How Much Can Large-Scale Video-on-Demand Benefit From Users' Cooperation?

Delia Ciullo, Valentina Martina, Michele Garetto, *Member, IEEE*, and Emilio Leonardi, *Senior Member, IEEE*

Abstract—We propose an analytical framework to tightly characterize the scaling laws for the additional bandwidth that servers must supply to guarantee perfect service in peer-assisted Video-on-Demand systems, taking into account essential aspects such as peer churn, bandwidth heterogeneity, and Zipf-like video popularity. Our results reveal that the catalog size and the content popularity distribution have a huge effect on the system performance. We show that users' cooperation can effectively reduce the servers' burden for a wide range of system parameters, confirming to be an attractive solution to limit the costs incurred by content providers as the system scales to large populations of users.

Index Terms—Cooperative networking, stochastic models, Video-on-Demand.

I. INTRODUCTION

ACCORDING to Cisco [1], by the end of 2016, the sum of all forms of Internet video [TV, Video-on-Demand (VoD), P2P] will be approximately 86% of global consumer traffic. In particular, the traffic component due to Video-on-Demand is expected to triple from 2011 to 2016, reaching the equivalent of 4 billion DVDs per month.

Increasing traffic volumes force video providers to continuously upgrade the content delivery network (CDN) infrastructure that feeds the contents to local ISPs. To partially alleviate this burden, a recent trend of VoD providers is to exploit cloud services, which permit fine-grained resource reservation [2]. As an example, in 2010, Netflix decided to migrate its infrastructure into the Amazon EC2 cloud, as it could not build data centers fast enough to keep pace with growing demand.

However, any solution based on CDNs has severe limitations in terms of scalability. CDNs can significantly reduce the traffic in the Internet core and improve the user-perceived performance (e.g., by reducing the latency) by "moving" contents close to the users. Nevertheless, the aggregate resources required at data centers (bandwidth/storage/processing), and the corresponding costs incurred by content providers, inevitably scale linearly with the user demand and data volume.

D. Ciullo is with the EURECOM Sophia Antipolis, 06410 Biot, France (e-mail: delia.ciullo@gmail.com).

V. Martina and E. Leonardi are with the Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, 10129 Turin, Italy.

M. Garetto is with the Dipartimento di Informatica, Università di Torino, 10149 Turin, Italy

The only scalable solution proposed so far is to exploit the peer-to-peer paradigm, according to which users contribute their resources (bandwidth/storage/processing) to the system while they use it [3], [4]. Although the peer-assisted approach is an attractive solution to the scalability problem, and it has already been experimented in several applications [5]–[8], it brings with it several issues that tend to discourage its adoption by many content providers: the unpredictable nature of users' cooperation, the added complexity on the control plane due to signaling and chunk scheduling, and the need to provide incentive mechanisms to the users [9].

Streaming architectures that primarily rely on users' cooperation can hardly guarantee the strict quality-of-service requirements of online video, where a steady download rate no smaller than the video playback rate is necessary for a smooth watching experience, and any interruption tends to be very annoying to the user [10].

For these reasons, we argue that peer-assisted architectures should be supported by properly dimensioned CDNs (or cloud services) that intervene whenever the resources provided by users are not enough to satisfy the current demand. In our theoretical work, we are specifically interested in characterizing the additional bandwidth that servers must supply to guarantee ideal service to all users (i.e., requests are immediately satisfied, and videos can be watched without interruptions until the end). Our main contribution is a stochastic analytical framework that allows to derive general upper and lower bounds to the bandwidth requested from the servers in a peer-assisted VoD system, capturing essential aspects such as peer churn, bandwidth heterogeneity, and Zipf-like video popularity. Our analysis permits to tightly characterize the system performance as the number of users (and the number of available videos) grows large, and thus assess the scalability of large-scale VoD exploiting users' cooperation.

In our previous work [11], we have considered the case of a single video, providing for the first time an asymptotic characterization of the servers' bandwidth as the number of watchers increases. Here, we extend the analysis to a multivideo system, in which users can browse a catalog of available contents and asynchronously issue requests to watch videos.

Our main contribution is a precise definition of the conditions (related to physical system parameters such as the growth rate of the catalog size, the Zipf's exponent of video popularity, videos' characteristics, and user behavior) under which the additional bandwidth requested from the servers asymptotically goes to zero as the size of the system grows large. When such conditions are not met, we provide the asymptotic laws for the required servers' bandwidth.

We consider both the cases in which users can only assist the distribution of the last video they have selected (we call this

the *passive* system because the utilization of peer resources is tied to the video popularity distribution, which is not under the system control) and the general case in which users can assist the distribution of any video (we call this the *active* system, also referred to in the literature as *universal streaming*). For the *active* system, we also devise the resource allocation strategies that permit to achieve the optimal theoretical performance.

We emphasize that a full exploitation of peers' upload bandwidth is not a trivial task in the presence of high degrees of peer churn (i.e., when users tend to abandon the system after watching a few videos), in consideration of the obvious fact that users can only upload data that they have previously downloaded. For the same reason, unpopular videos, which tend to be scarcely replicated among peers, can pose a significant stress on the system. Hence, another important contribution of our work is the definition of suitable strategies to mitigate the joint impact of peer churn and heterogeneous video popularity.

II. SYSTEM ASSUMPTIONS

A. Service Specification and Users Cooperation

We model a VoD system where users run applications that allow them to browse an online catalog of videos. When a user selects a video, we assume that the request is immediately satisfied and the selected video can be watched uninterruptedly until the end, i.e., the system is able to steadily provide to the user a data flow greater than or equal to the video playback rate. We consider that users watch at most one video at a time.

We assume that the system catalog contains K different videos. Video k, $1 \le k \le K$, is characterized by the following: its size $l_k \in [l_{\min}, l_{\max}]$, expressed in bytes; a selection probability p_k, which is the probability that a user selects video k among all videos in the catalog; and a minimum playback rate. We assume that video k is downloaded at constant rate greater than or equal to the minimum playback rate. Specifically, we denote by d_k the download rate, where $d_k \in [d_{\min}, d_{\max}]$ (in bytes/second).

Users contribute with their available upload bandwidth to the video distribution: They can retrieve part of a requested video (or even the entire video) from other users, saving servers resources.

We model the amount of available upload bandwidth at a given time by a user by a random variable U with cumulative distribution function F_U and mean $\overline{U}$. In this way, we can take into account effects related to Internet access heterogeneity and cross-traffic fluctuations. The random variables U's denoting the upload bandwidths of the users are assumed to be i.i.d.

Users contribute to the system also a limited amount of storage capacity. The exact amount of buffer space available at each user is not important in our analysis. As a minimum requirement, our schemes assume that users can store at least one whole video in addition to the one currently played out.

B. User Dynamics

Users join the system when they request the first video. We denote by λ^u the arrival rate of new users. While they are in the system, users can be in two states: $\{contributing, \ sleeping\}$. The *contributing* state is defined as the state in which a user is contributing its upload bandwidth to the system. In the *contributing* state, a user can download (and watch) video contents. Notice that a user can be contributing its upload bandwidth

even if it is not currently downloading/watching any video, but simply because it keeps its VoD application up and running.

During the *sleeping* phase, the user's application is not running, hence it is neither downloading nor uploading data. We assume that users download the entire requested videos (aborted downloads could be easily included in our model, but we have preferred not to do so for simplicity). Note that since a video is retrieved at constant rate, its download time, $\tau_k = l_k/d_k$, is a deterministic attribute of video k, taking values in range $[\tau_{\min} = l_{\min}/d_{\max}, \tau_{\max} = l_{\max}/d_{\min}]$. After completing a download, users remain in the *contributing* state for a random amount of time T_{seed} with mean $\overline{T}_{\text{seed}}$ (part of this time can be spent finishing watching the video, if the download rate is larger than the playback rate). Then, they transit to the *sleeping* state, where they stay for a random amount of time of mean $\overline{T}_{\text{sleep}}$. Users can choose to abandon the system (i.e., to stop the VoD application and never open it again) after watching just a single video. We assume that, after watching a video, each user independently decides to leave the system with probability p_{out}. It follows that the number of videos requested by a user is geometrically distributed with mean $m = 1/p_{\text{out}}$. Moreover, the average time spent by a user in the system can be computed as $\overline{T} = m \cdot (\sum_{k=1}^{K} p_k \tau_k + \overline{T}_{\text{seed}} + \overline{T}_{\text{sleep}})$.

From the above assumptions, and the fact that the system provides guaranteed service, the set of videos requested by a user, the total time spent by a user in the system, as well as the amounts of time spent by a user in the *contributing/sleeping* states are independent from user to user.

C. System Scaling

Our goal is to asymptotically characterize the average additional bandwidth $\overline{S}$ that servers must supply to guarantee perfect service to all users, as the system grows large. Let n be the average number of users in the system. By Little's law, we have $n = \lambda^u \overline{T}$. Note that $\overline{T}$ is a constant, hence our asymptotic analysis for increasing number of users is performed by letting λ^u (and thus n) go to infinite.

Since the catalog size is expected to grow, just like the number of users, we consider that the number K of videos available in the catalog is tied to the number of users, according to the law $K = \Theta(n^\beta)$, with[1] $\beta \le 1$.

As the system grows, new videos are made available to the users. We assume that the characteristics of new videos inserted into the catalog, in terms of file size l_k and download rate d_k, are random. Hence, l_k and d_k should be regarded as instances of i.i.d. random variables L_k and D_k, respectively, with assigned distributions (possibly correlated). Recall from Section II-A that we (reasonably) assume that the distributions of L_k, D_k have finite support independent of n.

D. Content Popularity

To specify the selection probabilities of videos, we need to model the relative popularity of the videos in the catalog. For this, we adopt the standard Zipf's law, which has been frequently observed in traffic measurements and widely adopted in performance evaluation studies [12], [13]. More specifically, having

---

[1]Note that the assumption $\beta \le 1$ is not particularly restrictive, as long as the number of contents introduced into the catalog by each user can be bounded by a constant. Indeed, in this case the total catalog size scales at most linearly with the number of users.

sorted the videos in decreasing order of popularity, a request is directed to video k with probability

$$p_k \triangleq \frac{H(K)}{k^\alpha}, \qquad 1 \le k \le K \qquad (1)$$

where α is the Zipf's law exponent, and $H(K) \triangleq (\sum_{i=1}^{K} i^{-\alpha})^{-1}$ is a normalization constant. Depending on the exponent α, we have

$$H(K) = \begin{cases} \Theta(1), & \text{if } \alpha > 1 \\ \Theta((\log K)^{-1}), & \text{if } \alpha = 1 \\ \Theta(K^{\alpha-1}), & \text{if } \alpha < 1. \end{cases} \qquad (2)$$

Let Λ be the aggregate rate at which users request videos. By construction, $\Lambda = \lambda^u m$. The rate at which a specific video k is requested is $\lambda_k = \Lambda p_k$.

Recalling that $\Lambda = \Theta(n)$, and using (2), we have that $\lambda_k = \Theta(nH(K)k^{-\alpha})$, and more precisely

$$\lambda_k = \begin{cases} \Theta(nk^{-\alpha}), & \text{if } \alpha > 1 \\ \Theta(n(k \log n)^{-1}), & \text{if } \alpha = 1 \\ \Theta(n^{1+\beta(\alpha-1)}k^{-\alpha}), & \text{if } \alpha < 1. \end{cases} \qquad (3)$$

In this work, we ignore the evolution of contents' popularity over time by invoking a timescale separation principle, i.e., by assuming that popularity dynamics occur at a timescale much slower than contents' download time, which is fairly acceptable in many cases of practical interest.

E. System Load

For a given system catalog, i.e., for given video characteristics $\{d_k\}_k$ and $\{l_k\}_k$, we can compute a fundamental quantity γ characterizing the global system load (i.e., the load induced by all videos)

$$\gamma \triangleq \frac{\sum_{k=1}^{K} p_k l_k}{\sum_{k=1}^{K} p_k \overline{U} (\tau_k + \overline{T}_{\text{seed}})}. \qquad (4)$$

Indeed, consider a large time interval Δ. During this time interval, a video k will be requested on average $\lambda_k \Delta$ times. Each request for video k has a double effect on the system: It requires an amount of bytes l_k to be downloaded; it lets the requesting user potentially to upload an average amount of data $\overline{U} (\tau_k + \overline{T}_{\text{seed}})$. The ratio between the average amount of downloaded data and the average amount of uploaded data during interval Δ, for $\Delta \to \infty$, leads to the expression in (4).

We remark that (4) holds for both *passive* and *active* systems introduced in Section I. However, in the case of *active* systems, it does not account for the additional data that users might be instructed to download by the system (data bundling). The effect of bundling on the system load will be considered later. Borrowing the terminology adopted in previous work [3], [14], we say[2] that the system operates in *deficit* mode if $\gamma > 1$, and in *surplus* mode if $\gamma < 1$.

We emphasize that since video characteristics are random, γ should be itself interpreted as an instance of a random variable Γ obtained deconditioning (4) with respect to $\{d_k\}$ and $\{l_k\}$.

We will also use a video-specific notion of load, denoted by γ_k, and its corresponding random variable Γ_k

$$\gamma_k \triangleq \frac{d_k \tau_k}{\overline{U} (\tau_k + \overline{T}_{\text{seed}})}. \qquad (5)$$

[2]In this paper, we do not consider the special case $\gamma = 1$.

TABLE I
NOTATION

Symbol	Definition
λ^u	user's arrival rate
$\overline{T}$	average time spent by users in the system
n	average number of users
K	catalog size (number of available videos)
β	scaling exponent of $K = n^\beta$
Λ	aggregate video request rate
λ_k	request rate of video k
d_k	download rate of video k
τ_k	download time of video k
$\overline{U}$	average user upload bandwidth
$\overline{T}_{\text{seed}}$	average time spent in the *contributing* state after downloading a video
$\overline{N}_{d,k}$	average number of users downloading video k
$\overline{N}_{\text{seed},k}$	average number of *seeds* for video k
$\overline{S}$	average bandwidth requested from the servers
γ	system load
γ_k	load associated to video k
m	average number of videos requested by a user

TABLE II
AVERAGE BANDWIDTH REQUESTED FROM THE SERVERS, $\overline{S}$

Conditions	Upper bound		Lower bound
	Passive system $\mathbb{P}(\Gamma_k < 1 - \sigma) = 1$	*Active* system $\mathbb{P}(\Gamma < 1 - \sigma) \to 1$	
$(\alpha \le 1 \wedge \beta < 1)$ $\vee (\alpha > 1, \beta < 1/\alpha)$	$o(1)$	$o(1)$	0
$\alpha > 1 \wedge 1/\alpha < \beta < 1$	$O(n^{1/\alpha})$	$o(1)$	0
$\alpha > 1 \wedge \beta = 1$	$O(n^{1/\alpha})$	$O\left(n^{2-\alpha}(\log n)^{\frac{\alpha-1}{1-\delta}}\right)$	$\Omega(n^{2-\alpha})$
$\alpha \le 1 \wedge \beta = 1$	$O(n)$	$O(n)$	$\Theta(n)$

We observe that γ_k would coincide with γ if all γ_k were equal. With abuse of language, we say that a video is in *deficit* mode if $\gamma_k > 1$, and in *surplus* mode if $\gamma_k < 1$.

Table I summarizes the notation introduced so far.

III. SUMMARY OF RESULTS

First, we observe that, in the worst possible case, the servers have to transmit at rate $d_{\max}$ to all downloading users. It follows that a trivial upper bound to the bandwidth requested from the servers is $\overline{S} = O(n)$. A trivial lower bound is $\overline{S} \ge 0$.

For the *passive* system, we obtain the following results. If the probability to include in the catalog a video with load $\gamma_k > 1$ is greater than zero, i.e., $\mathbb{P}(\Gamma_k > 1) > 0$, we have $\overline{S} = \Theta(n)$. If, instead, there exists an arbitrarily small constant σ such that $\mathbb{P}(\Gamma_k < 1 - \sigma) = 1$, we obtain the asymptotic upper bounds reported in the second column of Table II, which depend on the Zipf's exponent α and the catalog growth rate exponent β.

For the *active* system, we obtain the following results. If $\Gamma > 1$ with nonvanishing probability as the system size increases, we have $\overline{S} = \Theta(n)$. If $\Gamma < 1 - \sigma$ (w.h.p.),[3] for some $\sigma > 0$, we obtain the asymptotic upper bounds reported in the third column of Table II, which depend on the Zipf's exponent α and the catalog growth rate exponent β, while δ is an arbitrarily small positive number.

The fourth column of Table II reports corresponding lower bounds for $\overline{S}$, which are valid also for the extreme case in which the user upload bandwidth is arbitrarily large.

In Fig. 1, we show a graphical representation of our upper bounds, reporting $\overline{S}$ (on a $\log_n$ scale) versus the parameter α.

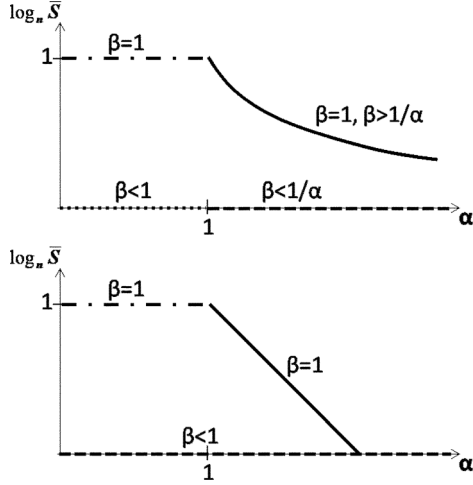[3]With high probability, i.e., with probability that tends to 1 as $n \to \infty$.

Fig. 1. $\log_n \overline{S}$ versus the parameter α: *(top plot)* passive system and *(bottom plot)* active system.

Fig. 2. Open network of $\cdot/G/\infty$ queues modeling users' dynamics.

The different line types (accompanied with labels) refer to different values assumed by the parameter β: The top plot refers to the passive system, while the bottom plot refers to the active one.

Our results for the *active* system provide the following fundamental insights: If $\beta < 1$, i.e., if the number of contents in the system scales sublinearly with respect to the average number of users, an active system operating (globally) in surplus mode can asymptotically eliminate the need of additional bandwidth from the servers (i.e., $\overline{S}$ tends to zero as the number of users increases), for any value of α. This can be done even under the sequential delivery scheme (i.e., when downloading users can only help future downloaders of the same file). We remark that the only requirement to achieve this desirable behavior is that the global system load is smaller than one, which does not imply that all videos are individually in the surplus mode.

If, instead, $\beta = 1$, (i.e., when the number of contents in the system scales as fast as the number of users),[4] the exploitation of users' cooperation is more difficult and depends on the Zipf's exponent: For $\alpha \leq 1$, we cannot do any better than the worst case $\overline{S} = \Theta(n)$. For $\alpha > 1$ (and $\Gamma < 1 - \sigma$), we approximately need $n^{2-\alpha}$ additional servers' bandwidth (notice that in this case upper and lower bounds differ only by a poly-log term), which unfortunately goes to infinite for any $\alpha < 2$. Basically, this is due to the fact that, for $\beta = 1$, there are too many contents available in the system (whose aggregate data volume becomes comparable to the total buffer space available at users). In this case, contents cannot be distributed/replicated at peers in such a way that the distribution of all of them can be effectively assisted by the users, considering also the effect of peer churn.

Passive systems perform, obviously, worse than active systems. First of all, they can lead to something better than $\overline{S} = \Theta(n)$ only when all videos are in surplus mode, which is a rather restrictive condition. Nevertheless, if this condition is satisfied, for $\beta < 1$ we still obtain that $\overline{S} = o(1)$, provided that $\alpha \leq 1$ or $\beta < 1/\alpha$. Instead, the servers' bandwidth goes to infinite as $n^{1/\alpha}$ for $\beta > 1/\alpha$, and the same occurs if $\beta = 1$ (of course without exceeding the worst case $\overline{S} = \Theta(n)$).

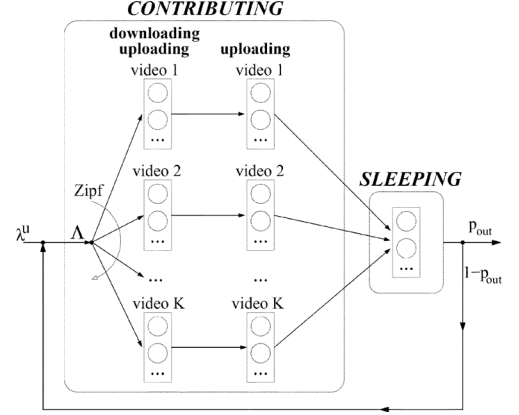[4]Our results could be extended to the case $\beta > 1$, reaching identical conclusions.

IV. PASSIVE SYSTEM

We start considering the *passive* system, in which users are constrained to assist only the distribution of the last selected video. This means that, after requesting a video, they can only download/upload data belonging to the selected video (until they request a new content from the catalog). A passive system is conceptually simple to implement and manage, since swarms of different videos are decoupled, and can be controlled independently of each other.

A. Preliminaries

We can describe the dynamics of users in the system by the open queueing network illustrated in Fig. 2. We consider a separate queue for all users downloading the same video. When the download is complete, users who keep the application running continue contributing their uploading bandwidth to the system, transiting to queues arranged in the second column of the network. Users who stop the application transit to the *sleeping* state, represented by a single queue on the right-hand side (r.h.s.).

Lemma 1: At any time, the number of users who are downloading a given video, the number of users who remain in the *contributing* state after downloading a video, and the number of users in the *sleeping* state follow independent Poisson distributions.

Proof: The dynamics of users in the open queueing network (in terms of transitions among the queues and sojourn times at queues) are decoupled since users behave independently of each other.[5] The resulting queueing network admits a product-form solution by the BCMP theorem. Since all queues have infinite servers, the numbers of users in the queues follow independent Poisson distributions. ∎

Let $N(t)$ be the total number of users in the system at time t. Note that $N(t)$ is itself Poisson distributed, with mean $n = \lambda^u \overline{T}$. We denote by $\overline{N}_{d,k} = \lambda_k \tau_k$ the average number of users downloading file k, and by $\overline{N}_{\text{seed},k} = \lambda_k \overline{T}_{\text{seed}}$ the average number of users remaining in the *contributing* state after downloading file k. In a *passive* system, $\overline{N}_{\text{seed},k}$ represents also the average number of users acting as seeds for video k.

---

[5]Notice that here we are not considering as part of user dynamics the data downloaded/uploaded by a user, which obviously depend on which videos the other users have requested.

B. Asymptotic Results for Single-Video System

Before considering the bandwidth requested from the servers to support the distribution of all videos available in the catalog, we analyze the simple case in which there is just one video (i.e., $K = 1$), whose request rate λ tends to infinite. Notice that in this case γ in (4) equals γ_1 in (5).

The following theorem characterizes how the servers' bandwidth $\overline{S}$ scales when $\lambda \to \infty$:

Theorem 1: Assume the following properties hold for U.
  i) $\overline{U} > 0$.
  ii) $\mathbb{E}[e^{\theta U}]$ is finite in a neighborhood of the origin.
  iii) $F_U(w) > 0$ for every $w > 0$.
If $\gamma < 1 - \sigma$ for some $\sigma > 0$, uniformly over $\overline{U}$, d, and γ, a $\lambda_0 > 0$ can be found such that

$$\overline{S} \leq 4\lambda^\delta e^{-C_1(\overline{U}, \tau, \gamma')\lambda^{1-\delta}} \qquad \forall \delta \in (0, 1) \text{ and } \lambda > \lambda_0 \quad (6)$$

with $\gamma' = \gamma(1 + \sigma/2)$, and $C_1(\overline{U}, \tau, \gamma') = \tau \frac{\overline{U}(1-\gamma')(1+\gamma')}{4\gamma'}$. A detailed proof of Theorem 1 is reported in Appendix A.

The upper bound stated in Theorem 1 is valid for $\lambda \to \infty$. When λ remains finite or vanishes as n grows large, we can apply the following rough bound based on the pessimistic assumption that the entire bandwidth necessary to sustain the downloads is provided by the servers (i.e., neglecting the contribution of seeds and simultaneously downloading users).

Lemma 2: A universal upper bound to the bandwidth requested from the servers is $\overline{S} \leq d\tau\lambda$.

Combining Theorem 1 and Lemma 2, we get the following corollary.

Corollary 1: Under the assumptions on U of Theorem 1, if $\gamma < 1 - \sigma$, for some $\sigma > 0$, then uniformly over $\overline{U}$, d, and γ a $\lambda_0 > 0$ can be found such that

$$\overline{S} \leq \begin{cases} d\tau\lambda, & \text{if } \lambda < \lambda_0 \\ 4\lambda^\delta e^{-C_1(\overline{U}, \tau, \gamma')\lambda^{1-\delta}} \ \forall \delta \in (0, 1), & \text{if } \lambda \geq \lambda_0 \end{cases} \quad (7)$$

with $\gamma' = \gamma(1 + \sigma/2)$.

C. Asymptotic Results for Multivideo System

Corollary 1 can be readily exploited to compute the aggregate bandwidth requested from the servers in the case of multivideo systems. Indeed, we basically have to add up the contributions of individual videos (as predicted by Corollary 1) to obtain an upper bound on the overall bandwidth requested from the servers. Thus, we can divide the video catalog into two portions: The *hottest* portion of the catalog comprises videos whose request rate tends to infinite as $n \to \infty$; the *coldest* portion of the catalog (which could be empty) comprises videos whose request rate remains constant or vanishes as $n \to \infty$. For the former portion, Corollary 1 (from Theorem 1) provides a tight bound to the bandwidth requested from the server. For the latter portion, we limit ourselves to apply the bound in Lemma 2 (repeated in Corollary 1). Although this bound may appear particularly coarse, it correctly captures (in order sense) the impact that cold videos have on the aggregate bandwidth requested from the servers. This is because, for any cold video, peer assistance is rather ineffective (in a passive system). Indeed, since the average number of available seeds for any cold video keeps bounded also when n grows large, it is possible (with a non-vanishing probability) that at a given time either there are no seeds supporting its distribution or all the seeds are sleeping. In this case (occurring with nonvanishing probability), a minimum amount of bandwidth equal to d is needed from the servers to

sustain the distribution of a video. At last, observe that d constitutes a finite fraction of the whole bandwidth needed by peers since the number of concurrent downloaders is finite. We obtain the following theorem.

Theorem 2: Under the same assumptions on U of Theorem 1, if there exists an arbitrarily small constant $\sigma > 0$ such that $\mathbb{P}(\Gamma_k < 1 - \sigma) = 1$, then the average bandwidth $\overline{S}$ requested from the servers satisfies the following asymptotic bound w.h.p. as the number of users n tends to ∞:

$$\overline{S} = \begin{cases} O(n^{1/\alpha}), & \text{if } \alpha > 1, 1/\alpha < \beta \leq 1 \\ o(1), & \text{if } (\alpha \leq 1, \beta < 1) \vee (\alpha > 1, \beta < 1/\alpha) \\ O(n), & \text{if } \alpha \leq 1, \beta = 1. \end{cases}$$

If, instead, $\mathbb{P}(\Gamma_k \geq 1) > 0$, then $\overline{S} = \Theta(n)$ w.h.p., $\forall \beta > 0$.

A detailed proof is reported in Appendix C.

Remarks: We emphasize that when all videos are in the surplus mode, the dominant contribution to the bandwidth requested from the servers is always due to the coldest portion of the video catalog, as it clearly emerges from the proof of Theorem 2. In particular, when $\alpha > 1$, $1/\alpha < \beta < 1$, the scaling law of $\overline{S}$ is determined by videos whose request rate either remains constant or decays to zero.

Although a passive system is conceptually simple to implement and manage, it is a very rigid (and potentially suboptimal) scheme since users are constrained to devote their entire upload bandwidth to the last requested video, and by so doing their resources might not be fully utilized. In Section V, guided by the insights gained from the analysis of passive systems, we will investigate the performance achievable by active systems.

V. ACTIVE SYSTEMS

In active systems, users can be instructed to essentially download/upload data belonging to arbitrary videos, with the following obvious constraints: 1) they must at least download (at constant rate) the videos that they want to watch; 2) they can upload only data previously downloaded. In particular, users can download/upload chunks or stripes belonging to videos they have not requested (data bundling). However, we will not consider the extreme case in which chunks/stripes can be made arbitrarily small (fluid limit), i.e., chunks/stripes whose size asymptotically goes to zero, because this is not implementable in practice.

We will show that, even with this restriction (i.e., the size of chunks/stripes cannot go to zero), we can devise efficient active strategies that can overcome the fundamental limitations of passive systems. In particular, we need to solve two orthogonal problems: 1) the possible presence of videos in deficit mode, which prevent any passive system to scale (we call this the *load balancing* problem); 2) the possible presence of cold videos, which are especially detrimental to the system (we call this the *catalog warming* problem). For each problem, we will present more than one solution, reporting the main results and the basic intuition about how they work. We anticipate that, once we solve both problems above, the computation of the resulting system performance will be an easy task.

Before describing the proposed techniques, we want to emphasize that, also for the active systems, the dynamics of users can be described by an open network of infinite servers queues, which turns out to be a BCMP queue as specified by the following lemma.

Lemma 3: The dynamics of users in the system are represented by a BCMP (i.e., product-form) open queuing network in which a different queue corresponds to a possible user configuration.

Also in this case, system evolution can be represented by a network of infinite servers queues associated to different user states. However, the resulting queuing network is more complex than in the passive case since the number of possible states in which a user can be and to which a queue corresponds is much larger than in the passive case. For example, a user can download a content while hosting a different content to which it is assigned as seed.

A. Load Balancing by Seed Reallocation

The goal of video equalization is to make the loads induced by individual videos equal to the global system load (4). The simplest approach to redistribute the peer upload resources to achieve this goal is to remove the constraint that peers must act as seed only for the last downloaded video. Here, we note that videos for which the video rate is small ($d_k < \overline{U}$ ideally) do not need seeds, while video contents with large video rate ($d_k > \overline{U}$) and long duration are much more critical. A good performing strategy is to concentrate most of the seeds among the videos that are more critical. This goal can be achieved in a fairly natural way by allowing peers to offer service (as seed) for an arbitrary video content that has been previously downloaded. To implement such a scheme, peers are only requested to keep in memory the content for which they act as seed. Essentially, we can allocate extra seeds to those videos having $\gamma_k > \gamma$ in the passive system. Although the approach is simple (it does not require any chunk/stripe bundling), the performance of this strategy is clearly limited by the fact that users download only a finite number of videos m before leaving the system, hence they cannot act as seed for arbitrary videos.

At the same time, video k for which $d_k/\overline{U} < \gamma$ may be cause of potential bandwidth wastage for the system since it may subtract excessive amount of bandwidth resources to other videos. Observe, indeed, that the kth video load remains too small (smaller than γ) even when we subtract to it all the seeds. One remedial action, in this case, is to artificially increase the nominal download rate of content k, decreasing simultaneously the download time. By doing so, as a side effect we increase the average time during which peers may act as seed after downloading video k. Thus, by increasing the download speed of low loaded contents, we can spare more bandwidth resources for other videos.

In accordance with previous considerations, we propose a seed reallocation strategy that works as follows: 1) All videos are downloaded at the same speed $d_{\max}$; this way we decrease the download time (that becomes $\tau_k' = l_k/d_{\max}$) and increase the average time during which peers may act as seed after downloading video k (we denote this quantity by $\overline{T}_{\mathrm{cont},k} = \overline{T}_{\mathrm{seed}} + \tau_k - \tau_k'$). 2) Peers acting as seeds are divided into $K+1$ categories: seeds assigned to a specific video and unassigned seeds. Seeds assigned to video k act as seed for video k for all their residual lifetime in the system; unassigned seeds, instead, act as seeds for the last downloaded video. Every fresh new peer joining the system is initially unassigned. An unassigned peer, after downloading video k, is assigned to video k with probability q_k, while it remains unassigned with probability $1 - q_k$.

Theorem 3: Given $\{d_k\}_k$, $\{l_k\}_k$, the proposed seed reallocation strategy guarantees perfect load balancing (by properly selecting probabilities q_k), iff $\gamma < 1$, and the following condition on τ_k', $\overline{T}_{\mathrm{cont},k}$ is satisfied:

$$\max_k \left[\tau_k' \cdot \max_h \left[\frac{\overline{T}_{\mathrm{cont},h}}{\tau_h'} \right] - \overline{T}_{\mathrm{cont},k} \right] \le (m-1)\overline{T}_{\mathrm{cont}} \quad (8)$$

where $\overline{T}_{\mathrm{cont}} = \sum_k p_k \overline{T}_{\mathrm{cont},k}$. Deconditioning with respect to d_k and l_k, we obtain that a perfect balance of video loads is feasible w.h.p. iff

$$l_{\max} \left(\frac{\overline{T}_{\mathrm{seed}}}{l_{\min}} - \frac{\overline{T}_{\mathrm{seed}}}{l_{\max}} + \frac{1}{d_{\min}} - \frac{1}{d_{\max}} \right) \le (m-1) \mathop{\mathbb{E}}_{L_k, D_k} [\overline{T}_{\mathrm{cont}}].$$

The proof is reported in Appendix D.

Note that if users stayed indefinitely in the system, they would sooner or later download any video that requires additional seeds, hence by properly setting probabilities q_k, we would surely be able to equalize the loads. Theorem 3 provides the sufficient and necessary condition on the average number of videos m downloaded by a user (which is proportional to average residence time in the system) such that perfect load balancing is still possible. The previous approach can be further boosted by allowing view-upload decoupling, i.e., by letting users assigned to video k to act as seed for it also while they are downloading a different video.

Theorem 4: Adding universal streaming boost to the previous described seed reallocation policy, we are able to perfectly equalize the load for different videos iff $\gamma < 1$ and τ_k', $\overline{T}_{\mathrm{cont},k}$ satisfy the following condition:

$$\max_k \left[\tau_k' \cdot \max_h \left[\frac{\overline{T}_{\mathrm{cont},h}}{\tau_h'} \right] - \overline{T}_{\mathrm{cont},k} \right] \le (m-1)(\overline{T}_{\mathrm{cont}} + \overline{\tau}') \quad (9)$$

where $\overline{\tau}' \triangleq \sum p_k l_k/d_{\max}$.

We omit the proof of Theorem 4 since it follows exactly the same lines of the proof of Theorem 3.

B. Load Balancing by Stripe Bundling

This technique is based on the following idea: Each video is divided into M stripes (substreams), which have to be downloaded in parallel by a user requesting the video (the distribution of each stripe can be assisted by a different set of peers) and reassembled by the decoder. Users who are downloading a video with $\gamma_k < \gamma$ are forced to download also one stripe of a video with $\gamma_k \ge \gamma$ and devote a fraction of their upload bandwidth (actually, all of their excess upload bandwidth with respect to the target average system load) to the additional bundled stripe.

The upcoming Theorem 5 guarantees that, by making M large enough (but not infinite), we can approximately equalize all videos, bringing all of them in surplus mode.

Before going into technical details, we provide an intuitive understanding of why this strategy turns out to be very effective to balance the video loads while minimally increasing the global system load. Indeed, while on the one hand some users (those requesting a video with $\gamma_k < \gamma$) have to download additional unwanted data (but this additional amount of data, corresponding to a single stripe, can be made smaller and smaller by increasing M), on the other hand these users can exploit all of their excess upload bandwidth to assist the distribution of the bundled stripe, typically retransmitting many copies of it to other peers before leaving the system, with an obvious gain in terms of system performance. This technique is more complex

to implement than the previous one based only on seed realloca-tion. However, it has the great advantage that it does not require any additional condition on the system parameters. In particular, it works also in the extreme case in which users leave the system after downloading just one video ($m = 1$).

We now describe in more details the proposed load balancing technique, and then provide the main result characterizing its performance.

Definition 1: For each video k, we define the amount of band-width ΔU_k

$$\Delta U_k = \overline{U} \frac{\gamma - \gamma_k}{\gamma}. \tag{10}$$

It can be easily verified that ΔU_k is the amount of bandwidth that, if subtracted to the average upload bandwidth of a peer, would make the load of video k equal to γ, i.e.,

$$\frac{d_k \tau_k}{(\overline{U} - \Delta U_k) \overline{T}_k} = \gamma$$

where $\overline{T}_k = \tau_k + \overline{T}_{\text{seed}}$. We observe that ΔU_k is positive for videos with $\gamma_k < \gamma$, and negative for videos with $\gamma_k \geq \gamma$. Moreover, the following equation holds:

$$\sum_{k=1}^{K} \lambda_k \overline{T}_k \Delta U_k = 0 \tag{11}$$

which suggests that, for a proper equalization, we have to con-sider the weighted contribution of each video to the overall system load, which requires to multiply ΔU_k both by the av-erage stay in the system of users downloading video k and by the request rate λ_k. Before thinking about a scheme to distribute the aggregate excess bandwidth of videos with $\gamma_k < \gamma$ to videos with $\gamma_k \geq \gamma$, we need to address the fundamental problem that a peer can help another peer only if it already stores (or it is concurrently downloading) data of interest to the other peer. In the extreme case, peers leave the system after downloading just one video, hence they cannot assist any other video distribution unless we force them to download at least one piece of another video (bundling), which can be done as soon as they enter the system.

To minimize the amount of bundled data, we divide each video k into M stripes, which are downloaded at constant rate d_k/M. We can treat a stripe exactly in the same way as a full video and view the system catalog as being composed by KM independent contents corresponding to the set of all stripes. Users independently retrieve the M stripes of each requested video in parallel and devote an equal fraction of their upload bandwidth U to assist the distribution of each stripe. By doing so, the load associated to each stripe equals the load of the entire video.

If a video k has some excess bandwidth (i.e., $\gamma_k < \gamma$), a peer downloading it acts as follows: As soon as it requests the video, it starts concurrently to download also a stripe of a video having $\gamma_k \geq \gamma$ (to be specified later). Then, it devotes a fraction $\frac{\overline{U} - \Delta U_k}{\overline{U}} \frac{U}{M}$ of its upload bandwidth to each stripe of the origi-nally requested video. The remaining upload bandwidth $\frac{U \cdot \Delta U_k}{\overline{U}}$ is devoted to assist the distribution of the bundled stripe. Notice that, on average, a peer requesting a video with $\gamma_k < \gamma$ reduces the upload bandwidth devoted to this video by ΔU_k. It follows that the new load associated to a video with $\gamma_k < \gamma$ (and to any of its stripes) exactly equals the target load γ.

We still need to specify which stripe is bundled to peers down-loading a video with $\gamma_k < \gamma$. This is done by the following water-filling-like algorithm. Initially, we put in a set $\mathcal{S}$ of all in-dexes of videos with $\gamma_k < \gamma$. Moreover, $\forall j \in \mathcal{S}$, we initialize to $\lambda_j^* = \lambda_j$ the arrival rate of users requesting file j who have not yet been assigned a bundled stripe.

We then consider (in arbitrary sequence) all of the stripes be-longing to videos with $\gamma_k \geq \gamma$, one at a time. For each of them (say stripe i), let k be the index of the video to which it belongs. We initialize to $Q_k = |\lambda_k \overline{T}_k \Delta U_k / M|$ a quantity representing the residual impact that this stripe has on the system unbalance. As long as $Q_k > 0$, we perform this step: We randomly ex-tract an index j from set $\mathcal{S}$ and consider the (positive) quantity $Q_j = \lambda_j^* \overline{T}_j \Delta U_j$. Two cases are possible: If $Q_j \leq Q_k$, we need to use the bandwidth of all remaining users requesting file j. This means that peers requesting file j have to download stripe i with probability λ_j^* / λ_j (notice that this quantity equals 1, if λ_j^* is still maintaining its initial value). Furthermore, index j is re-moved from set $\mathcal{S}$, and Q_k is diminished to $Q_k - Q_j$. We extract a new index j from $\mathcal{S}$ and repeat the same step; if $Q_j > Q_k$, it means that a subset of peers downloading video j is enough to fill the remaining bandwidth deficit of stripe i: In particular, peers requesting file j are forced to download stripe i with prob-ability $p_{j,i} = \frac{Q_k}{\lambda_j \overline{T}_j \Delta U_j}$. At last, before considering the next stripe with $\gamma_k \geq \gamma$, we update λ_j^* to $\lambda_j^* - p_{j,i} \lambda_j$.

Equation (11) guarantees that the above algorithm terminates after handling all stripes with $\gamma_k \geq \gamma$, finding for each of them enough peers downloading videos with $\gamma_k < \gamma$ to fill its deficit of bandwidth. At the same time, all peers downloading a video with $\gamma_k < \gamma$ are assigned exactly one bundled stripe to down-load. Notice that if the bundling mechanism were for free (i.e., no need for any peer to download additional data), after the redistribution of upload bandwidth performed by the previous algorithm, we would have all stripes with equal load γ. How-ever, we need to consider the cost of bundling. The following theorem guarantees that such additional cost can be made arbi-trarily small by increasing M, by employing a slightly modified version of the previously described bundling scheme.

Theorem 5: For any value γ' such that $\gamma < \gamma' < 1$, there exists a value $M^* < \infty$ for the number of stripes such that for all $M > M^*$ a stripe bundling scheme can be found that brings the system to operate at global load smaller than γ'. At the same time, the load associated to each video becomes smaller than or equal to γ' (the same holds considering the load induced by individual stripes).

The proof is reported in Appendix E.

Remark: Comparing the two previously proposed techniques to achieve load balancing, we can say that seed relocation is much simpler to implement. Indeed, it does not require any major modification to the system architecture and protocols. The only actions requested to peers for the implementation of such technique are: 1) to download videos at maximum rate $d_{\max}$; 2) to become seed of a video in a predetermined/coordinated fashion. The implementation of the stripe bundling mechanism is, instead, more complex, as it requires the application of both video striping and video bundling techniques. On the other end, stripe bundling turns out to be significantly more effective in highly dynamic scenarios in which peers download only few videos (in the limit, just one video) before disappearing from the system.

C. Catalog Warming by Video Bundling

While analyzing the case of a passive system, we learned that cold videos (videos whose request rate does not increase with n) are responsible for the dominant component of the bandwidth requested from the servers. Hence, if we could artificially increase the request rate of cold videos, we would expect to get a significant reduction of $\overline{S}$. Now, it turns out that we do not need to warm up the coldest portion of the catalog too much: Optimal performance is already achieved when the request rate of videos go to infinite at least as fast as a poly-log function, i.e., when $\lambda_k = \Omega((\log n)^z)$, $\forall k$ (actually, this is needed only for a "critical" portion of the catalog) where z is a suitable constant. Provided that $\beta < 1$, the amount of data bundling necessary to achieve this goal is rather small, hence it is possible to warm the catalog up enough, while at the same time increasing the global system load to a value γ' such that

$$\gamma' - \gamma = \Delta\gamma \to 0, \qquad \text{for } n \to \infty \qquad (12)$$

as it will be shown in the proof of Theorem 6.

The simplest approach to achieve this goal is to make some peers in the *contributing* state download entire (unrequested) videos while they are not downloading any other content. This mechanism does not require any video chunking/striping and can be superposed to the load balancing strategy described in Section V-A. In essence, the strategy works as follows. Let λ'_k be the target new rate at which video k should be downloaded in the active system. Peers who have just finished downloading a hot video $h \le K_0$ (for a specific constant K_0), without having been assigned to it by the seed reallocation strategy, are induced to start the download of a cold video $k > K_0$ with probability $p'_k = \frac{(\lambda'_k - \lambda_k)}{\sum_{h \le K_0} \lambda^u p_h (1-q_h)}$ (note that, by construction, $p_d = \sum_{k>K_0} p'_k \to 0$ as $n \to \infty$, whenever $\beta < 1$.[6] If the download of the bundled video is interrupted because the peer goes into the sleeping state, the download is promptly resumed as soon as the peer restarts contributing to the system without concurrently downloading any other video (they cannot be, as it will be shown in the proof of Theorem 6, assigned to any video they possibly download in the meanwhile). Since $p_d \to 0$, the negative effect that this strategy has on the load induced by hot videos (those videos whose request rate is not increased, and from which the mechanism subtracts some seeds) becomes negligible for $n \to \infty$. Our strategy has a potential effect also on the load of videos whose request rate is artificially increased. However, it guarantees that the new load γ'_k of such videos is maintained less than 1, provided that the average upload bandwidth of adjoint seeds exceeds the average bandwidth consumed to download them, i.e., $d_k \min(m\overline{T}_{\text{cont}}, l_k/d_k) < \overline{U} m \overline{T}_{\text{cont}}$. When this condition is met with probability 1, i.e.,

$$\mathbb{P}\left(\frac{d_{\max} \min(m\overline{T}_{\text{cont}}, l_{\max}/d_{\max})}{m\overline{U}\overline{T}_{\text{cont}}} > 1\right) = 1 \qquad (13)$$

the above scheme can be effectively employed, in sufficiently large systems, without bringing any video in deficit mode.

D. Catalog Warming by Chunk Bundling

The previous technique imposes again a constraint on the system parameters (13). When (13) is violated, the same approach can be applied to individual pieces of cold videos

[6]When $\beta = 1$, only a portion of the coldest video collection can be warmed up while guaranteeing $p_d \to 0$, as shown in the proof of Theorem 6.

(chunks), instead of entire videos, with less stringent constraint. Indeed, peers who are forced to contribute to an unrequested video neither need to completely retrieve it nor to download it sequentially. Thus, we can cut a cold video in M chunks and ask some peers to download just a randomly chosen chunk contributing to its distribution. Chunkization reduces the bandwidth that every artificial downloader consumes by a factor M, while keeping constant its potential contribution on the upload.

E. Catalog Warming by Stripe Bundling

A similar idea can be applied to stripes, instead of chunks, and superposed to the load balancing technique proposed in Section V-B. Essentially, peers who request for the first time a hot content $h \le K_0$ (for some constant K_0), with probability $p'_k = \frac{\lambda'_k - \lambda_k}{\sum_{h \le K_0} \lambda_h}$, are forced to download also a randomly chosen stripe of cold video k, of size l_k/M (where M is a suitable constant), contributing to its distribution for the rest of their stay into the system, with an opportunely chosen fraction f_k of their upload bandwidth.

To properly choose fraction f_k and constant M, we need to evaluate the impact that the proposed bundling scheme has on the load induced by individual videos (or stripes), guaranteeing that such load remains smaller than 1.

To do this, let us assume that, after applying (if necessary) the load balancing technique proposed in Section V-B, all videos (or stripes) induce a load smaller than $\gamma' < 1$. In particular, let $\gamma_k = a/b$ (with $a < \gamma' b$) be the load induced by a generic cold video k. After applying the catalog warming technique by stripe bundling, the new load induced by each stripe of video k can be upper-bounded by

$$\frac{a + (\lambda'_k - \lambda_k)l_k/M}{b + (\lambda'_k - \lambda_k)f_k\overline{U}(\tau_{\min} + \overline{T}_{\text{seed}})} \qquad (14)$$

under the pessimistic assumption that all bundled downloads of hot videos last for the minimum duration $\tau_{\min} + \overline{T}_{\text{seed}}$. Moreover, considering the worst case in which all cold videos have size $l_k = l_{\max}$, we can guarantee that the r.h.s. of (14) is smaller than γ' by selecting a fraction of upload bandwidth

$$f_k > \frac{l_{\max}}{M\gamma'\overline{U}(\tau_{\min} + \overline{T}_{\text{seed}})} = f_{\min}$$

where $f_{\min}$, which no longer depends on k, can be made arbitrarily small by increasing M. Since $f_{\min}$ must be smaller than 1, we also need to select

$$M > \frac{l_{\max}}{\gamma'\overline{U}(\tau_{\min} + \overline{T}_{\text{seed}})}.$$

At last, observe that the load increase for hot videos (due to the subtraction of some upload bandwidth) vanishes as the system size increases since $p'_d = \sum_{k>K_0} p'_k \to 0$ as $n \to \infty$. As a consequence, the proposed scheme can always be applied in sufficiently large systems.

F. Discussion on Implementation Issues and Overheads

We emphasize that previously proposed techniques to achieve load balancing and catalog warming incur very different costs in terms of system complexity and overhead. Load balancing by seed relocation has a marginal cost on the system since peers need just to be instructed about which content to become seed of (except for this, they behave exactly as in a passive system). Such information can be computed in a centralized fashion by one or more servers that oversee the overall

downloading process and periodically send control messages to peers (these servers essentially mimic the role of trackers in BitTorrent).

Load balancing by stripe bundling is significant more complex since it requires the system to support video striping. This mechanism also has an unavoidable overhead since peers are forced to download stripes of unwanted videos, as discussed in Section V-B.

Similarly, catalog warming requires peers to download possible unwanted data (peers must be instructed to do so by a central authority, as before). Such intrinsic overhead can be mitigated/made more efficient by employing chunking/striping, as discussed in Section V-C, which however adds complexity to the control plane. Observe that chunk and stripe bundling are conceptually similar and require peers to execute essentially the same high-level operations. On the one hand, video chunking/unchunking are very simple operations typically supported by every peer-assisted system (and most CDNs), while video striping/reconstruction are more complex. On the other hand, video striping, once supported, can be advantageously exploited also for load balancing.

At last, we remark that in our analysis we have ignored the effects of protocol overheads and signaling bandwidth (necessary to coordinate the peers). The impact of protocol overheads and signaling, once quantified for a given application, can be easily incorporated in the model by redefining the peer upload bandwidth (i.e., subtracting from it the fraction devoted to signaling/overhead). Our paper also ignores: 1) possible constraints on the number of peers from which a peer can download data; 2) the effects of possible congestions inside the network. Although these issues can have an impact on the system performance, we have preferred not to consider them for the sake of simplicity and analytical tractability.

G. Asymptotic Bandwidth Requested From the Servers

At last, we can evaluate the asymptotic performance achievable by applying the active schemes described in previous sections.

Theorem 6: Under the same assumptions on U of Theorem 1, if there exists an arbitrarily small constant $\sigma > 0$ such that $\mathbb{P}(\Gamma < 1 - \sigma) \to 1$ for some $\sigma > 0$, the bandwidth $\overline{S}$ requested from the servers satisfies the following asymptotic bound w.h.p. as the number of users n tends to ∞:

$$\overline{S} = \begin{cases} o(1), & \text{if } \alpha > 1, \beta < 1 \\ O(n^{2-\alpha}(\log n)^{\frac{\alpha-1}{1-\delta}}), & \text{if } \alpha > 1, \beta = 1 \end{cases} \quad (15)$$

provided that a suitable combination of active techniques is employed: 1) to balance the loads induce by individual videos; 2) to sufficiently increase the download rate of cold videos. The proof of Theorem 6 is reported in Appendix F.

VI. LOWER BOUND

Here, we present a simple universal lower bound to the bandwidth requested from the servers. Notice, however, that this bound holds under the assumption that the size of a chunk/stripe cannot go to zero. Consider first the case in which videos are not divided into chunks/stripes. For any video k, the servers must provide at least a bandwidth equal to d_k when the following

two conditions jointly occur: 1) there is at least one user downloading the video; 2) there are no seeds assisting its distribution. Thus, we can write

$$\overline{S} \geq \sum_{k=1}^{K} d_k \mathbb{P}(N_{d,k} > 0)\mathbb{P}(N_{\text{seed},k} = 0). \quad (16)$$

Previous argument can be extended to the case in which videos are divided into a finite number of chunks/stripes, considering every chunk/stripe as an individual object. By algebraically manipulating (16), we obtain the following theorem.

Theorem 7: The average bandwidth requested from the servers, $\overline{S}$, satisfies the following asymptotic bound as the number of active users n tends to ∞:

$$\overline{S} = \begin{cases} \Omega(n^{2-\alpha}), & \text{if } \alpha > 1, \beta = 1 \\ \Theta(n), & \text{if } \alpha \leq 1, \beta = 1 \\ 0, & \text{if } \beta < 1. \end{cases} \quad (17)$$

The proof is reported in Appendix G. Essentially, the proof consists in finding a lower bound for (16) that uniformly holds under any possible distribution of seeds to videos (i.e., satisfying $\sum_k \overline{N}_{\text{seed},k} = O(N)$).

VII. RELATED WORK

A stochastic fluid model showing fundamental characteristics and limitations of P2P streaming systems was proposed in [15]. In [16], performance bounds on the minimum server load, maximum streaming rate, and minimum tree depth under different peer selection constraints are derived. However, the two papers above focus only on single-channel streaming system. Universal streaming architectures have been analytically studied in [12], where authors develop queueing network models to describe multichannel live streaming systems incorporating peer churn, bandwidth heterogeneity, and Zipf-like popularity. We remark that VoD systems are different from live streaming systems in which users join the distribution of a given TV channel at random points in time, but peers connected to the same channel watch the content almost synchronously. In VoD, a given video is watched asynchronously by users, and downloading peers can only help peers who have started the download later on in time (sequential delivery). Moreover, asymptotic results in [12] are restricted to the case of two values of peer upload bandwidth (low and high) and require finding the solution (if any) to a set of linear equations. In contrast to [12], we consider VoD systems and obtain a simpler characterization of the asymptotic system performance for general upload bandwidth distribution. In [17], authors propose an algorithm to allocate server bandwidth that can predict the minimum server bandwidth requested for each channel, based on historical information. However, this work focuses only on live streaming systems.

The first mathematical formulation of the server bandwidth needed by a VoD system based on sequential delivery appeared in [4], in which authors resort to a Monte Carlo approach to get basic insights into the system behavior (like surplus and deficit modes). The same formulation has been considered in [14], where authors explore by simulation the effectiveness of different replication strategies to minimize the server load in the slightly surplus mode, as well as distributed replacement algorithms to achieve it. In [18], upper and lower bounds to the server load are derived. However, detailed information about the movie set stored by each peer and its upload capacity must

be collected by the server. In [19], authors propose a queuing model to predict the dynamic demand of the users in a P2P VoD system and provide on-the-fly elastic amounts of processing/bandwidth resources with a minimum cost.

Reference [20] presents a trace-driven evaluation of server load savings for VoD streaming. Specifically, it shows the potential savings by using hybrid CDN-P2P systems for two major CDNs: Akamai and Limelight. [21] defines a per-chunk capacity model focusing on the allocation of upload bandwidth resources among different chunks. In particular, it points out the fundamental tradeoffs that exist among system throughput, sequentiality of downloaded data, and robustness to heterogeneous network conditions. In [22], authors develop an analysis of peer-assisted VoD systems with scalable video streams. Analytical models that estimate the number of peers that can be admitted into the system in the case of flash crowds are provided. Similarly to [23], which focuses on P2P live streaming only, [24] studies the achievable streaming capacity of large-scale P2P VoD systems with sparse connectivity among peers and investigates P2P control strategies that can achieve close-to-optimal streaming capacity. With respect to all previous work, in this paper we propose a general framework that allows us to analytically estimate the potential benefits in terms of scalability of peer-assisted VoD architectures, revealing under which conditions gains (in terms of servers' bandwidth savings) can be significant or limited.

An interesting implementation of the kind of systems considered in our work is Xunlei [25], a download acceleration application that is becoming enormously popular in China. Xunlei combines both peer-assisted and server-assisted techniques, letting users download portions of the requested contents from other peers while also downloading portions from independent servers. Recently, the Xunlei network started also a peer-assisted VoD service (Kankan), which generated massive-scale swarms.

VIII. CONCLUSION

Our results indicate that users' cooperation can dramatically reduce the servers' burden in large-scale VoD systems. Although peer-assisted architectures incur several issues related to the added complexity on the control plane, the need to provide incentive mechanism to the users, and the need to protect the system against attacks and misbehavior, nevertheless we believe they should be taken seriously into consideration in the coming years, as they are the only known solution (up to now) to make VoD systems arbitrarily scalable. However, we have shown that the potential gains derived by users' cooperation are reduced when the service is targeted to the distribution of user-generated contents (especially for small values of the Zipf's law exponent) since in this case the number of videos intrinsically scales linearly with the number of users.

APPENDIX A
PROOF OF THEOREM 1

For the proof of this theorem, we need to state a preliminary result.

Lemma 4: Assume the following properties hold for U: i) $\overline{U} > 0$; ii) $\mathbb{E}[e^{\theta U}]$ is finite in a neighborhood of the origin; iii) $F_U(w) > 0$ for every $w > 0$. The average bandwidth requested from the servers, $\overline{S}$, satisfies the following bound: If $\gamma < 1$, for any $\delta \in (0, 1)$, there exists $\lambda_0 \in \mathbb{R}^+$ such that

$$\overline{S} \le \begin{cases} 4\lambda^{\delta}e^{-C_1\lambda^{1-\delta}}, & \text{if } d > \overline{U}, \lambda > \lambda_0 \\ C_3 e^{-C_2\lambda}, & \text{if } d \le \overline{U} \end{cases} \quad (18)$$

with $C_1 = \tau\frac{\overline{U}(1-\gamma)(1+\gamma)}{4\gamma}, C_2 = \left(\frac{d}{\overline{U}\gamma} - 1\right)\tau(1-e^{-\theta^*d}), C_3 = (d + 1/\theta^*)e^{\theta^*d}$, where θ^* is the only strictly positive solution of the equation $\mathbb{E}[e^{\theta(d-U)}] = 1$. Furthermore in (18), λ_0 can be found uniformly with respect to $\overline{U}$, d, and γ as long as $\gamma < 1 - \sigma$ for any $\sigma > 0$. If, on the other hand, $\gamma > 1$, the average bandwidth requested from the servers grows linearly with the number of users, namely, $\overline{S} = \Theta(\lambda)$.

The proof of Lemma 4 can be found in Appendix B, and it is based on some results that we proved in [11].

Considering the constants specified in Lemma 4, we observe that C_1 is insensitive to the distribution of U, and it does not depend explicitly from $\overline{U}$. Actually, its dependency from $\overline{U}$ is mediated by γ. As a consequence, the expression of the bound for $d > \overline{U}$ is robust to the distribution of U and its mean, provided that $\gamma < 1 - \sigma$. Instead, C_2 and C_3 are sensitive to the distribution of U, through the quantity θ^*. For this reason, the upper bound on $\overline{S}$ for $d \le \overline{U}$ is more delicate. In particular, note that if θ^* becomes arbitrarily small or large, the bound $C_3 e^{-C_2\lambda}$ becomes arbitrarily weak. For these reasons, we need to strengthen the bound for $d \le \overline{U}$, which is exactly the result in Theorem 1.

We state now the last preliminary result that we need for proving Theorem 1. The following proposition states that, given all the other system parameters, the average bandwidth requested to server $\overline{S}$ is, by construction, monotonically non-increasing with respect to the available upload bandwidth of peers. More formally, we can state Proposition 1.

Proposition 1: Whenever we compare two systems, 1 and 2, with identical system parameters $(d, \tau, T_{\text{seed}})$ except for the available peer upload bandwidth, then

$$\overline{S}^{(1)} \le \overline{S}^{(2)}$$

whenever the available bandwidth in the first system $U^{(1)}$ is stochastically greater than the available bandwidth in the second system $U^{(2)}$.

This property directly descends from the observation that peers in the first system can always contribute to the redistribution with only a fraction of their available bandwidth, properly choosing the fraction so that the contributed bandwidth to the P2P system is distributed as $U^{(2)}$.

Finally, we can prove Theorem 1.

Exploiting the definition of θ^*, we can derive the following lower and upper bounds for θ^*: $\theta^* > \frac{2E[U-d]}{\mathbb{E}[(U-d)^2]}$ and $\theta^* < \sup_{x<d} -\frac{\log\mathbb{P}(U<d-x)}{x}$, which guarantees that θ^* cannot be arbitrarily small or large when d is sufficiently smaller than $\overline{U}$, let us say when $d(1 + \sigma/2) \le \overline{U}$. Therefore, whenever $\overline{U} > d(1 + \sigma/2)$, in Lemma 4 we can jointly lower-bound C_2 by a positive constant, and upper-bound C_3 by a constant. It follows that $C_3 e^{-C_2\lambda} = o(\lambda^{\delta}e^{-C_1\lambda^{1-\delta}})$, as $\lambda \to \infty$, $\forall \delta \in (0, 1)$. When $\overline{U} \to d$ from the right, $\theta^* \to 0$ and the bound in Lemma 4 becomes arbitrarily weak (it tends to infinite). To overcome this problem, we exploit the monotonicity property of $\overline{S}$ with respect to the peers upload bandwidth (Proposition 1) obtaining a useful bound also for $d \le \overline{U} < d(1 + \sigma/2)$. We consider a system in which the available upload bandwidth of all peers has been reduced by a factor $\frac{2}{2+\sigma}$. In this new system, the average

effective upload bandwidth is smaller than d, and the video load is equal to $\gamma' = \gamma(1 + \sigma/2) < 1$. For this system, we can bound $\overline{S}$ by the expression valid for $d > \overline{U}$, obtaining a bound that clearly applies in the original case thanks to Proposition 1.

<div align="center">APPENDIX B
PROOF OF LEMMA 4</div>

We recall the following two propositions from [11]. They establish important properties of θ^* as function[7] of ϵ.

Proposition 2: If $d > \overline{U}$, the equation $\mathbb{E}[e^{\theta(\overline{U} - U - \epsilon)}] = 1$ (in θ) admits a unique solution for $\epsilon \in (0, \overline{U})$. Furthermore, $\theta^*(\epsilon) = \arg_{\theta > 0}(e^{-\theta\epsilon}\mathbb{E}[e^{\theta(\overline{U} - U)}] = 1)$ is strictly increasing and C^1 on the interval $(0, \overline{U})$. Moreover, it holds $\lim_{\epsilon \to 0} \theta^*(\epsilon) = 0$.

Proposition 3: Provided that $d > \overline{U}$, and U is not constant, the image of $\theta^*(\epsilon)$ for $0 < \epsilon < \overline{U}$ is $\mathbb{R}^+ \setminus \{0\}$.

We recall also the following upper bound for the bandwidth requested from the servers from [11]:

$$\overline{S} \leq \left(d + \frac{1}{\theta^*} e^{-\theta^* A} e^{\overline{N}_d(e^{\theta^* A} - 1)}\right) e^{-\overline{N}_{\text{seed}}(1 - \phi_U(-\theta^*))} e^{\theta^* d} \tag{19}$$

where ϕ_U is the moment generating function of U, ϵ is an arbitrary positive constant smaller than $\overline{U}$, and $A \triangleq d - \overline{U} + \epsilon$. Moreover, θ^* is the unique strictly positive solution to the equation $\mathbb{E}[e^{\theta(d - U - A)}] = \mathbb{E}[e^{\theta(\overline{U} - U - \epsilon)}] = 1$.

Using Propositions 2 and 3, and the bound in (19) we can prove the lemma. We emphasize that Lemma 4 is not a straightforward extension of the results in [11], where we have only shown that $\overline{S} \to 0$ as $\lambda \to \infty$ (provided that $\gamma < 1$). Indeed, Lemma 4 characterizes also *how fast* $\overline{S}$ scales to 0 as λ grows large, providing a basic building block of our analysis.

Our goal is to tightly characterize the asymptotic behavior of bound (19) as $\lambda \to \infty$. We first focus on the case $d > \overline{U}$. In this case, we will make $\epsilon \to 0$ as $\lambda \to \infty$, exploiting Propositions 2 and 3.

Consider, first, quantity $e^{-\theta^* A} e^{\overline{N}_d(e^{\theta^* A} - 1)}$ in (19). Note that $e^{-\theta^* A} \leq 1$, and $e^{\theta^* A} - 1 > \theta^* A$. Thus, for all $\eta \in (0, 1)$, there exists $\theta_1^*(\eta, A) > 0$ such that if $\theta^* \in (0, \theta_1^*)$, then $e^{\theta^* A} - 1 \leq \theta^* A/(1 - \eta)$. In particular, $\theta_1^*(\eta, A) = -\frac{1 - \eta + \mathbb{W}((\eta - 1)e^{\eta - 1})}{A}$, where $\mathbb{W}(\cdot)$ is the Lambert function. Since $A = d - \overline{U} + \epsilon$, we define $\theta_2^*(\eta) \triangleq \inf_{\epsilon \in (0, \overline{U})} \theta_1^* = -\frac{1 - \eta + \mathbb{W}((\eta - 1)e^{\eta - 1})}{d - \overline{U}}$. We obtain the following: If $\theta^* \in (0, \theta_2^*)$

$$e^{-\theta^* A} e^{\overline{N}_d(e^{\theta^* A} - 1)} \leq e^{\overline{N}_d \frac{\theta^* A}{1 - \eta}}. \tag{20}$$

Consider now quantity $e^{-\overline{N}_{\text{seed}}(1 - \phi_U(-\theta^*))}$ in (19). Since for $\epsilon \to 0$ we have $\theta^* \to 0$ (Proposition 2), $1 - \phi_U(-\theta^*) = 1 - \mathbb{E}[e^{-\theta^* U}] = \theta^* \overline{U} + R(\theta^*)$, where $R(\theta^*)$ is the Taylor remainder. If we express $R(\theta^*)$ in the Lagrange form, we immediately obtain the following: $|R(\theta^*)| \leq \mathbb{E}[U^2]\theta^{*2}/2$.

After some elementary algebra, it is possible to show that for every $\eta' > 0$, defining $\theta_3^*(\eta') \triangleq \frac{\eta'}{1 + \eta'} \frac{2\overline{U}}{\mathbb{E}[U^2]}$, we have that if $\theta^* \in (0, \theta_3^*)$, the following holds: $1 - \eta' \leq \frac{\theta^* \overline{U}}{1 - \phi_U(-\theta^*)} \leq 1 + \eta'$, and therefore

$$1 - \phi_U(-\theta^*) \geq \frac{\theta^* \overline{U}}{1 + \eta'}. \tag{21}$$

---
[7] In the following, whenever not necessary, we use θ^* instead of $\theta^*(\epsilon)$.

Consider now quantity $e^{\theta^* d}$ in (19): Defining $\theta_4^* \triangleq (\log 2)/d$, it is immediate to see that for $\theta^* \in (0, \theta_4^*)$, it holds that

$$e^{\theta^* d} \leq 2. \tag{22}$$

By (19)–(22), we can conclude that for all $\theta^* \in (0, \theta_5^* \triangleq \min\{\theta_2^*, \theta_3^*, \theta_4^*\})$, it holds that

$$\overline{S} \leq 2\left(d + \frac{e^{\overline{N}_d \frac{\theta^* A}{1 - \eta}}}{\theta^*}\right) e^{-\overline{N}_{\text{seed}} \frac{\theta^* \overline{U}}{1 + \eta'}}. \tag{23}$$

From (4) we can derive a relation between the number of downloaders, $\overline{N}_d = \lambda\tau$, and the number of seeds

$$\overline{N}_{\text{seed}} = \left(\frac{d}{\overline{U}\gamma} - 1\right)\overline{N}_d = \left(\frac{d}{\overline{U}\gamma} - 1\right)\lambda\tau. \tag{24}$$

Substituting (24) in (23), we get

$$\overline{S} \leq 2\left(d + \frac{e^{\lambda\tau \frac{\theta^* A}{1 - \eta}}}{\theta^*}\right) e^{-\left(\frac{d}{\overline{U}\gamma} - 1\right)\lambda\tau \frac{\theta^* \overline{U}}{1 + \eta'}}. \tag{25}$$

Recall from Proposition 2 that, for $d > \overline{U}$ and $\epsilon \to 0$, we have $\theta^* \to 0$. Moreover, by Propositions 2 and 3, as $\lambda \to \infty$, we can set $\theta^* = \lambda^{-\delta}$ (i.e., we can find the proper law for ϵ that leads to $\theta^* = \lambda^{-\delta}$) for all $\delta \in (0, 1)$.

Thus, there exists a $\lambda_1 > 0$ such that $\forall \lambda > \lambda_1$, we have that $\theta^* \in (0, \theta_5^* \triangleq \min\{\theta_2^*, \theta_3^*, \theta_4^*\})$. Then, if $\lambda > \lambda_1$, for all η, $\eta' > 0$ we have

$$\overline{S} \leq 2\left(d + \lambda^\delta e^{\lambda^{1-\delta}\tau \frac{A}{1-\eta}}\right) e^{-\left(\frac{d}{\overline{U}\gamma} - 1\right)\tau \frac{\overline{U}}{1+\eta'}\lambda^{1-\delta}}$$
$$= 2de^{-\lambda^{1-\delta}\tau \frac{d - \overline{U}\gamma}{\gamma(1+\eta')}} + 2\lambda^\delta e^{-\lambda^{1-\delta}\tau \frac{(d - \overline{U}\gamma)(1-\eta) - A\gamma(1+\eta')}{(1+\eta')(1-\eta)\gamma}}. \tag{26}$$

Now we prove that the first term in (26) is negligible with respect to the second one. First, note that if $\lambda > \lambda_1$, $d < \lambda^\delta$: Indeed, by (22), we have that $d \leq \frac{\log 2}{\theta_5^*} = \log 2 \cdot \lambda_1^\delta < \lambda^\delta$.

Remembering that $A \triangleq d - \overline{U} + \epsilon \geq 0$, we can say that the exponent in the second term is larger than the exponent in the first one. Thus, we can say that, for $\lambda > \lambda_1$, we have

$$\overline{S} \leq 4\lambda^\delta e^{-\lambda^{1-\delta}\tau \frac{(d - \overline{U}\gamma)(1-\eta) - A\gamma(1+\eta')}{(1+\eta')(1-\eta)\gamma}}. \tag{27}$$

Remembering that $A \triangleq d - \overline{U} + \epsilon$, for all $\epsilon > 0$, $\eta, \eta' \in (0, 1)$ we define the quantity

$$f(\epsilon, \eta, \eta', d, \gamma, \overline{U}) \triangleq \tau \frac{(d - \overline{U}\gamma)(1 - \eta) - A\gamma(1 + \eta')}{(1 + \eta')(1 - \eta)\gamma} \tag{28}$$

that appears in (27).

We compute now $f(\epsilon, \eta, \eta', d, \gamma, \overline{U})$ for $\epsilon \to 0$ and $\eta = \eta' = \frac{1 - \gamma}{1 + \gamma}$

$$\lim_{\epsilon \to 0} f(\epsilon, d, \gamma, \overline{U}) = \tau \frac{(d - \overline{U}\gamma)\frac{2\gamma}{1+\gamma} - (d - \overline{U})\gamma\frac{2}{1+\gamma}}{(4\gamma^2)/(1 + \gamma)^2}$$
$$= \tau \frac{\overline{U}(1 - \gamma)(1 + \gamma)}{2\gamma} > 0.$$

Since $f(\epsilon, d, \gamma, \overline{U})$ for $\gamma \neq 0$ is a continuous function and hence a uniformly continuous one over compact sets that do not contain points with $\gamma = 0$, a strictly positive constant $\tilde{\epsilon} > 0$ can be found such that, for all $0 < \epsilon < \tilde{\epsilon}$, we have

$$f(\epsilon, d, \gamma, \overline{U}) > \tau \frac{\overline{U}(1 - \gamma)(1 + \gamma)}{4\gamma} > 0$$

uniformly with respect to parameters d, γ, and $\overline{U}$ as long as they take values over a compact set that does not contain the points with $\gamma = 0$ (as in our case). Let λ_2 be the value of λ that corresponds to $\tilde{\epsilon}$. Defining the constant

$$C_1 \triangleq \tau \frac{\overline{U}(1-\gamma)(1+\gamma)}{4\gamma} \tag{29}$$

we can conclude that, for $d > \overline{U}$, it holds that

$$\overline{S} \leq 4\lambda^{\delta} e^{-C_1 \lambda^{1-\delta}} \qquad \text{as } \lambda > \lambda_0 \triangleq \max\{\lambda_1, \lambda_2\}. \tag{30}$$

Consider now the case $d < \overline{U}$. From [11, Theorem 1], we have that θ^* is a constant as $\lambda \to \infty$, and we can set ϵ in such a way that $A = 0$. Thus, in this case, the bound in (19) becomes

$$\begin{aligned}
\overline{S} &\leq (d + 1/\theta^*) e^{-\overline{N}_{\text{seed}}(1-\phi_U(-\theta^*))} e^{\theta^* d} \\
&= (d + 1/\theta^*) e^{-\left(\frac{d}{\overline{U}\gamma} - 1\right)\lambda\tau(1-\phi_U(-\theta^*))} e^{\theta^* d} \\
&= C_3 e^{-\left(\frac{d}{\overline{U}\gamma} - 1\right)\lambda\tau(1-\phi_U(-\theta^*))} \tag{31}
\end{aligned}$$

where $C_3 \triangleq (d + 1/\theta^*) e^{\theta^* d}$. Since θ^* is a constant, we can conclude that, if $d < \overline{U}$, it holds that

$$\overline{S} \leq C_3 e^{-C_2 \lambda} \qquad \text{as } \lambda \to \infty \tag{32}$$

where $C_2 \triangleq \left(\frac{d}{\overline{U}\gamma} - 1\right)\tau(1 - \phi_U(-\theta^*))) = \left(\frac{d}{\overline{U}\gamma} - 1\right)\tau(1 - e^{-\theta^* d})$ since, by construction, $\phi_U(-\theta^*) = e^{-\theta^* d}$.

At last, we consider the case $\gamma > 1$ (i.e., the deficit mode). In this case, the bandwidth requested from servers scales as $\Theta(n)$, as shown in [11]. Indeed, the servers have to provide at least the bandwidth deficit. ∎

APPENDIX C
PROOF OF THEOREM 2

First, we assume $\mathbb{P}(\Gamma_k < 1 - \sigma) = 1$. In this case, all γ_k are deterministically smaller than $1 - \sigma$, and by Corollary 1 for each video, we get

$$\overline{S}_k \leq \begin{cases} d_k \tau_k \lambda_k, & \text{if } \lambda_k < \lambda_0 \\ 4\lambda_k^{\delta} e^{-C_{1,k}\lambda_k^{1-\delta}} \; \forall \delta \in (0,1), & \text{if } \lambda_k \geq \lambda_0 \end{cases} \tag{33}$$

where $C_{1,k} \triangleq C_1(\overline{U}_k, \tau_k, \gamma_k')$, $\gamma_k' = (1 + \sigma/2)\gamma_k$. We divide videos in two categories, depending on the request rate λ_k. Video k belongs to the first category if $0 \leq k \leq K_1$, with K_1 such that $\lambda_{K_1} = \lambda_0$, where λ_0 is the threshold defined in Corollary 1. We thus obtain that $K_1 = \lambda_0^{-1/\alpha} n^{1/\alpha} H(K)^{1/\alpha}$. We distinguish the following cases depending on α:

$$K_1 = \begin{cases} \Theta(n^{1/\alpha}), & \text{if } \alpha > 1 \\ \Theta(n/\log n), & \text{if } \alpha = 1 \\ \Theta(n^{\frac{1+\beta(\alpha-1)}{\alpha}}), & \text{if } \alpha < 1. \end{cases} \tag{34}$$

Videos k such that $K_1 \leq k \leq K$ belong to the second category. Comparing asymptotically K_1 to K, we obtain

$$\begin{aligned}
K_1 &= o(K) & \text{if } \alpha \geq 1, 1/\alpha < \beta \leq 1 \\
K_1 &= \omega(K) & \text{if } \alpha \geq 1, \beta \leq 1/\alpha \text{ or } \alpha < 1, \beta \leq 1. \tag{35}
\end{aligned}$$

Therefore, when $\alpha < 1$, we set $K_1 \equiv K$, and we have only one video category. Now, to compute $\overline{S}$, we can just sum up the contributions of all videos, obtaining

$$\begin{aligned}
\overline{S} &= \sum_{k=1}^{K} \overline{S}_k \\
&\leq \sum_{k=1}^{K_1-1} 4\lambda_k^{\delta} e^{-C_{1,k}\lambda_k^{1-\delta}} + \sum_{k=K_1}^{K} d_k \tau_k \lambda_k. \tag{36}
\end{aligned}$$

We define $C \triangleq d_{\max}\tau_{\max}$, and substitute λ_k with its value $np_k = nH(K)k^{-\alpha}$. We obtain

$$\begin{aligned}
\overline{S} &\leq 4(nH(K))^{\delta} \sum_{k=1}^{K_1-1} k^{-\alpha\delta} e^{-C_{1,k}(nH(K)k^{-\alpha})^{1-\delta}} \\
&\quad + C n H(K) \sum_{k=K_1}^{K} k^{-\alpha}. \tag{37}
\end{aligned}$$

Let $\overline{S}_{up,1}$ be the first term in (37). Furthermore, since by (29) $C_{1,k} \triangleq \tau_k \frac{\overline{U}_k(1-\gamma_k)(1+\gamma_k)}{4\gamma_k}$, we have $C_{1,\inf} \triangleq \inf_k C_{1,k} = \tau_{\min}\overline{U}_{\min}\frac{\sigma(2-\sigma)}{4(1-\sigma)}$. Thus, we have

$$\begin{aligned}
\overline{S}_{up,1} &\triangleq 4(nH(K))^{\delta} \sum_{k=1}^{K_1-1} k^{-\alpha\delta} e^{-C_1(d_k,\tau_k,\gamma_k)(nH(K)k^{-\alpha})^{1-\delta}} \\
&\leq 4 \sum_{k=1}^{K_1-1} (nH(K)k^{-\alpha})^{\delta} e^{-C_{1,\inf}(nH(K)k^{-\alpha})^{1-\delta}} \\
&< \Theta\left(\int_1^{K_1-1} (nH(K)x^{-\alpha})^{\delta} e^{-C_{1,\inf}(nH(K)x^{-\alpha})^{1-\delta}} dx\right). \tag{38}
\end{aligned}$$

Now we make the substitution $y = (nH(K)x^{-\alpha})^{1-\delta}$ and get $dx = \frac{(nH(K))^{\frac{1}{\alpha}}}{\alpha(1-\delta)y^{1+\frac{1}{\alpha(1-\delta)}}} dy$. We have

$$\overline{S}_{up,1}$$
$$< \Theta\left(\frac{(nH(K))^{\frac{1}{\alpha}}}{\alpha(1-\delta)} \int_{(nH(K)(K_1(n)-1)^{-\alpha})^{1-\delta}}^{(nH(K))^{1-\delta}} e^{-yC_{1,\inf}} \frac{y^{\frac{\delta}{1-\delta}}}{y^{1+\frac{1}{\alpha(1-\delta)}}} dy\right).$$

If $y^{\frac{2\alpha\delta-\alpha-1}{\alpha(1-\delta)}} < 1$, that is if $\delta < 1/(2\alpha) + 1/2$, we obtain

$$\overline{S}_{up,1}$$
$$< \Theta\left(\frac{(nH(K))^{\frac{1}{\alpha}}}{\alpha(1-\delta)} \int_{(nH(K)(K_1-1)^{-\alpha})^{1-\delta}}^{(nH(K))^{1-\delta}} e^{-yC_{1,\inf}} y^{\frac{2\alpha\delta-\alpha-1}{\alpha(1-\delta)}} dy\right)$$
$$< \Theta\left(\frac{(nH(K))^{\frac{1}{\alpha}}}{\alpha(1-\delta)} \int_{(nH(K)(K_1-1)^{-\alpha})^{1-\delta}}^{(nH(K))^{1-\delta}} e^{-yC_{1,\inf}} dy\right)$$
$$< \Theta\left((nH(K))^{\frac{1}{\alpha}} e^{-C_{1,\inf}(nH(K)(K_1(n))^{-\alpha})^{1-\delta}}\right).$$

From (34) and (35), we get

$$\overline{S}_{up,1}$$
$$< \begin{cases} \Theta(n^{\frac{1}{\alpha}} e^{-C_{1,\inf}}) = O(n^{1/\alpha}), & \text{if } \alpha > 1, \beta > 1/\alpha \\ \Theta(n^{\frac{1}{\alpha}} e^{-C_{1,\inf}} n^{(1-\alpha\beta)(1-\delta)}) = o(1), & \text{if } \alpha > 1, \beta < 1/\alpha \\ \Theta(n \log n e^{-C_{1,\inf}}) = O(n \log n), & \text{if } \alpha = 1, \beta = 1 \\ \Theta(n \log n^{\beta} e^{-C_{1,\inf}(n^{1-\beta}\log n^{\beta})^{(1-\delta)}}) \\ \qquad = o(1), & \text{if } \alpha = 1, \beta < 1 \\ \Theta(n e^{-C_{1,\inf}}) = O(n), & \text{if } \alpha < 1, \beta = 1 \\ \Theta(n^{\beta(\alpha-1)+1} e^{-C_{1,\inf}n^{1-\beta}}) = o(1), & \text{if } \alpha < 1, \beta < 1. \end{cases}$$

Now we consider the second term in (37), $\overline{S}_{up,2} \triangleq CnH(K)\sum_{k=K_1}^{K} k^{-\alpha}$. Note that this term exists only when $K_1 = o(K)$; see (35). Since function $f(x) = x^{-\alpha}$ is decreasing, by the integral test for series we obtain, for $\alpha > 1$

$$\overline{S}_{up,2} < nH(K)\left(K_1^{-\alpha} + \int_{K_1}^{K} x^{-\alpha}\,\mathrm{d}x\right)$$
$$= nH(K)\left(K_1^{-\alpha} + \frac{K_1^{1-\alpha} - K^{1-\alpha}}{\alpha - 1}\right)$$
$$\leq nH(K)\left(K_1^{-\alpha} + K_1^{1-\alpha}/(\alpha - 1)\right)$$
$$= \Theta\left(nn^{-1} + nn^{\frac{1-\alpha}{\alpha}}\right) = \Theta\left(n^{\frac{1}{\alpha}}\right).$$

If $\alpha = 1$, with the same calculation as above we obtain

$$\overline{S}_{up,2} < nH(K)\left(K_1^{-1} + \int_{K_1}^{K} x^{-1}\,\mathrm{d}x\right)$$
$$= nH(K)\left(K_1^{-1} + \log K - \log K_1\right)$$
$$\leq nH(K)\left(K_1^{-1} + \log K\right)$$
$$= \Theta\left(n\log^{-1}n(n^{-1}\log^{-1}n + \log n)\right)$$
$$= \Theta(n). \tag{39}$$

Summing $\overline{S}_{up,1}$ and $\overline{S}_{up,2}$, we obtain

$$\overline{S} = \begin{cases} O(n^{1/\alpha}), & \text{if } \alpha > 1, \beta > 1/\alpha \\ o(1), & \text{if } (\alpha > 1, \beta < 1/\alpha) \vee (\alpha \leq 1, \beta < 1) \\ O(n), & \text{if } \alpha \leq 1, \beta = 1. \end{cases}$$

When, instead, $\mathbb{P}(\Gamma_k \leq 1) < 1$, for any $\beta > 0$ standard concentration arguments allow to say that for any function $f(n) \to \infty$, a finite fraction of videos with index $k \leq f(n)$ will have w.h.p. an associated load $\gamma_k > 1$. Since the associated request rate for such videos scales linearly with n (i.e., it scales faster that any sublinear function), as immediate consequence, $\overline{S}$ scales also linearly with n.

APPENDIX D
PROOF OF THEOREM 3

First observe that if $d_{\max} \leq \overline{U}$, we do not really need load equalization since $\gamma_k < 1 \ \forall k$ even when we set $q_k = 0 \ \forall k$, thus we focus on the case $d_{\max} \leq \overline{U}$ (we emphasize, however, that also in this case it is possible to devise a scheme that equalizes the load following the same lines as for the case $d_{\max} \leq \overline{U}$). Assuming $d_{\max} > \overline{U}$, observe that to perfectly balance the load among all videos, we need to make the average number of seeds for a video proportional to the average content downloading time. Indeed, resolving the equation $\gamma_k = \frac{d_{\max}\overline{N}'_{d,k}}{\overline{U}(\overline{N}'_{d,k} + \overline{N}'_{\text{seed},k})}$ with respect to $\overline{N}'_{\text{seed},k}$, we obtain $\overline{N}'_{\text{seed},k} = \frac{d_{\max} - \gamma_k\overline{U}}{\gamma_k\overline{U}}\overline{N}'_{d,k}$, thus to achieve $\gamma_k = \gamma, \forall k$ we need to enforce: $\overline{N}'_{\text{seed},k} = \frac{d_{\max} - \gamma\overline{U}}{\gamma\overline{U}}\overline{N}'_{d,k}$. Now since the distribution of downloaders among the videos is proportional, by construction, to the average content downloading time τ'_k, we get the assert.

Now according to the seed-freezing scheme, the average number of seeds for video k, $\overline{N}'_{\text{seed},k}$, can be partitioned into two components: the average number of unassigned peers that are currently acting as seeds for content k, and the average number of k-content assigned seeds that are currently seeding.

For what concerns the average number of unassigned seeds, we can easily obtain it, applying Little law as the average between the rate at which unassigned peers become seeds for content k times the average time they spend seeding content k: $(\Lambda + \Lambda_1)p_k(1 - q_k)\overline{T}_{\text{cont},k}$, where $\Lambda_1 = \frac{\Lambda(1 - \sum_k p_k q_k)}{\sum_k p_k q_k}$ represents the aggregate rate of unassigned peers in the system starting/ending a download. Similarly, the average number of assigned seeds can be computed again exploiting Little law as the rate at which peers are assigned to content k, $(\Lambda + \Lambda_1)p_k q_k$ times the average time they spend seeding content k as assigned seeds which is given $\overline{T}_{\text{cont},k} + (m - 1)\overline{T}_{\text{cont}}$; in this regard, observe that the number of contents downloaded by peers follows a geometric memoryless distribution.

Now considering any pair of different videos k, h, we can balance the load between them if we are able to find values $0 \leq q_k \leq 1$ and $0 \leq q_h \leq 1$, in correspondence of which the ratio between seeds $\frac{\overline{N}'_{\text{seed},k}}{\overline{N}'_{\text{seed},h}}$ becomes equal to the ratio between downloaders $\frac{\tau'_k}{\tau'_h}$, i.e., we have to force

$$\frac{\overline{N}'_{\text{seed},k}}{\overline{N}'_{\text{seed},h}} = \frac{\overline{T}_{\text{cont},k} + q_k[(m-1)\overline{T}_{\text{cont}}]}{\overline{T}_{\text{cont},h} + q_h[(m-1)\overline{T}_{\text{cont}}]} = \frac{\tau'_k}{\tau'_h}.$$

To achieve our goal under (8), first we sort all videos according to the associated metric $\frac{\tau'_k}{\overline{T}_{\text{cont},k}}$. Let k_0 be the video that minimizes the associated metric: $k_0 = \arg\min_k \frac{\tau'_k}{\overline{T}_{\text{cont},k}}$. We set $q_{k_0} = 0$, for any other video k, we obtain q_k as solution of

$$\frac{\overline{T}_{\text{cont},k} + q_k[(m-1)\overline{T}_{\text{cont}}]}{\overline{T}_{\text{cont},k_0}} = \frac{\tau'_k}{\tau'_{k_0}}$$

i.e., $q_k = \frac{1}{(D-1)\overline{T}_{\text{cont}}}\left(\frac{\tau'_k\overline{T}_{\text{cont},k_0}}{\tau'_{k_0}} - \overline{T}_{\text{cont},k}\right)$. Observe, however, that q_k is feasible only if it lies in the interval $[0,1]$; $q_k \geq 0$ since, by construction, $\left(\frac{\tau'_k\overline{T}_{\text{cont},k_0}}{\tau'_{k_0}} - \overline{T}_{\text{cont},k}\right) \geq \left(\tau'_k\frac{\overline{T}_{\text{cont},k}}{\tau'_k} - \overline{T}_{\text{cont},k}\right) = 0$. Furthermore, it turns out that $q_k \leq 1, \forall k$ iff (8) is satisfied.

APPENDIX E
PROOF OF THEOREM 5

We first consider the load associated to individual stripes (or videos). Note that we need to worry only about stripes requiring bundling (i.e., those belonging to videos with $\gamma_k \geq \gamma$) since, by construction, the load of stripes initially having load $\gamma_k < \gamma$ becomes, after applying the proposed equalization technique, exactly equal to $\gamma < \gamma'$.

Consider a generic stripe i belonging to a video k having an initial load $\gamma_k \geq \gamma$. Let $\mathcal{S}_i$ be the set of indexes of videos (having initial load smaller than γ) that are assigned (in total or in part) to assist the distribution of stripe i by the bundling algorithm. After equalization, the new load induced by stripe i (or, equivalently, by video k) is

$$\gamma_i = \frac{\lambda_k\frac{l_k}{M} + \sum_{j\in\mathcal{S}_i}\lambda_j p_{j,i}\frac{l_k}{M}}{\lambda_k\frac{l_k}{M\gamma}} = \gamma\left(1 + \frac{\sum_{j\in\mathcal{S}_i}\lambda_j p_{j,i}}{\lambda_k}\right) \tag{40}$$

where $p_{j,i}$ is the probability that a peer downloading video j has concurrently to download stripe i (see scheme in Section V-B). Unfortunately, the r.h.s. of (40) cannot always be made smaller than γ' by increasing M. Indeed, in the worst possible case,

all videos (except video k) are assigned to video k, obtaining the bound $\sum_{j \in \mathcal{S}_i} \lambda_j p_{j,i} \leq \Lambda/M$ (notice that the M stripes of video k are equally bundled). However, the resulting ratio $\Lambda/(M\lambda_k)$ can, in some cases, go to infinite as the system size increases (for any finite M), especially for very unpopular videos k. Note that this worst case can only happen when videos $j \in \mathcal{S}_i$ contribute a vanishing amount of bandwidth ΔU_j to the bundled stripe, making the scheme inefficient.

To overcome this problem, we need to modify the amount of bandwidth devoted to the distribution of bundled data.[8] Specifically, for each video j having $\gamma_j < \gamma$, we compute the amount of bandwidth $\Delta U_j'$ that, if subtracted to the average upload bandwidth of a peer, makes the load of video j equal to $(\gamma + \gamma')/2$

$$\frac{d_j \tau_j}{(\overline{U} - \Delta U_j')\overline{T}_j} = \frac{\gamma + \gamma'}{2}.$$

We remark that, under the modified scheme, the selection of the stripe to be bundled to videos with an excess of bandwidth is still performed by the algorithm described in Section V-B, which is based on the original values of ΔU_j. Simple calculations lead to the following equation:

$$\Delta U_j' = \Delta U_j + \overline{U}\frac{\gamma_j}{\gamma'}\frac{\gamma' - \gamma}{\gamma' + \gamma}$$

which permits to rewrite (40) as

$$\gamma_i = \frac{\lambda_k \frac{l_k}{M} + \sum_{j \in \mathcal{S}_i} \lambda_j p_{j,i} \frac{l_k}{M}}{\lambda_k \frac{l_k}{M\gamma} + \sum_{j \in \mathcal{S}_i} \lambda_j p_{j,i} \frac{l_j}{\gamma'}\frac{\gamma' - \gamma}{\gamma' + \gamma}}. \tag{41}$$

Since $\gamma < \gamma'$, and using the fact that $l_k \leq l_{\max}$ and $l_j \geq l_{\min}$, $\forall j$, a sufficient condition such that the r.h.s. of (41) is smaller than γ' is

$$M > \frac{l_{\max}}{l_{\min}}\frac{\gamma' + \gamma}{\gamma' - \gamma} = M_1. \tag{42}$$

At last, we show that also the overall system load, including the impact of bundling, can be kept smaller than γ' by a proper selection of M. Indeed, in the worst possible case, all peers downloading a video have to additionally download a stripe of the largest video. Hence, the overall system load, after applying the equalization scheme, can be upper-bounded by

$$\frac{\sum_{k=1}^{K} \lambda_k l_k + \frac{l_{\max}}{M}}{\sum_{k=1}^{K} \lambda_k \overline{U}(\tau_k + \overline{T}_{\text{seed}})}.$$

The above quantity is smaller than γ' whenever

$$M > \frac{l_{\max}}{l_{\min}}\frac{\gamma}{\gamma' - \gamma} = M_2.$$

Since $M_1 > M_2$, by selecting $M^* = M_1$, we satisfy all requirements about global, video-specific, and stripe-specific loads. ∎

APPENDIX F
PROOF OF THEOREM 6

For the sake of clarity, we prove the theorem in the case of a load balancing strategy by seed allocation (as described in Section V-A), so that the system works at global load γ' such that for all file k, $\gamma_k < \gamma' < 1$. We suppose also to apply a full video catalog warming approach (Section V-C). The proof can be easily extended in the other cases, for example using stripe bundling (Section V-B). Note that in this case each video

---

is divided into M stripes, and that $\overline{U}_k$ becomes the users' upload bandwidth devoted to stripe k.

In particular, we increase artificially the popularity of the file whose λ_k is smaller than a poly-log function of the number of active users n. More precisely, we impose the following new request rate: For a certain $z > 0$ and $c > 0$

$$\lambda_k' = \max\{\lambda_k, c(\log n)^z\} \qquad \forall k \leq K_2' \tag{43}$$

where K_2' is a threshold on the number of videos with increased popularity, such that the system load γ is still smaller than 1. Moreover, there exists a threshold K_1' such that $\lambda_{K_1'}' = c(\log n)^z$. Thus, we have that

$$K_1' = c^{-1/\alpha}(\log n)^{-z/\alpha} n^{1/\alpha} H(K)^{1/\alpha}$$
$$= \Theta(c^{-1/\alpha}(\log n)^{-z/\alpha} n^{1/\alpha})$$

where the last equality follows from the fact that $H(K) = \Theta(1)$ if $\alpha > 1$. To sum up, we can distinguish among three video categories

$$\lambda_k' = \begin{cases} \lambda_k, & \text{if } 1 \leq k < K_1' \\ c(\log n)^z, & \text{if } K_1' \leq k < K_2' \\ \lambda_k, & \text{if } K_2' \leq k \leq K. \end{cases} \tag{44}$$

By (4), we know that system load γ is proportional to $\Lambda = \Theta(n)$. Thus, we need to verify that the modified popularities λ_k' do not change the asymptotical behavior of Λ. We have

$$\sum_{k=K_1'}^{K_2'} \lambda_k' = c(K_2' - K_1')(\log n)^z$$

$$\leq cK_2'(\log n)^z.$$

If $\beta < 1$, and $K_2' \equiv K$, the previous formula becomes

$$\sum_{k=K_1'}^{K_2'} \lambda_k' = O(n^\beta (\log n)^z) = o(n)$$

thus we can conclude that the asymptotical behavior of Λ is the same as the passive strategy, and so the system load is still smaller than 1.

On the other hand, if $\beta = 1$, we cannot set $K_2' \equiv K$ because we would obtain $\Lambda = \omega(n)$. The maximum value for K_2' is thus $K_2' \equiv n/(\log n)^z$, so that we obtain

$$\sum_{k=K_1'}^{K_2'} \lambda_k' = O(n).$$

Now we can compute the bandwidth requested from servers and show that it tends to zero as the number of users n grows to infinity. Essentially, we use the reasonings in the proof of Theorem 2, in Appendix C. Note that $K_1' = \Theta(c^{-1/\alpha}(\log n)^{-z/\alpha} n^{1/\alpha})$ is asymptotically smaller than $K_1 = \Theta(n^{1/\alpha})$. As in (36), we compute $\overline{S}$ as the sum of the contributions for all the videos in the three categories

$$\overline{S} = \overline{S}_{up,1}' + \overline{S}_{up,2}' + \overline{S}_{up,3}'$$

$$\leq \sum_{k=1}^{K_1'-1} 4\lambda_k^\delta e^{-C_{1,\inf}\lambda_k^{1-\delta}}$$

$$+ \sum_{k=K_1'}^{K_2'-1} 4c^\delta (\log n)^{z\delta} e^{-C_{1,\inf} c^{(1-\delta)}(\log n)^{z(1-\delta)}}$$

$$+ \sum_{k=K_2'}^{K} d_k \tau_k \lambda_k.$$

For the first sum in the previous equation, we can use the same computation as in (38), and we obtain

$$\overline{S}'_{up,1} \leq \Theta\left((nH(K))^{\frac{1}{\alpha}} e^{-C_{1,\inf}(nH(K)(K'_1(n))^{-\alpha})^{1-\delta}} \right)$$
$$= \Theta\left(n^{\frac{1}{\alpha}} e^{-C_{1,\inf}c^{1-\delta}(\log n)^{z(1-\delta)}} \right). \quad (45)$$

Now we can set $c = \left(C_{1,\inf}^{-1}(1+\epsilon) \right)^{\frac{1}{1-\delta}}$ and $z = \frac{1}{1-\delta}$, for an $\epsilon > 0$, and we get

$$\overline{S}'_{up,1} = O\left(n^{\frac{1}{\alpha}} n^{-(1+\epsilon)} = o(1). \right) \quad (46)$$

For the second sum, we obtain

$$\overline{S}'_{up,2} < K'_2 4c^{\delta}(\log n)^{z\delta} e^{-C_{1,\inf}c^{1-\delta}(\log n)^{z(1-\delta)}}. \quad (47)$$

If $\beta < 1$, we have that $K'_2 \equiv K = \Theta(n^{\beta})$, thus $\overline{S}'_{up,3} = 0$, and we get

$$\overline{S}'_{up,2} = O\left(n^{\beta} c^{\delta}(\log n)^{z\delta} n^{-(1+\epsilon)} \right) = o(1). \quad (48)$$

Thus, when $\alpha > 1, 1/\alpha < \beta < 1$, we conclude that $\overline{S} = o(1)$.

If $\alpha > 1$ and $\beta = 1$, we get

$$\overline{S}'_{up,2} = O\left(\frac{n}{(\log n)^z} c^{\delta}(\log n)^{z\delta} n^{-(1+\epsilon)} \right) = o(1). \quad (49)$$

For the third part, analogously to (39), we obtain

$$\overline{S}'_{up,3} = O\left(nH(K)\left((K'_2)^{1-\alpha}/(\alpha-1)\right) \right)$$
$$= O\left(n(\frac{n}{(\log n)^z})^{1-\alpha} \right)$$
$$= O\left(n^{2-\alpha}(\log n)^{\frac{\alpha-1}{1-\delta}} \right). \quad (50)$$

Thus, we conclude that $\overline{S} = \Theta(\overline{S}'_{up,3})$ when $\alpha > 1$ and $\beta = 1$ and $\overline{S} = o(1)$ when $\alpha > 1, \beta < 1$.

APPENDIX G
PROOF OF THEOREM 7

We want to compute a lower bound on $\overline{S}$ based on (16). By Lemma 1, we know that $N_{d,k}$ is a Poisson distributed random variable with parameter λ_k. We have that

$$\mathbb{P}(N_{d,k} > 0) = 1 - e^{-\lambda_k \tau_k}.$$

If $\lambda_k \tau_k \to 0$, we can use the Taylor expansion for the previous expression, and we get

$$\mathbb{P}(N_{d,k} > 0) = 1 - e^{-\lambda_k \tau_k} = \lambda_k \tau_k + o(\lambda_k \tau_k)$$

thus we obtain that $\mathbb{P}(N_{d,k} > 0) = \Theta(\lambda_k \tau_k)$ if $\lambda_k \tau_k \to 0$. On the other hand, when $\lambda_k \tau_k$ does not tend to zero, we can always say that $\mathbb{P}(N_{d,k} > 0) = \Theta(1)$. In the end, we obtain

$$\mathbb{P}(N_{d,k} > 0) = \Theta\left(\min(1, \lambda_k \tau_k) \right). \quad (51)$$

Consider now the probability $\mathbb{P}(N_{\text{seed},k} = 0)$, which is the probability that no peer is acting as a seed for file k: A peer is not acting as a seed if it is in sleeping mode. We know that the number of seed $N_{\text{seed},k}$ is a Poisson distributed random variable with parameter $\overline{N}_{\text{seed},k}$

$$\mathbb{P}(N_{\text{seed},k} = 0) = e^{-\overline{N}_{\text{seed},k}}. \quad (52)$$

We use now (51) and (52), and therefore (16) becomes

$$\overline{S} \geq \sum_{k=1}^{K} d_k \mathbb{P}(N_{d,k} > 0)\mathbb{P}(N_{\text{seed},k} = 0)$$
$$= \Theta\left(\sum_{k=1}^{K} d_k \left(\min(1, \lambda_k \tau_k) \right) e^{-\overline{N}_{\text{seed},k}} \right). \quad (53)$$

Remembering that the minimum λ_k is equal to

$$\lambda_K = \inf_k \lambda_k$$
$$= \Theta(K^{-\alpha}nH(K))$$
$$= \begin{cases} \Theta(n^{1-\beta\alpha}), & \text{if } \alpha > 1 \\ \Theta(n^{1-\beta}\log n), & \text{if } \alpha = 1 \\ \Theta(n^{1-\beta}), & \text{if } \alpha < 1 \end{cases} \quad (54)$$

and that the quantities $d_{\min}$ and $\tau_{\min}$ do not depend on n, we obtain that

$$\overline{S} \geq \Theta\left(d_{\min}\lambda_K \tau_{\min} \sum_{k=1}^{K} e^{-\overline{N}_{\text{seed},k}} \right)$$
$$= \Theta\left(d_{\min}\lambda_K \tau_{\min} \sum_{k=1}^{K} e^{-\overline{N}_{\text{seed},k}} \right)$$
$$\geq \Theta\left(d_{\min}\lambda_K \tau_{\min} \int_{1}^{K+1} e^{-\overline{N}_{\text{seed}}(x)} dx \right). \quad (55)$$

We use now Jensen's inequality in the following form. Let φ be a convex function on the real line and let $f : [a, b] \subseteq \mathbb{R} \to \mathbb{R}$ be a nonnegative real-valued function that is Lebesgue-integrable. Then, it holds

$$\int_{a}^{b} \varphi((b-a)f(x)) dx \geq (b-a)\varphi\left(\int_{a}^{b} f(x)dx \right). \quad (56)$$

We make use of (56) with $a = 1$, $b = K+1$, $f(x) = \overline{N}_{\text{seed}}(x)/K$, $\varphi(x) = e^{-x}$ and we get

$$\overline{S} \geq \Theta\left(d_{\min}\tau_{\min}\lambda_K K e^{-\int_{1}^{K+1} \overline{N}_{\text{seed}}(x)/K dx} \right)$$
$$= \Theta\left(d_{\min}\tau_{\min}\lambda_K K e^{-n/K} \right)$$
$$= \Theta\left(d_{\min}\tau_{\min}\lambda_K n^{\beta} e^{-n^{1-\beta}} \right)$$
$$= \begin{cases} \Theta\left(d_{\min}\tau_{\min}n^{1-\beta\alpha}n^{\beta} e^{-n^{1-\beta}} \right), & \text{if } \alpha > 1 \\ \Theta\left(d_{\min}\tau_{\min}n \log n e^{-n^{1-\beta}} \right), & \text{if } \alpha = 1 \\ \Theta\left(d_{\min}\tau_{\min}n e^{-n^{1-\beta}} \right), & \text{if } \alpha = 1. \end{cases} \quad (57)$$

In the previous formula, every time $\beta < 1$, the exponential function $e^{-n^{1-\beta}}$ goes to zero as $n \to \infty$, and we get a trivial lower bound $\overline{S} \geq 0$. On the other hand, if $\beta = 1$, we obtain

$$\overline{S} \geq \begin{cases} \Theta\left(n^{2-\alpha}\right), & \text{if } \alpha > 1, \beta = 1 \\ \Theta\left(n \log n\right), & \text{if } \alpha = 1 \beta = 1 \\ \Theta\left(n\right), & \text{if } \alpha = 1, \beta = 1. \end{cases} \quad (58)$$

Note that since n is a trivial upper bound to $\overline{S}$, in the end we obtain the following result:

$$\overline{S} \geq \Theta\left(n^{2-\alpha}e^{-1}\right), \quad \text{if } \alpha > 1, \beta = 1$$
$$\overline{S} = \Theta(n), \quad \text{if } \alpha \leq 1, \beta = 1. \quad (59)$$

REFERENCES

[1] Cisco, San Jose, CA, USA, "Cisco visual networking index: Forecast and methodology, 2011–2016," White paper, 2012.

[2] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-assured cloud bandwidth auto-scaling for video-on-demand applications," in *Proc. IEEE IN-FOCOM*, 2012, pp. 460–468.

[3] C. Huang, J. Li, and K. Ross, "Can Internet video-on-demand be profitable?," in *Proc. ACM SIGCOMM*, 2007, pp. 133–144.

[4] Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale P2P VoD system," in *Proc. ACM SIGCOMM*, 2008, pp. 375–388.

[5] Softonic, Barcelona, Spain, "PPLive," [Online]. Available: http://pplive.en.softonic.com

[6] "GridCast," [Online]. Available: http://www.gridcast.cn/

[7] "PPStream," 2006 [Online]. Available: http://www.ppstream.com/

[8] TVU Netwworks Corporation, Mountain View, CA, USA, "TVU," [Online]. Available: http://www.tvunetworks.com/

[9] W. Wu, R. Ma, and J. Lui, "On incentivizing caching for P2P-VoD systems," in *Proc. NetEcon Workshop*, 2012, pp. 164–169.

[10] F. Dobrian *et al.*, "Understanding the impact of video quality on user engagement," in *Proc. ACM SIGCOMM*, 2011, pp. 362–373.

[11] D. Ciullo, V. Martina, M. Garetto, E. Leonardi, and G. L. Torrisi, "Stochastic analysis of self-sustainability in peer-assisted VoD systems," in *Proc. IEEE INFOCOM*, 2012, pp. 1539–1547.

[12] D. Wu, Y. Liu, and K. Ross, "Modeling and analysis of multichannel P2P live video systems," *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1248–1260, Aug. 2010.

[13] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. IMC*, 2007, pp. 1–14.

[14] W. Wu and J. C. Lui, "Exploring the optimal replication strategy in P2P-VoD systems: Characterization and evaluation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 8, pp. 1492–1503, Aug. 2012.

[15] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proc. IEEE INFOCOM*, 2007, pp. 919–927.

[16] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance bounds for peer-assisted live streaming," in *Proc. SIGMET-RICS*, 2008, pp. 313–324.

[17] C. Wu, B. Li, and S. Zhao, "On dynamic server provisioning in multichannel P2P live streaming," *IEEE/ACM Trans. Netw*, vol. 19, no. 5, pp. 1317–1330, Oct. 2011.

[18] Y. Zhou, T. Z. J. Fu, and D. M. Chiu, "Statistical modeling and analysis of P2P replication to support VoD service," in *Proc. IEEE INFOCOM*, 2011, pp. 945–953.

[19] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "CloudMedia: When cloud on demand meets video on demand," in *Proc. ICDCS*, 2011, pp. 268–277.

[20] C. Huang, A. Wang, J. Li, and K. W. Ross, "Understanding hybrid CDN-P2P: Why LimeLight needs its own red swoosh," in *Proc. NOSSDAV*, 2008, pp. 75–80.

[21] B. Fan, D. G. Andersen, M. Kaminsky, and K. Papagiannaki, "Balancing throughput, robustness, and in-order delivery in P2P VoD," in *Proc. ACM CoNEXT*, 2010, Art. no. 10.

[22] K. Mokhtarian and M. Hefeeda, "Analysis of peer-assisted video-on-demand systems with scalable video streams," in *Proc. MMSys*, 2010, pp. 133–144.

[23] C. Zhao, X. Lin, and C. Wu, "The streaming capacity of sparsely-connected P2P systems with distributed control," in *Proc. IEEE IN-FOCOM*, 2011, pp. 1449–1457.

[24] C. Zhao, J. Zhao, X. Lin, and C. Wu, "Capacity of P2P on-demand streaming with simple, robust and decentralized control," in *Proc. IEEE INFOCOM*, 2013, pp. 2697–2705.

[25] P. Dhungel, K. Ross, M. Steiner, Y. Tian, and X. Hei, "Xunlei: Peer-assisted download acceleration on a massive scale," in *Proc. PAM*, 2012, pp. 231–241.

Delia Ciullo received the Master's degree in telecommunications engineering and Ph.D. degree in electronics and communications engineering from Politecnico di Torino, Turin, Italy, in 2007 and 2011, respectively.

In 2009, she was a visiting student with the CNRG Group, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. Between 2012 and 2013, she was a Post-Doctoral ERCIM Fellow with INRIA Sophia Antipolis, Valbonne, France. She is currently a Post-Doctoral Researcher with EURECOM Sophia-Antipolis, Biot, France.

Valentina Martina received the Master's degree in mathematical modeling in engineering and Ph.D. degree in electronics and communication engineering from Politecnico di Torino, Turin, Italy, in 2007 and 2011, respectively.

In 2010, she was a visiting student with the Technicolor Paris Research Lab, Paris, France. She is currently a Post-Doctoral Researcher with Politecnico di Torino.

Michele Garetto (M'04) received the Dr.Ing. degree in telecommunication engineering and Ph.D. degree in electronic and telecommunication engineering from Politecnico di Torino, Turin, Italy, in 2000 and 2004, respectively.

In 2002, he was a Visiting Scholar with the Networks Group, University of Massachusetts, Amherst, MA, USA, and in 2004, he held a post-doctoral position with the Electrical and Computer Engineering Department, Rice University, Houston, TX, USA. He is currently an Assistant Professor with the University of Torino, Turin, Italy.

Emilio Leonardi (M'99–SM'09) received the Dr.Ing. degree in electronics engineering and Ph.D. degree in telecommunications engineering from Politecnico di Torino, Turin, Italy, in 1991 and 1995, respectively.

He is an Associate Professor with the Dipartimento di Elettronica, Politecnico di Torino. His research interests are in the field of performance evaluation of wireless networks, P2P systems, and packet switching.