subsections we will provide a thorough justification for these choices. The coefficients for the computation of the weighted moving average of DR_i and RWS_i are $w_D = \frac{1}{3}$ and $w_W = \frac{2}{3}$, respectively, revealing that, with regard to the local delivery ratio we tend to privilege stability in the estimate in (3), whereas for the request window state in (4) we give priority to what happens at current time. Last, the periodicity that the rate control algorithm employs is $\Delta t = 4$ s.

We begin investigating system behavior in three indicative scenarios, termed conservative, uniform and aggressive: the aim is to quantify system behavior in distinct, meaningful settings, each representing a reference case that might indeed occur. Within the conservative setting, every peer is allowed to prudently stream the highest representation whose bit rate is lower than or equal to its upload capacity: so, class 1 and class 2 users can only stream representation 1, class 3 users representation 2 and class 4 users representation 4. Within this scenario every overlay is exclusively composed by peers whose upload capacity is greater than or equal to the streaming rate of the video that the overlay distributes and bandwidth resources are therefore abundant. The second investigated scenario is termed uniform, as the generic peer uniformly and randomly selects the representation it desires, subject to the natural constraint that it is lower than the peer's download capacity. The underlying idea is to mimic the spreading of different representation requests due to the users' displays, with heterogeneous resolution capabilities. It follows that the representation range is limited for class 1 users only, that choose between representation 1 and 2 with equal probability, whereas users belonging to classes 2, 3, and 4 equally distribute their video requests among the four available representations. The third scenario is termed aggressive, as every peer aims at streaming the video at the highest representation whose bit rate is lower than its download capacity: in this case all peers aim at streaming representation 4, except for class 1 peers, that are confined to representation 2. In the aggressive scenario it is impossible to fulfill all users' requests: if peers were uncritically placed within the overlays distributing the desired representations (2 and 4), the aggregate upload capacity of each swarm would not be enough to satisfactorily deliver all peers the video, the resource index values being markedly lower than 1, namely, $\sigma_2 = 0.48$ and $\sigma_4 = 0.91$. So, in this setting the challenge is to place as many peers as possible within the desired overlay, in a manner that secures good values to the local quality indicators.

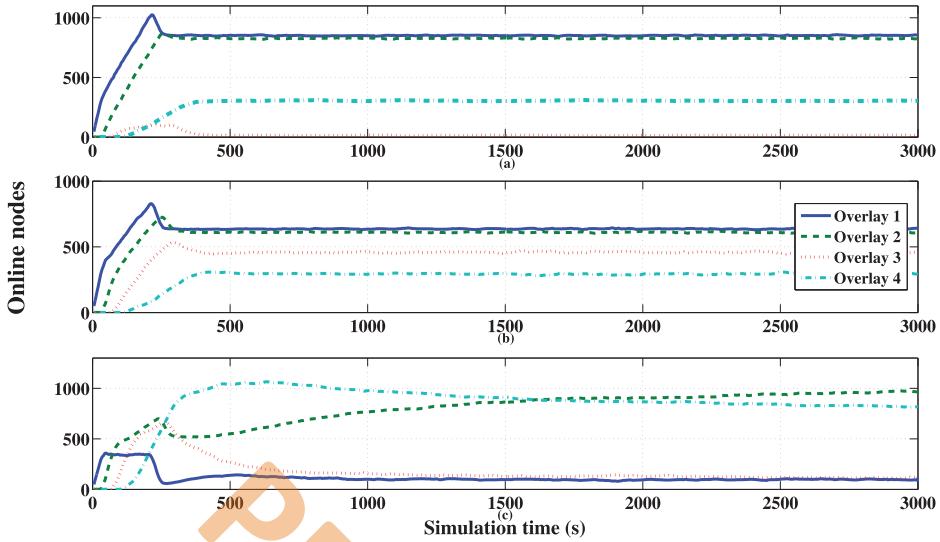


Fig. 4. Node distribution in the examined scenarios: (a) conservative, (b) uniform, (c) aggressive.

Table III. Number of Nodes Wishing to Stream Each Representation

Scenario	repr. 1	repr. 2	repr. 3	repr. 4
conservative	820	840	—	340
uniform	600	600	400	400
aggressive	—	400	—	1600

The results of next subsection numerically illustrate our findings.

5.2. Evaluating Performance in Different Scenarios

Figure 4 reports the distribution of online nodes among the four overlays, each corresponding to one video alternative, as a function of time. Figure 4(a) refers to the conservative setting: after the initial transient, when we compare the number of nodes within each overlay shown in this figure to the number of nodes that wish to stream every representation, reported in the first row of Table III, we conclude that they are very close. On the contrary, Figure 4(b) shows that in the uniform scenario, the nodes that stream representation 4 are fewer than the ones that would like to stream it, as indicated in the second row of Table III, revealing that it is not possible to serve all peers requesting the highest rate representation, due to the peers' scarce upload capacity. As a result, the rate control algorithm redirects some of these peers towards the overlays that distribute lower rate videos: this also explains why the number of online nodes watching representations 1 and 3 slightly deviates from the users requests indicated in the same table row. Figure 4(c) refers to the aggressive scenario: here, it is worth noting that the initial transient prolongs longer and that the number of nodes in each distinct overlay significantly differs from the number of peers wishing to stream the different representations indicated in Table III, third row. This happens because there are not enough resources to satisfy all users' requests, only a fraction of nodes are placed in the desired overlay and the remaining ones must be redistributed among the other overlays. Moreover, the unsatisfied nodes will try, whenever possible, to move to a higher rate representation.

Figure 5 illustrates the comparison between the number of online nodes within each overlay obtained by simulation and the number provided by the numerical solution of

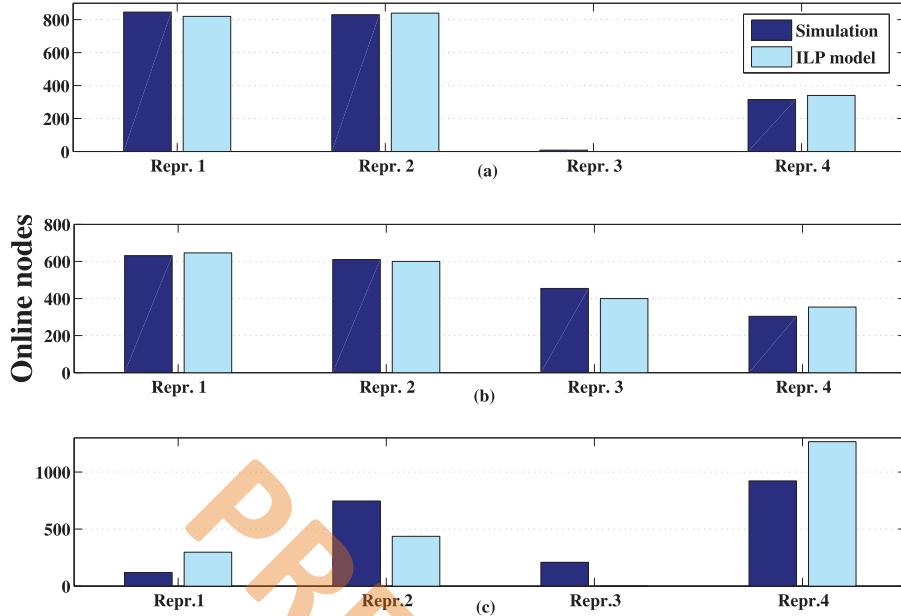


Fig. 5. Comparison between the P2P-DASH architecture and the ILP model in the examined scenarios: (a) conservative, (b) uniform, (c) aggressive.

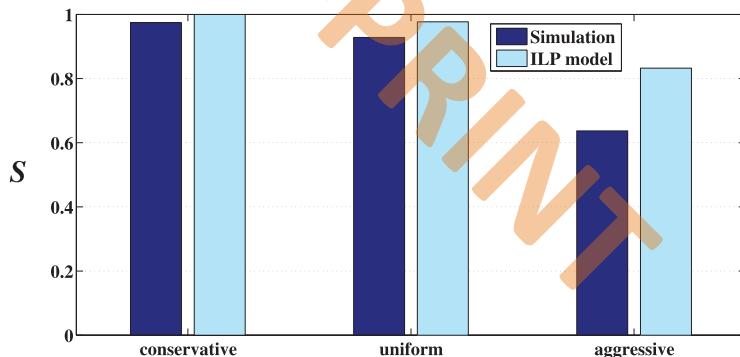


Fig. 6. System satisfaction normalized to the number of users.

the ILP problem, for the three examined scenarios. The number of online nodes within the simulated P2P-DASH system has been averaged over time (considering only the last 1500 s of the simulation) and also over 10 distinct simulation runs. The confidence intervals are not reported, as they are so small that they cannot be visually appreciated; the same also applies to all subsequent simulative outcomes. In the conservative and uniform scenarios, Figure 5(a) and (b) indicate that the two distributions are extremely close. This proves that our algorithm correctly controls the peer movements when system resources are not overly stressed. In the aggressive scenario, Figure 5(c) shows that the difference is more noticeable.

Figure 6 further shows the comparison between the system satisfaction S that the proposed P2P-DASH architecture attains and the satisfaction value provided by the numerical solution of the ILP problem for the reference system of Section 4; the reported values have been normalized with respect to $N = 2000$, the total number of

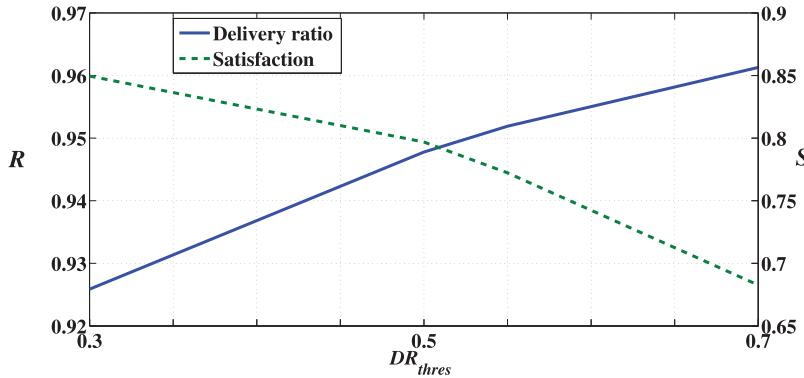


Fig. 7. Satisfaction and delivery ratio as a function of DR_{thres} .

users. Within the P2P environment, satisfaction has been monitored during every simulation run with a periodicity of 10s over the last 1500s of the streaming session, then these instantaneous values have been averaged over time and also over 10 distinct simulation runs. The figure shows that our system nearly satisfies as many nodes as the system modeled through the ILP approach in both the conservative and the uniform scenarios; this further corroborates the conclusion that the proposed solution is effective at exploiting resources when no critical shortage occurs. It does not, only when the most demanding, aggressive scenario is examined: however, as next figures will demonstrate, the proposed system succeeds at guaranteeing its users good local conditions, a goal the ILP model cannot pursue.

5.3. Tuning the Rate Control Parameters

From now onward, we will exclusively concentrate on the most challenging setting, which is the aggressive scenario, and we will begin tuning the thresholds that the rate control algorithm employs so as to optimize system performance. We first investigate the DR_{thres} effect on system behavior, and take as reference metrics the delivery ratio that the overlays exhibit, averaged over their population, and the system satisfaction. In order to have the downward migration process exclusively ruled by the comparison between $DR_i^{(t)}$, the value of the peer local delivery ratio, and the DR_{thres} value, we set $RW_{thres} = 1$ in Algorithm 1. Figure 7 accordingly shows the satisfaction S and the delivery ratio R that the system achieves, when varying the DR_{thres} threshold. The delivery ratio R that appears in the figure has been computed as the weighted average of the average delivery ratio of each overlay, where the weight associated to each value corresponds to the average number of users within the same overlay. We note that R increases as DR_{thres} increases: as a matter of fact, a greater value of the threshold translates into a looser constraint ruling the downward migration process. In other words, in every overlay other than the first a larger number of peers migrates to the lower overlay, while those who remain exhibit relatively high values of delivery ratio. On the other hand, being easier to move downward reflects into a lower chance for the peer to reside within the desired overlay, wherefrom a lower value of satisfaction S . It is therefore necessary to find a compromise value for DR_{thres} , that at the same time guarantees good values for the considered metrics. As Figure 7 suggests, we chose to work with $DR_{thres} = 0.5$, a tradeoff value that warrants a satisfying value for both. We next examine the influence of RW_{thres} , the threshold on the request window state, on the algorithm behavior and in turn onto the achievable system performance. This threshold acts on the local parameter $RWS^{(t)}$, that provides the peer with a forecast of the status of

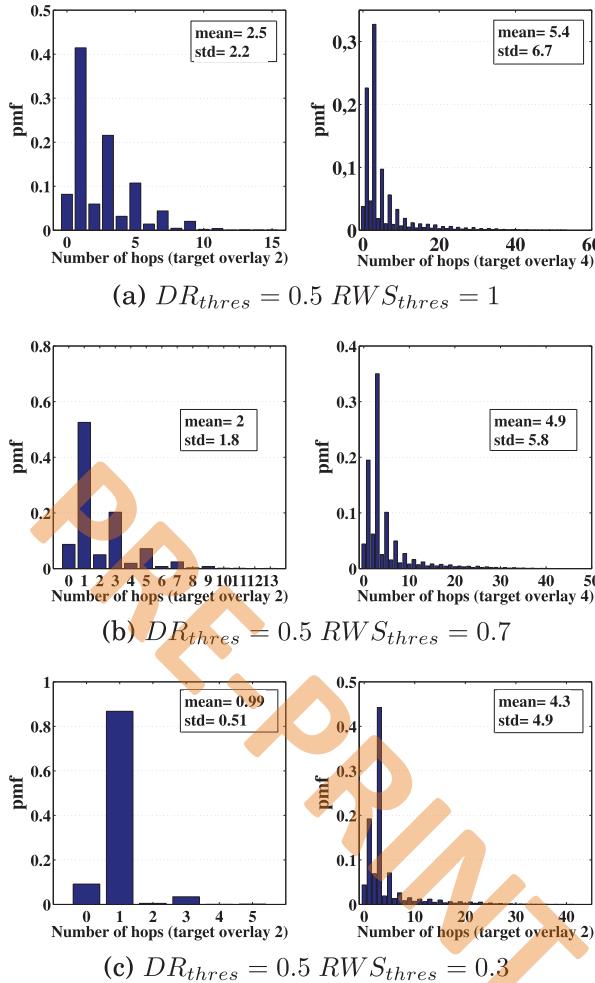


Fig. 8. pmf of the number of hops for peers targeting overlay 2 and overlay 4.

its playout buffer in the very next future. Whereas it is meaningful to consider the condition on the local delivery ratio alone, as we did before, we believe that the same does not hold for the condition on the request window state: as a matter of fact, the former offers a reliable snapshot of the current situation at the peer's site, the latter displays the intrinsic uncertainty of a projection, and this inevitably affects the reliability of the rate control decision process. On the other hand, to consider both conditions when deciding whether to downgrade can turn out beneficial, especially if RW_{thres} takes on a sufficiently modest value. In order to validate this statement, we begin by observing that both the delivery ratio R and the system satisfaction S are not heavily influenced by different choices for RW_{thres} : when decreasing RW_{thres} from 1 to 0.3, we observed that they modify from 0.95 to 0.94, and from 0.77 to 0.76, respectively. What is significantly affected by this threshold is the number of hops and the time each peer spends within the overlay that distributes its desired streaming rate. The number of hops is a crucial metric to monitor, as the viewing quality is heavily influenced by it [Yitong et al. 2013; Zink et al. 2003]. We therefore fix $DR_{thres} = 0.5$ and report in Figure 8 the probability mass function (pmf) of the number of hops that the generic peer makes

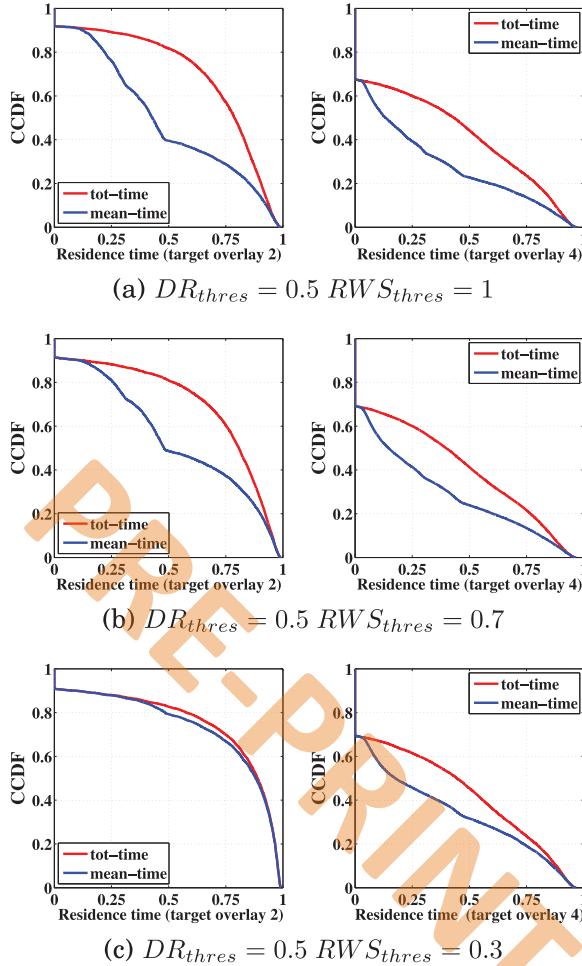


Fig. 9. CCDF of the normalized residence time.

from one overlay to another during its lifetime for three reference values of RW_{thres} , namely, $RW_{thres} = 1$ (that corresponds to bypassing the condition), $RW_{thres} = 0.7$ and $RW_{thres} = 0.3$. We built the pmf's on the basis of the target overlay that the peers aim at reaching, as a user targeting the overlay that distributes the highest representation will make the largest number of hops with the highest probability. In the examined aggressive setting, the two target overlays are overlay 2 and overlay 4. The figures on the left column refer to peers that wish to stream representation 2, the ones on the right column to peers that wish to stream representation 4. In detail, the comparison between Figure 8(a), (b), and (c) indicates that the introduction of the RWS condition reduces the number of the hops of nodes, the effect being substantial for the nodes aiming at representation 2; as an example, when $RW_{thres} = 0.3$ a peer makes only one hop with probability 0.87. A similar advantage, although not so impressive, is experienced by peers requesting representation 4: for this class of peers the average number of hops reduces from 5.4 to 4.3 and also the standard deviation decreases. Figure 9 depicts the complementary cumulative distribution function (CCDF) of the total time (red line) and the average consecutive time (blue line) the generic peer

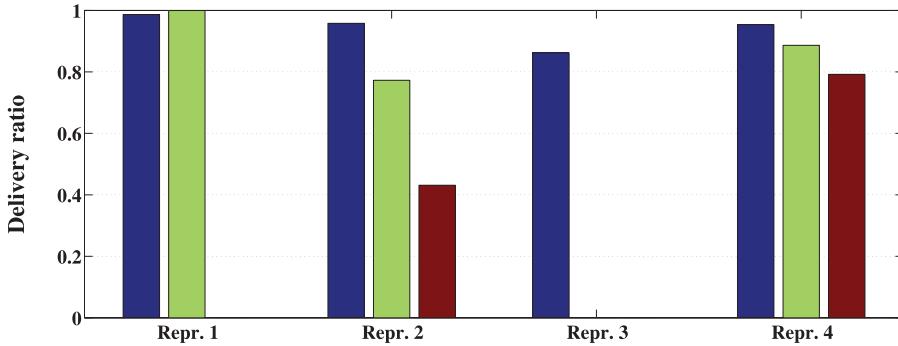


Fig. 10. Comparison among average delivery ratios: P2P-DASH architecture and systems with isolated swarms.

spends within the desired overlay, normalized to the peer’s life time. Figure 9(a) refers to the case where no $RWS^{(t)}$ control is exerted, and Figure 9(b) and (c) refer to the adoption of the threshold $RW_{thres} = 0.7$ and 0.3, respectively. We pleasingly observe that for the peers aiming at overlay 2 the two CCDF’s progressively get closer as RW_{thres} decreases: if RW_{thres} is sufficiently low, the probability that the peer spends more consecutive time within the desired overlay significantly increases. An additional threshold, E_{thres} , comes into play when the peer locally implements the rate control algorithm, but we deliberately postpone its investigation to Subsection 5.5, where we analyze system behavior in the presence of a flash crowd.

5.4. DASH-Unaware System Comparison

Having tuned the algorithm, we next proceed to demonstrate the effectiveness of our proposal. To this end, Figure 10 confronts the average delivery ratio that each overlay exhibits in the P2P-DASH architecture (blue bars) to the average delivery ratio of two DASH-unaware P2P systems made of isolated overlays. First, we consider a system where there is one overlay for every distinct streaming rate that peers might request. In this solution, nodes do not support DASH and therefore cannot dynamically move from one swarm to another; rather, they join and reside in the overlay that distributes the video at the bit rate they request. We term this solution as “ISO-desired” (red bars). In the third system we examine, peers join and reside in the overlay that is indicated by the solution of the ILP problem (see Figure 5(c)) to obtain the optimal value of global satisfaction; we refer to it as “ISO-ILP” (green bars). The values of delivery ratio have been computed as averages over the total number of active peers, and also over the streaming session duration, for 10 different simulation runs. For the P2P-DASH system, although not all peers belong to the desired overlay, Figure 10 indicates that on average they experience excellent values of delivery ratio, always greater than 0.94 except for the overlay distributing representation 3, which is a transition overlay where very few nodes reside. As for the “ISO-desired” system, only the two overlays distributing representation 2 and 4 are present and both display unsatisfactory values of the delivery ratio (as low as 0.43 for overlay 2 and about 0.79 for overlay 4). Observing the behavior of the “ISO-ILP” system, we can further outline that, except for overlay 1, the average delivery ratio of overlays 2 and 4 is lower than the one experienced by the corresponding overlays in the P2P-DASH system. As regards the playback delay, defined as the interval between the time when the generic video chunk is generated at the video server and the instant the same chunk is rendered at the peer’s site, the improvement that the P2P-DASH architecture achieves is even more significant: Figure 11 reveals that the average playback delay is approximately reduced by an

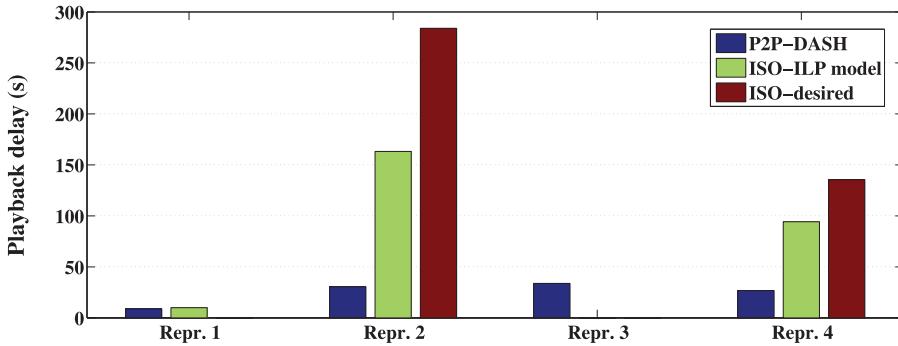


Fig. 11. Comparison among average playback delays: P2P-DASH architecture and systems with isolated swarms.

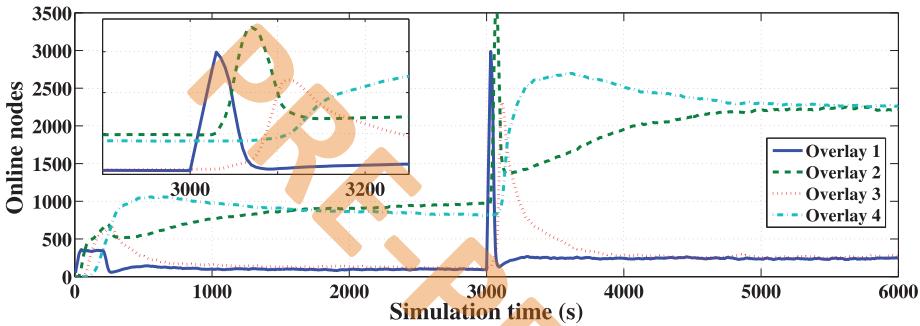


Fig. 12. Online nodes in the presence of a flash-crowd event.

order of magnitude. Overall, the comparison definitely plays in favor of the proposed architecture.

5.5. Flash-Crowd Event

Given dynamism is exactly the reason why adaptive streaming through DASH has been introduced, it is significant to explore how the examined system reacts to the presence of a flash crowd, so common in a P2P live streaming environment. In this subsection, we investigate how the proposed architecture reacts to the occurrence of a step join of new users that massively want to enter the system in a relatively short amount of time. We therefore overload the system, which has reached the steady-state condition and accommodates $N = 2000$ users, with $N' = 3000$ new incoming peers that begin entering the system at time $t_{FC} = 3000$ s for the next 30s. The same percentages and capacity values that originally described the composition of the peer population are applied to this bulk arrival of new users.

Figure 12 evidences the different tides of new peers moving from one overlay to the next: All users are forced to enter within the overlay that distributes the lowest representation of the video, and it is here that the number of online nodes first peaks; then they move forward, their wave investing overlay 2 and 3, to finally reach overlay 4. In less than 3 minutes, the P2P platform is able to react to this massive strain.

Within such framework, it is also crucial to discuss the relevance of the efficiency displayed by the system overlays as defined in (2) and employed by the rate control algorithm as a global indicator. To this end, we recall that in Algorithm 1 the upward migration of the peer is ruled by the combined observation of the current value of the

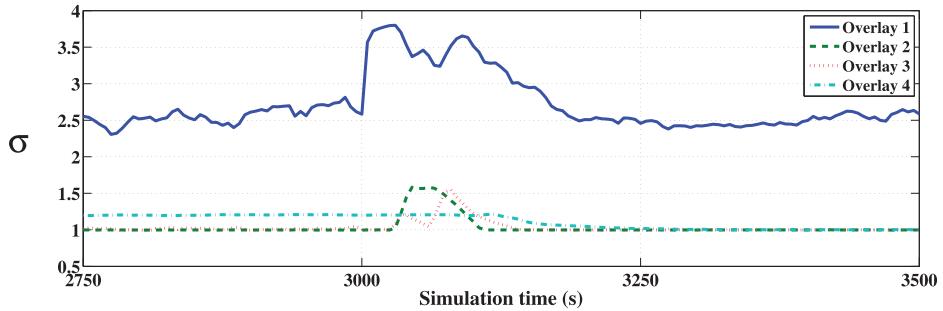


Fig. 13. Evolution of the resource index.

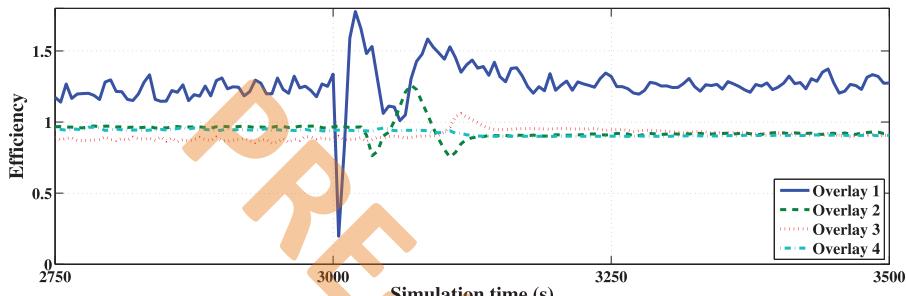


Fig. 14. Evolution of efficiency.

resource index and of the efficiency that the target overlay displays. The condition on the first indicator reflects the static resource balance, the condition on the latter points to the true capability that the target overlay owns to successfully deliver the video. In greater detail, Figure 13 indicates that the massive arrival determines a huge peak in $\sigma_1(t)$, the resource index of the overlay distributing the lowest streaming rate, as all new peers enter overlay 1 and all of them have an upload capacity c_i greater than or equal to r_1 ; the same trend, although on a reduced scale, is observed for the remaining overlays, except for the one distributing the video at the highest streaming rate, where on the contrary the majority of the joining peers have an upload capacity c_i lower than r_4 .

If in parallel we examine what happens to efficiency, then we notice from Figure 14 that the flash crowd occurrence immediately determines a sharp, negative peak in $E_1(t)$, which in turn reflects in $E_2(t)$: It is a manifest signal that a critical regime will soon appear. As a matter of fact, even a robust overlay can momentarily operate in critical conditions, and the reason is that its newcomers will temporarily act as free riders: They do require video content, without sharing any.

Figure 15 shows the behavior of the average delivery ratio (the average being computed over the peer population in every overlay) when the flash crowd occurs and compares the behavior of the system where the algorithm exerts no control on $E(t)$ (Figure 15(a)) to the system where the algorithm duly takes it into account (Figure 15(b)). We set the threshold to the conservative value $E_{thres} = 0.9$ to significantly slow down the upward migration during the abrupt step join. Figure 15(b) indicates that the negative spikes of the delivery ratio are either eliminated or markedly confined within all overlays except for overlay 1. It is also interesting to notice that the average delivery ratio of overlay 2 and 4 stays above 0.9, indicating that the video diffusion process is taking place in a satisfying manner and that in overlay 3 the delivery ratio does not

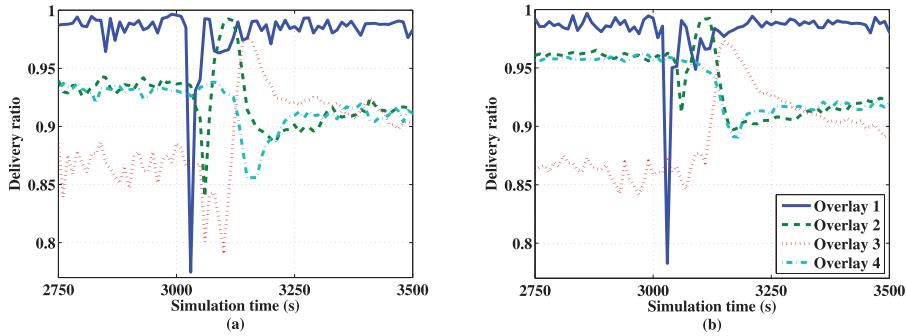


Fig. 15. Comparison between delivery ratios.

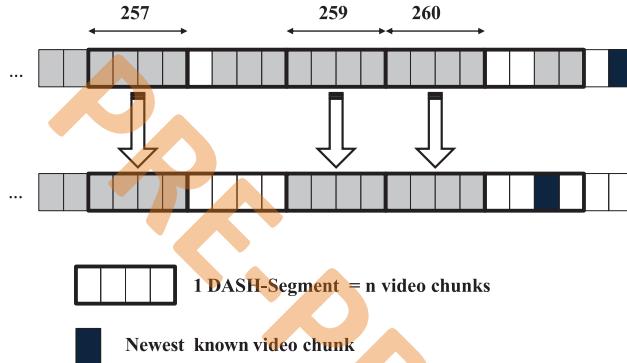


Fig. 16. DASH-segment inheritance in migration process.

even show any reduction after $t = t_{FC}$. As for the negative impact that the flash crowd has on overlay 1, it is unavoidable: This overlay represents the system entry point and as such the arrival of new peers is not influenced by the rate control algorithm. We conclude observing that if a pure client-server architecture had to guarantee an adequate performance in the presence of a flash crowd, then the server should feature an upload capacity of 15.5 Gbit/s, in order for all users of the aggressive scenario to download the desired video representation.

5.6. Buffer Content Reuse

The last numerical investigation we perform is centered on DASH availability to move from one representation to another with a time granularity which is equal to the size of a segment. Namely, we want to assess how this very peculiar feature can be profitably employed in a P2P-based platform, and for doing so we assume that when a peer moves from one overlay to another, it can successfully inherit all DASH segments currently present within its buffer. This content will contribute to avoid video stalls and it will ease the transition to the new video representation. Only completely received segments are inherited, and this statement is made clear by the example portrayed in Figure 16, where the segment with sequence number 258 is lost in the migration. We assume the duration of the DASH segment equal to 2s.

To have a sound term of comparison, we consider the reference case where the node joining the new overlay is forced to remove from its buffer all the video chunks of the representation it was previously receiving before collecting the video chunks from the new representation. We then evaluate the switching delay, that we define as the

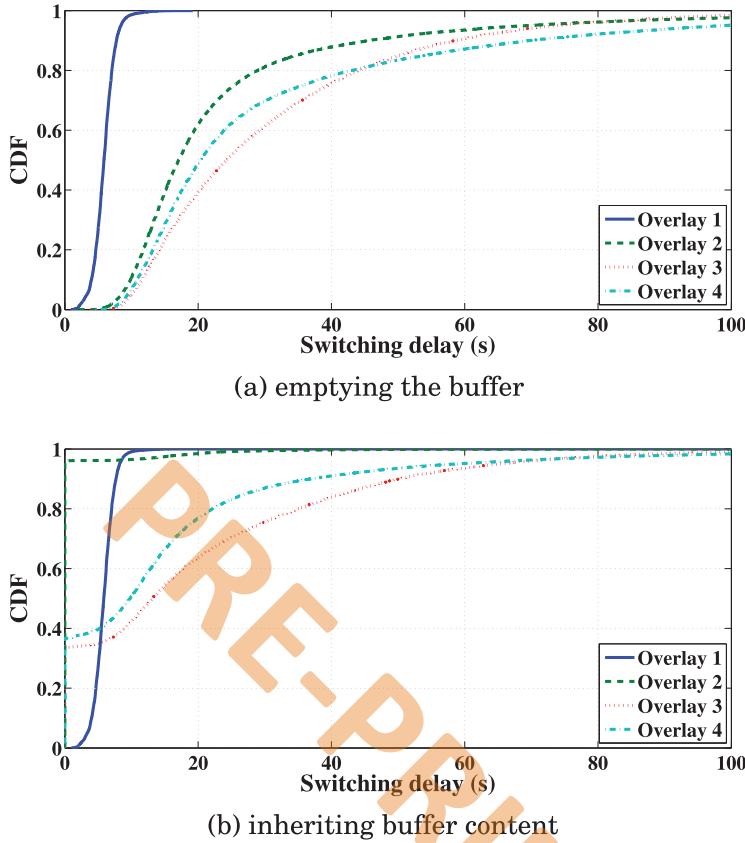


Fig. 17. Switching delay CDF.

time a peer needs to gather 8s of consecutive video chunks every time it migrates from its current overlay to a new one and compute this delay from the time the peer joins the new overlay. After a node collects 8s of video, it starts the playout; we believe 8s to be a reasonable duration to cope with possible bandwidth fluctuations.

Figure 17(a) reports the CDF of the switching delay in the different overlays, as numerically determined for the reference case. This figure shows that the delay increases with increasing values of the video streaming rate, that is, from overlay 1 to 2, 3, and 4 and quantitatively provides evidence that the choice to force every node to start from the lowest representation is correct, as it statistically guarantees the lowest latency. Next, in Figure 17(b), we consider the DASH-supported solution where the peer scans its buffer to evaluate if it can reuse some of its video content, that is, the completely received DASH segments. From the comparison with Figure 17(a) we notice that the delay for overlay 1 does not change, as it is modestly influenced by the migration of the peers; on the contrary, DASH guarantees a considerable reduction of the delay for the remaining overlays. As an impressive example, the time needed to accumulate 8s of video is 0s with probability 0.95 when the peer moves to the overlay that distributes representation 2, meaning that with high probability a significant portion of the current buffer content is profitably employed during the migration process. By visual inspection of the two figures, we conclude that a similar improvement is observable when the peer moves to the overlays distributing representations 3 and 4.

6. CONCLUSIONS

This article has proposed a P2P-DASH architecture that jointly exploits the cooperation property of P2P and the flexibility of DASH to stream good-quality videos to a large population of users. Each peer within the system relies on a decentralized rate control strategy that steers its rate variations, and hence its movements from one overlay to another, on the basis of local and global indicators reflecting the health status of the single peer and of the system overlays. The effectiveness of the proposed solution has been demonstrated through simulations, indicating that the P2P-DASH platform is able to guarantee its users a very good performance; its overlays operate in much better conditions than the overlays of a conventional DASH-unaware architecture subject to the same streaming requests and relying on the same availability of peers' upload capacity. Moreover, through a comparison with a reference system modeled via an integer linear programming problem, it has been shown that our system also outperforms such reference architecture. Finally, system behavior has been investigated in the critical condition of a flash crowd, demonstrating that the harsh and rapid input of a large number of new peers can be successfully revealed and gradually accommodated.

REFERENCES

- Adobe. 2015. The HTTP Dynamic Streaming Website. Retrieved from <http://www.adobe.com/it/products/hds-dynamic-streaming.html>.
- Akamai. 2014. State of the Internet Q2-2014. Tech. Rep. (2014). <http://www.akamai.com/stateoftheinternet/>.
- Apple. 2015. The HTTP Live Streaming Website. (2015). <https://developer.apple.com/streaming/>.
- O. Babaoglu and M. Marzolla. 2014. The people's cloud. *IEEE Spect.* 51, 10 (October 2014), 50–55. DOI: <http://dx.doi.org/10.1109/MMUL.2011.71>
- X. Zhang, C. Zhou, C. W. Lin and Z. Guo. 2014. A control-theoretic approach to rate adaption for DASH over multiple content distribution servers. *IEEE Trans.actions on Circ. Syst. Vid. Technol.* 24, 4 (April 2014).
- Cisco WP. 2015. White Paper: Cisco Visual Networking Index: Forecast and Methodology, 2014–2019. White Paper. (May 2015).
- DASH-IF 2014. DASH Industry Forum. Retrieved from <http://dashif.org>.
- European Commission 2012. Digital Agenda for Europe Scoreboard 2012. Directorate-General for Communication Networks, Content and Technology (CONNECT). (Jun. 2012).
- M. Klusch, P. Kapahnke, X. Cao, B. Rainer, C. Timmerer, and S. Mangold. 2014. MyMedia: Mobile semantic peer-to-peer video search and live streaming. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS'14)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, 277–286. DOI: <http://dx.doi.org/10.4108/icst.mobiquitous.2014.258026>
- S. Lederer, C. Muller, and C. Timmerer. 2012. Towards peer-assisted dynamic adaptive streaming over HTTP. In *Proceedings of the 19th International Packet Video Workshop (PV)*. 161–166. DOI: <http://dx.doi.org/10.1109/PV.2012.6229730>
- Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran. 2014. Probe and adapt: Rate adaptation for HTTP video streaming at scale. *IEEE J. Select. Areas Commun.* 32, 4 (Apr. 2014), 719–733.
- C. Liu, I. Bouazzi, and M. Gabbouj. 2011. Rate adaptation for adaptive HTTP streaming. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems (MMSys'11)*. ACM, New York, NY, 169–174. DOI: <http://dx.doi.org/10.1145/1943552.1943575>
- Microsoft. 2015. Smooth Streaming. Retrieved from <http://www.microsoft.com/silverlight/smoothstreaming/>.
- R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang. 2012. QDASH: A QoE-aware DASH system. In *Proc. of ACM MMSys'12*. ACM, New York, NY, 11–22.
- Orange. 2014. Orange offres internet. Retrieved from <http://www.orange.fr>.
- I. Sodagar. 2011. The MPEG-DASH standard for multimedia streaming over the internet. *IEEE Multimed.* 18, 4 (April 2011), 62–67. DOI: <http://dx.doi.org/10.1109/MMUL.2011.71>
- D. Suh, I. Jang, and S. Pack. 2014. QoE-enhanced adaptation algorithm over DASH for multimedia streaming. In *Proceedings of the 2014 International Conference on Information Networking (ICOIN)*. 497–501. DOI: <http://dx.doi.org/10.1109/ICOIN.2014.6799731>
- Swisscom. 2014. Swisscom internet offers. Retrieved from <http://www.swisscom.com>.

- G. Tian, Y. Xu, Y. Liu, and K. Ross. 2013. Mechanism design for dynamic P2P streaming. In *Proceedings of the 2013 IEEE 13th International Conference on Peer-to-Peer Computing (P2P)*. IEEE, Los Alamitos, CA, 1–10. DOI:<http://dx.doi.org/10.1109/P2P.2013.6688702>
- T. Net. 2014. T.Net Connnettivitá e voce. Retrieved from <http://www.tnet.it>.
- D. Wu, C. Liang, Y. Liu, and K. Ross. 2009. View-upload decoupling: A redesign of multi-channel P2P video systems. In *IEEE INFOCOM*. IEEE, Los Alamitos, CA, 2726–2730. DOI:<http://dx.doi.org/10.1109/INFCOM.2009.5062220>
- L. Yitong, S. Yun, M. Yitian, L. Jing, L. Qi, and Y. Dacheng. 2013. A study on quality of experience for adaptive streaming service. In *2013 IEEE International Conference on Communications (ICC)*. IEEE, Los Alamitos, CA, 682–686. DOI:<http://dx.doi.org/10.1109/ICCCW.2013.6649320>
- A. Zambelli. 2009. *IIS Smooth Streaming Technical Overview*. Technical Report. Microsoft Corporation.
- M. Zhang, Q. Zhang, L. Sun, and S. Yang. 2007. Understanding the power of pull-based streaming protocol. *IEEE J. Select. Areas Commun.* 25, 9 (Dec. 2007), 1678–1694.
- M. Zink, O. Künzel, J. Schmitt, and R. Steinmetz. 2003. Subjective impression of variations in layer encoded videos. In *Proceedings of the 11th International Conference on Quality of Service (IWQoS'03)*. Springer-Verlag, Berlin, 137–154.

Received November 2015; revised March 2016; accepted March 2016

PRE-PRINT