Performance analysis of non-stationary peer-assisted VoD systems

(Article begins on next page)

25 July 2024

Performance Analysis of Non-stationary Peer-assisted VoD Systems

Delia Ciullo [*], Valentina Martina [*], Michele Garetto [†], Emilio Leonardi [*], Giovanni Luca Torrisi [‡]

[*] Dipartimento di Elettronica, Politecnico di Torino, Torino, Italy
[†] Dipartimento di Informatica, Università di Torino, Torino, Italy
[‡] Istituto per le Applicazioni del Calcolo, CNR, Roma, Italy

Abstract—We analyze a peer-assisted Video-on-Demand system in which users contribute their upload bandwidth to the redistribution of a video that they are downloading or that they have cached locally. Our target is to characterize the additional bandwidth that servers must supply to immediately satisfy all requests to watch a given video. We develop an approximate fluid model to compute the required server bandwidth in the sequential delivery case. Our approach is able to capture several stochastic effects related to peer churn, upload bandwidth heterogeneity, non-stationary traffic conditions, which have not been documented or analyzed before. We provide an analytical methodology to design efficient peer-assisted VoD systems and optimal resource allocation strategies.

I. INTRODUCTION

In this paper, we focus on peer-assisted Video-on-Demand (VoD) systems, in which users can browse a catalog of available videos and asynchronously issue requests to watch a given content. In peer-assisted VoD systems, users interested to a specific video can retrieve it from servers (CDN modality), from peers downloading/watching it, and from users storing a copy of it in their computer/Internet TV memory or in dedicated set-top-boxes remotely controllable by the network operator [1].

Our main contribution is a stochastic fluid framework that allows to approximately estimate the additional bandwidth that servers must provide to satisfy all requests to watch a given video. A stochastic fluid approach to analyze peer-assisted video distribution has been proposed in [2] in the context of live streaming, in which (heterogeneous) peers download and playback content synchronously. Here we apply the stochastic fluid approach to VoD systems, whose dynamics are quite different from live streaming, since users can watch the video asynchronously. A mathematical formulation of the server bandwidth needed under sequential delivery appeared in [3], in which authors resort to a Monte Carlo approach to get basic insights into the system behavior (like surplus and deficit modes). The sequential delivery scheme has been considered also in [4], 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. Differently from previous work, our methodology can account for several stochastic effects related to upload bandwidth heterogeneity and non-stationary traffic conditions, which have not been analyzed before, providing a useful tool for the analysis and design of VoD systems.

The analytical approach described in this paper complements the analysis presented in a companion paper [5] in which we obtain rigorous bounds for the sequential delivery scheme (under stationary traffic conditions) and asymptotic results as the number of users increases. With respect to [5], we extend the analysis to non-stationary traffic conditions, with a different goal in mind, *i.e.*, to provide a performance evaluation tool that can be readily used for system design and optimization.

We emphasize that in our work we do not consider issues related to optimal replication strategies of heterogeneous contents (in size and popularity) or optimal peer resource allocation (in terms of storage and upload bandwidth) in the presence of multiple videos. This because we focus on the bandwidth requested from the servers to distribute a given video, assuming that the peer resources allocated to it (i.e., number of copies available in the system and the amount of upload bandwidth devoted to the considered video) are given. Although we focus on a single video, our analysis can be combined with optimal resource and replication strategies for the case of multiple videos.

II. MODEL

A. System assumptions

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 till 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. Users contribute their upload bandwidth to the video distribution, thus they can retrieve part of the video (or even the entire video) from other peers, saving servers resources.

We focus on a given video of duration T_v seconds and size L bytes, which is played back by the users' applications at rate $d_v = L/T_v$ bytes/s. Clearly, to guarantee continuous playback each user must *at least* receive video chunks sequentially at rate d_v. As a widely adopted strategy to mitigate bandwidth fluctuations, applications pre-fetch and buffer video chunks before playback. In our model, we assume that the system provides to each user a fixed download rate $d \geq d_v$ (we assume unlimited download bandwidth at each user). Notice that the download rate d can be chosen by the system. We will show that in some cases, unexpectedly, the optimal value of d (*i.e.*, the one that minimizes the average bandwidth requested from the servers) is actually larger than d_v.

The amount of upload bandwidth with which peers contribute to the redistribution of the video that they are downloading may or may not be under the control of the system. In our analysis, we assume that the upload bandwidth available at a peer is a random variable with given distribution. The amount

of upload bandwidth with which users contribute at a given time to the redistribution of the considered video is modeled by a random variable U with cumulative distribution function $F_U(w)$, mean $\overline{U}$ and variance σ_U^2. The random variables denoting the instantaneous upload bandwidths of the users are assumed to be i.i.d. (identically and independently distributed).

B. Peers dynamics

We assume that the arrival process of requests for the considered video follows a time-varying Poisson process of intensity $\lambda(t)$; so doing our model is able to capture non stationary dynamics induced by either daily traffic fluctuations or rapidly-changing video popularity.

The dynamics of peer participation in the distribution of a given video must account for the fact that activity periods of the users are highly heterogeneous, as observed in several measurement studies [3]: some users stop watching the video after a very short time since the beginning; most users who decide to watch the video shut down the computer/Internet-TV towards the end of it; some of them keep the application running for prolonged time after the end of the video; those running set-top-boxes can be considered to be always active and serving other peers (until they stop contributing to the distribution of the considered video). We account for general user behavior assuming that the activity period of a user (*i.e.*, the interval during which a user contributes its upload bandwidth to the system, starting from the instant at which the video has been requested) is described by an arbitrary random variable T with finite mean $\overline{T}$ and complementary cumulative distribution function $G_T(x)$. The activity periods of the users are assumed to be i.i.d.

It follows from our assumptions that the number of active users $N(t)$ at time t is distributed as the number of customers in an M/G/∞ queue with time-varying arrival rate, hence it follows a Poisson distribution with time-varying mean $\overline{N}(t)$ given by

$$\overline{N}(t) = \int_0^\infty \lambda(t-x)G_T(x)\,\mathrm{d}x \qquad (1)$$

In our analysis we need to distinguish two classes of active users: those who are still *downloading* the video, and those who have completed the download (referred to as *seeds* in the following). Let $\tau_d = L/d$ be the time needed to download the whole video, and $\overline{T}_d = \int_0^{\tau_d} G_T(x)dx$ the average time spent by peers downloading the video. The number of *downloading* peers at time t, denoted by $N_d(t)$, follows a Poisson distribution of mean $\overline{N}_d(t)$ given by

$$\overline{N}_d(t) = \int_0^{\tau_d} \lambda(t-x)G_T(x)\,\mathrm{d}x \qquad (2)$$

Then standard properties of Poisson processes allow to say that the number of *seeds* at time t, denoted by $N_\text{seed}(t)$, follows a Poisson distribution of mean $\overline{N}_\text{seed}(t) = \overline{N}(t) - \overline{N}_d(t)$.

We define as instantaneous system load $\gamma(t)$ the quantity

$$\gamma(t) = \frac{d \cdot \overline{N}_d(t)}{\overline{U} \cdot \overline{N}(t)} \qquad (3)$$

which is the ratio between the average data rate requested at time t by downloading peers and the average upload rate provided by all active users at time t. Borrowing the terminology adopted in previous work [6], [4] we say that

at time t the system operates in *deficit* mode if $\gamma(t) > 1$, in *balanced* mode if $\gamma(t) = 1$, and in *surplus* mode if $\gamma(t) \leq 1$.

We also introduce the per-user system load $\gamma_p = \frac{d \cdot \overline{T}_d}{\overline{U} \cdot \overline{T}}$, which is the ratio between the average amount of data that are downloaded by a peer, and the average amount of data that a peer is able to offer to other peers. Note that by construction γ_p is equal to the (constant) instantaneous system load in the case of a stationary user arrival process. In ergodic systems, γ_p can be regarded as the time average of $\gamma(t)$.

C. Performance metrics

A fundamental goal of a VoD system is to minimize the bandwidth required from the servers. Let $S(t)$ be the random variable denoting the additional bandwidth that the servers must supply at time t to satisfy all active downloads of the considered video. We denote by $\overline{S}(t)$ and $\sigma_S^2(t)$ the mean and variance of $S(t)$, respectively.

Since in practice there are multiple videos to be served concurrently by the system, statistical multiplexing arguments suggest that a good design goal is to minimize the mean value $\overline{S}(t)$ of the server bandwidth required by a single video. Therefore, this will be the main metric that we will look at in our performance analysis.

III. ANALYSIS

We consider the simple case in which users download the video chunks sequentially. This scheme is simple to implement, as it does not require complex chunk/peer selection mechanisms such as those needed in BitTorrent-like chunk swarming schemes. More importantly, the sequential delivery scheme is analytically tractable and provides an upper bound to the server bandwidth requested by non-sequential schemes.

Let $S_d(t)$ be the aggregate bandwidth requested by the downloading users at time t, and $S_\text{seed}(t) = \sum_{i=1}^{N_\text{seed}(t)} U_i$ be the aggregate upload rate offered by the seeds at time t. Then the bandwidth requested from the servers at time t is given by

$$S(t) = \sup\{0, S_d(t) - S_\text{seed}(t)\}. \qquad (4)$$

Focusing on $S_d(t)$, we first condition this quantity on the number of downloading users k, defining

$$S_d(k) \triangleq (S_d(t) \mid N_d(t) = k)$$

After characterizing $S_d(k)$, the evaluation of $S(t)$ is easy, since the distribution of $N_d(t)$ is known (a Poisson distribution of mean $\overline{N}(t)$), while $S_\text{seed}(t)$ is a compound Poisson random variable which does not depend on k.

To evaluate $S_d(k)$ under sequential download, we start observing that, if all peers download the video sequentially at common rate d, a peer can only redistribute video pieces to peers arrived later on in time.

Proposition 1: Quantity $S_d(k)$ satisfies the following recursive equation:

$$S_d(k) = \begin{cases} d & k = 1 \\ d + \max\{0, S_d(k-1) - U_k\} & k > 1 \end{cases} \qquad (5)$$

Proof: The case $k = 1$ is obvious. The recursive expression for $k \geq 2$ can be easily explained if we look at the users in reverse order with respect to the arrival time into the system, *i.e.*, user k arrives before user $k-1$. Suppose that we know the server bandwidth $S_d(k-1)$ needed in the presence of $k-1$ users. Then user k can reduce this rate by its upload bandwidth

U_k, possibly bringing the server rate to zero. Instead, user k cannot be helped by any other peers, hence it requires fresh new content from the server at rate d. ∎

The expression in (5) provides the key to the analytical approximation developed in the next section.

Alternate formulations of quantity $S_d(k)$ exist (see [6], [4], [5]). Here, we just mention that in our companion paper [5] we find a connection between the stochastic process described by (5) and a random walk with increments $d - U$, which allows to obtain analytical upper bounds to the server bandwidth and to characterize its asymptotic behavior for large number of users.

A. Gaussian approximation

In the sequential delivery case, we can characterize the distribution of the server bandwidth using a second-order approximation. The idea is to approximate the distribution of the quantity $S_d(k-1) - U_k$ in (5) (for each $k \geq 2$) by a normal distribution matching the first two moments of this quantity. We can then apply standard formulas of the truncated normal distribution to derive the first two moments of $S_d(k)$ as a function of the first two moments of $S_d(k-1)$. This provides a recursive technique to compute the first two moments of $S_d(k)$ for any k, starting from the exact values known for $k = 1$. A similar approximation is subsequently applied to take into account the effect of the *seeds*.

Let $\mathcal{N}(w)$ be the probability density function of the standard normal distribution (having mean 0 and variance 1), and $Q(w)$ its complementary cumulative distribution function.

Let y be a random variable distributed according to a normal distribution $N(\mu, \sigma)$ of mean μ and standard deviation σ. Then it can be proved that the first moment of the random variable $y' = \max\{0, y\}$ has the following expression:

$$\mathbb{E}[y'] = \sigma \mathcal{N}\left(-\frac{\mu}{\sigma}\right) + \mu\, Q\left(-\frac{\mu}{\sigma}\right) \tag{6}$$

while the second moment is given by

$$\mathbb{E}[y'^2] = \sigma \mu \mathcal{N}\left(-\frac{\mu}{\sigma}\right) + (\sigma^2 + \mu^2) Q\left(-\frac{\mu}{\sigma}\right). \tag{7}$$

Let $\overline{S}_d(k)$ and $\sigma^2_{S_d}(k)$ be the mean and variance of $S_d(k)$. Our recursive procedure to approximately compute $\overline{S}_d(k)$ and $\sigma^2_{S_d}(k)$ for all k starts from the initial known values $\overline{S}_d(1) = d$ and $\sigma^2_{S_d}(1) = 0$ (see (5)). Now for a given $k \geq 2$ we approximate $S_d(k-1) - U_k$ by a normal random variable y of mean $\mu = \overline{S}_d(k-1) - \overline{U}$ and variance $\sigma^2 = \sigma^2_{S_d}(k-1) + \sigma^2_U$. Defining the random variable $y' \triangleq \max\{0, y\} \simeq \max\{0, S_d(k-1) - U\}$, from (5) we obtain:

$$\overline{S}_d(k) \simeq d + \mathbb{E}[y'] \tag{8}$$

$$\sigma^2_{S_d}(k) \simeq \mathbb{E}[y'^2] - \mathbb{E}[y']^2. \tag{9}$$

Finally, applying (6) and (7) we can compute the first and second moment of variable y' in (8,9). This provides the recursion to compute $\overline{S}_d(k)$ and $\sigma^2_{S_d}(k)$ for all k.

To account for the effect of the *seeds* (if any), we apply once more the normal approximation, as follows. Let $S(t, k)$ be the server bandwidth necessary at time t, assuming that there are k downloading users and $N_{\text{seed}}(t)$ seeds. Moreover, let $\overline{S}(t, k)$ and $\sigma^2_S(k, t)$ be the mean and variance of $S(t, k)$.

We observe that $S_{\text{seed}}(t)$ is a compound Poisson random variable, whose moments can be computed exactly in close-

form. In particular, the mean of $S_{\text{seed}}(t)$ is equal to $\overline{N}_{\text{seed}}(t)\overline{U}$, whereas its variance is equal to $\overline{N}_{\text{seed}}(t)(\sigma^2_U + \overline{U}^2)$. We approximate $S(t, k) - S_{\text{seed}}(t)$ by a normal distribution y of mean $\mu = \overline{S}(t, k) - \overline{N}_{\text{seed}}(t)\overline{U}$ and variance $\sigma^2 = \sigma^2_S(k, t) + \overline{N}_{\text{seed}}(t)(\sigma^2_U + \overline{U}^2)$, and apply again (6) and (7) to compute the first and second moment of $y' = \max\{0, y\} \simeq S(t, k)$.

Finally, the mean server bandwidth $\overline{S}(t)$ (and similarly its variance) can be obtained deconditioning with respect to k:

$$\overline{S}(t) = \sum_{k \geq 1} \overline{S}(t, k) \mathbb{P}(N_d(t) = k) \tag{10}$$

The entire computational procedure has numerical complexity $\Theta(k_{\max})$, where $k_{\max}$ is a suitable value such that $\mathbb{P}(N_d(t) > k_{\max}) < \epsilon$ is negligible (in our results we set $\epsilon = 10^{-6}$).

IV. Performance under stationary conditions

In this section we report a selection of the most interesting results that we have obtained by our analysis under stationary user arrival process. Since in this case all averages do not depend on t, we will omit for simplicity the indication of time. We normalize to 1 the video playback rate d_v, which thus serves as unit for all other bandwidth figures. We assume that users stay in the system for a time at least equal to the watching time, hence $\overline{T} \geq T_v$. Unless otherwise specified, we assume that users' upload bandwidth U is exponentially distributed. The results obtained by our analytical approximation in Section III-A are compared to those obtained by an event-driven ad-hoc simulator, which is used also to evaluate the simple lower bound:

$$\overline{S}(t) \geq \max\{0, d\,\overline{N}_d(t) - \overline{U}\,\overline{N}(t)\} \tag{11}$$

which can be derived assuming that, at any time, the upload bandwidth of the users can be entirely exploited by the system.
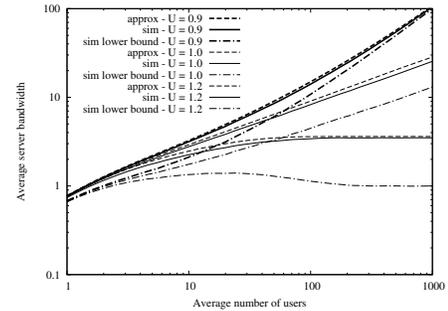

Fig. 1. Comparison of average server bandwidth in the case $d = d_v$, as function of the number of users $\overline{N}$, for different values of $\overline{U}$, in the absence of seeds.

A. Impact of the number of watching users and seeds

Figure 1 reports the average server bandwidth $\overline{S}$ as function of the average number of users $\overline{N}$, in the case $d = d_v$, $\overline{T} = T_v$. We consider three different values of average upload bandwidth $\overline{U} = 0.9, 1.0, 1.2$, corresponding to systems operating in deficit, balanced, and surplus mode, respectively (here $\gamma = 1/\overline{U}$).

Besides noticing the accuracy of the approximate analysis, it is interesting to see that the average server bandwidth saturates for $\overline{U} = 1.2$ (surplus mode) to a value about 3.5 times larger than the corresponding lower bound, which tends to
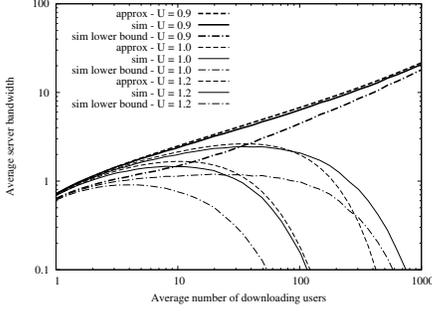
Fig. 2. Comparison of average server bandwidth in the case $d = d_v$, as function of the number of downloading users $\overline{N}_d$, for different values of $\overline{U}$, in the presence of $\overline{N}_{seed} = 0.1 \cdot \overline{N}_d$ seeds.
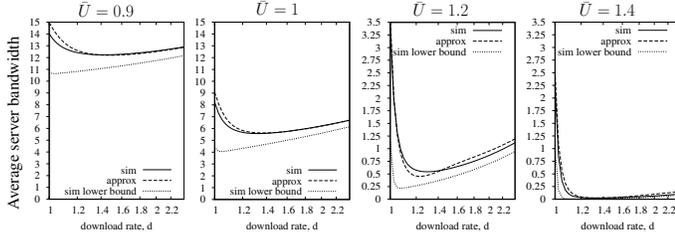

Fig. 3. Average server bandwidth as function of the download rate d, for different values of $\overline{U}$, with $\overline{N} = 100$, $\overline{T} = T_v$.

$d_v = 1$. As expected, the average server bandwidth diverges under the deficit and balanced modes. Moreover, in the deficit mode, the sequential system requires asymptotically the same bandwidth as the completely non-sequential system (*i.e.*, the lower bound).

In Figure 2 we compare the results obtained in the same system considered above, but assuming that users remain active after the end of the watching time for an exponentially distributed amount of time of mean equal to 10% of the watching time, generating an average number of seeds $\overline{N}_{seed} = 0.1 \cdot \overline{N}_d$. Now the systems with $\overline{U} = 1.0$ and $\overline{U} = 1.2$ operate in surplus mode, whereas the system with $\overline{U} = 0.9$ operates very close to the balanced mode (here $\gamma = 1/(1.1 \cdot \overline{U})$). We observe that, in the presence of seeds, the average server bandwidth requested by systems operating in surplus mode reaches a maximum, after which it goes to zero as the number of users increases. Results such as those reported in Figures 1 and 2 can be useful in system dimensioning, as they allow to estimate, in the surplus mode, the worst-case server bandwidth which is needed when the number of downloading users $\overline{N}_d$ is not known.

B. Impact of the download rate

Even if users tend to leave the system at the end of the watching time, it is still possible to benefit from the positive effect created by the seeds, who absorb part of the fluctuations in the bandwidth requested by downloading peers, shielding the servers. The trick to 'artificially' create seeds is to make the users download the video at rate $d > d_v$, so that they become seeds for other peers before the end of the watching time. Intuitively, however, d should not be set too large to offset the gain achievable by the seeds. Figure 3 illustrates the performance of this strategy in the case of $\overline{N} = 100$ users, $\overline{T} = T_v$, showing the average server bandwidth as function

of d. We observe that for all the considered values of $\overline{U}$, the average server bandwidth achieves a minimum for a value of d slightly larger than d_v. The impact is particularly striking in the surplus mode ($\overline{U} > 1$), in which setting $d > d_v$ brings the server bandwidth close to zero.

C. Impact of upload bandwidth heterogeneity

Figure 4 compares the average server bandwidth $\overline{S}$ as function of the variation coefficient of the peer upload bandwidth (keeping fixed the mean), considering $\overline{N} = 100$, $d = d_v$, $\overline{T} = T_v$. The average upload bandwidth is equal to either $\overline{U} = 0.9$ (deficit mode) or $\overline{U} = 1.1$ (surplus mode). Simulation results are supplemented by 95%-level confidence intervals. The upload bandwidth distribution used in the simulations depends on the variation coefficient: for values larger than one, we adopt a hyper-exponential distribution with balanced means; for values smaller than one, we employ an exponential distribution added to a constant.

We observe that the average server bandwidth increases significantly as the variability of upload bandwidth increases, while our approximation tends to provide a conservative prediction.
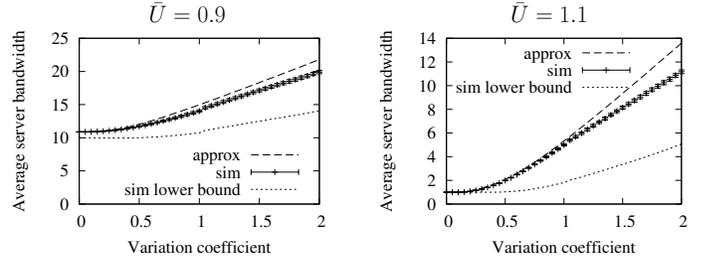

Fig. 4. Comparison of average server bandwidth with $\overline{N} = 100$, $d = d_v$, $\overline{T} = T_v$, as function of the variation coefficient of peer upload bandwidth, for $\overline{U} = 0.9$ (left plot) and for $\overline{U} = 1.1$ (right plot).

V. PERFORMANCE OF NON-STATIONARY SYSTEMS

In this section we show how our analytical framework can be applied to study the performance of time-varying systems in which the arrival rate of requests for a given content changes significantly over time. In particular, we will see that the behavior of a non-stationary system can dramatically differ from the one of a stationary system in which the arrival rate of requests is constant, due to a misalignment problem between the temporal evolution of the number of downloaders and the temporal evolution of the number of seeds.

A. The downloaders/seeds misalignment problem

We consider a reference scenario in which the arrival rate of requests for a given video follows a daily pattern which is modeled for simplicity by a sine function of period equal to 24 hours, between a minimum of $\lambda = 0.1$ and a maximum of $\lambda = 1$, represented by the thick solid line in the top plot of Fig. 5.

We first analyze a *software-based* system in which users contribute their upload bandwidth during the watching time of the video, plus a random additional time in which the application is kept running. We assume that the video duration is $T_v = 2$ h, and the additional activity time after the end of the video is exponentially distributed with mean 1 h. We normalize

$d_v = d = 1$ and assume that the upload bandwidth of users is exponentially distributed with mean $\overline{U} = 0.7$. The per-user load is $\gamma_p = 2/(0.7 \cdot 3) \approx 0.95$. The top plot of Fig. 5 reports the temporal evolution of both the number of downloaders and the number of seeds. Since peers become seeds only after the end of the watching time, the dynamics of downloaders and seeds are misaligned, with a temporal shift about $T_v = 2$ h.
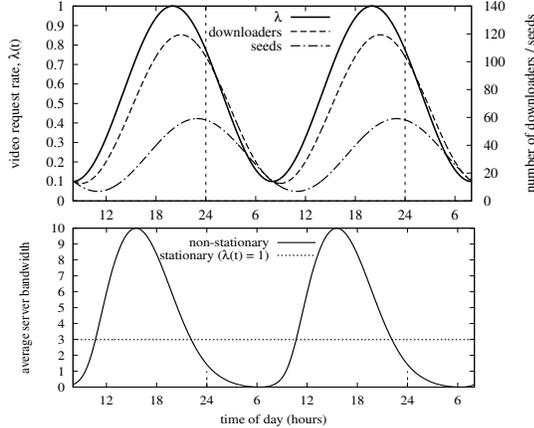

Fig. 5. Temporal evolution of video request rate (top plot, left y axes), number of downloaders/seeds (top plot, right y axes), and average server bandwidth $\overline{S}(t)$ (bottom plot), in the *software-based* system.

The effect of this misalignment on the required average server bandwidth is depicted on the bottom plot of Fig. 5, by the solid line labeled 'non-stationary', which exhibits a peak preceding the point at which the video request rate is maximum. Fig. 5 reports also a curve labeled 'stationary', representing the server bandwidth that would be necessary if the content request rate were constant and equal to $\lambda = 1$ (the maximum request rate). We observe that the performance of the non-stationary system is worse than that of the stationary system, both in terms of peak server bandwidth and average server load. This occurs even if the content request rate is always larger in the stationary system.

If we increase the activity time after the end of the video, while keeping the same per-user system load γ_p (either by reducing the upload bandwidth of the users, or equivalently by increasing the download rate of the video, *i.e.*, its resolution/quality), the negative effect of the misalignment problem become worse. As an extreme case, we consider a P2P-VoD system relying on *set-top-boxes* which are always active and serving the last watched video. To mimic the behavior of set-top-boxes with our model of peer dynamics, we assume that the activity time after watching a movie is much longer than before (in the order of a day), representing a set-top-box which remains always on before the user downloads the next video. In particular, we consider an additional activity time of 22 h, which added to the watching time of a movie leads to $\overline{T} = 1$ day. To obtain the same per-user load of the *software-based* system, we set $\overline{U} = 0.7 \cdot 3/24$.

Fig. 6 reports analytical results for this scenario, analogous to those in Figure 5. We have also reported on the bottom plot of Fig. 6 a sample path obtained from simulation, to confirm the analytical prediction. In this case the instantaneous system load $\gamma(t)$ is severely unbalanced across the day. During peak hours, the bandwidth requested at servers grows very large,
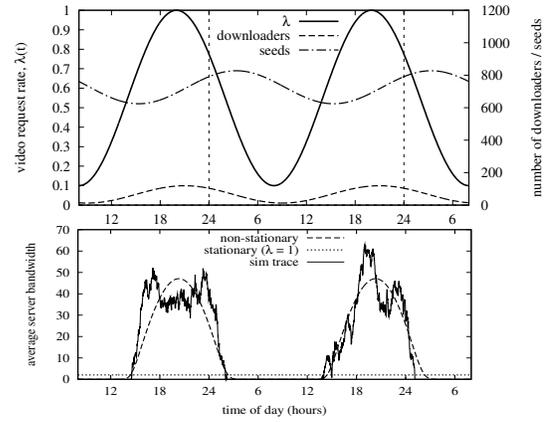

Fig. 6. Temporal evolution of video request rate (top plot, left y axes) number of downloaders/seeds (top plot, right y axes), and average server bandwidth $\overline{S}(t)$ (bottom plot), in the *set-top-box* system.

while for the rest of the day it is negligible. The problem is that the upload capacity of the seeds, which are very numerous and almost stable along the day (see top plot of Figure 6) is totally wasted for a large fraction of the day. Notice that we are not saying that *set-top-boxes* are not useful: increasing the activity time of peers (up to the point of having always-on user devices) is very beneficial to the system performance, since the per-user load γ_p is reduced. However, one must be careful that the instantaneous system load $\gamma(t)$ can vary significantly around γ_p, and peak traffic demand cannot be absorbed well by large populations of seeds (*set-top-boxes*) each devoting a small amount of upload bandwidth to the video distribution. Indeed, to minimize the bandwidth deficit at peak times, the ratio $\overline{U}/d$ should not become too small. In [7] we show how our model can be applied to design resource-allocation strategies (under server capacity constraints) to address the above misalignment problem.

VI. CONCLUSIONS

We have proposed a computationally-efficient methodology to analytically estimate the server bandwidth requested in non-stationary peer-assisted VoD systems. Our approach is highly flexible, and can account for several important effects such as peer upload bandwidth heterogeneity and churning. By applying our performance evaluation methodology under various parameters setting, we have discovered several interesting properties of P2P-VoD systems.

REFERENCES

[1] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, "On next-generation telco-managed P2P TV architectures," in *IPTPS*, 2008.
[2] R. Kumar, Y. Liu, and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," in *IEEE INFOCOM*, 2007.
[3] Y. H. et al., "Challenges, Design and Analysis of a Large-scale P2P VoD System," in *ACM SIGCOMM*, 2008.
[4] W. Wu, J. Lui, "Exploring the Optimal Replication Strategy in P2P-VoD Systems: Characterization and Evaluation," in *IEEE INFOCOM*, 2011.
[5] D. Ciullo, V. Martina, M. Garetto, E. Leonardi, and G. L. Torrisi, "Stochastic Analysis of Self-Sustainability in Peer-Assisted VoD Systems," in *IEEE INFOCOM*, 2012.
[6] C. Huang, J. Li, and K. W. Ross, "Can Internet Video-on-Demand Be Profitable?" in *ACM SIGCOMM*, 2007.
[7] D. Ciullo, V. Martina, M. Garetto, E. Leonardi, and G. L. Torrisi, "Performance Analysis of Non-stationary Peer-assisted VoD Systems," technical report, 2012, available at http://www.telematica.polito.it/~emilio/papers/Infocom12-2.pdf.