POLITECNICO DI TORINO
Repository ISTITUZIONALE

Modeling and Evaluation of Multisource Streaming Strategies in P2P VoD Systems

(Article begins on next page)

01 September 2025

Modeling and Evaluation of Multisource Streaming Strategies in P2P VoD Systems

Juan Pedro Muñoz-Gea, Stefano Traverso, and Emilio Leonardi, *Senior Member*, IEEE

Abstract — *In recent years, multimedia content distribution has largely been moved to the Internet, inducing broadcasters, operators and service providers to upgrade with large expenses their infrastructures. In this context, streaming solutions that rely on user devices such as set-top boxes (STBs) to offload dedicated streaming servers are particularly appropriate. In these systems, contents are usually replicated and scattered over the network established by STBs placed at users' home, and the video-on-demand (VoD) service is provisioned through streaming sessions established among neighboring STBs following a Peer-to-Peer fashion. Up to now the majority of research works have focused on the design and optimization of content replicas mechanisms to minimize server costs. The optimization of replicas mechanisms has been typically performed either considering very crude system performance indicators or analyzing asymptotic behavior. In this work, instead, we propose an analytical model that complements previous works providing fairly accurate predictions of system performance (i.e., blocking probability). Our model turns out to be a highly scalable, flexible, and extensible tool that may be helpful both for designers and developers to efficiently predict the effect of system design choices in large scale STB-VoD systems[1].*

Index Terms — **Peer-to-Peer, Video-on-Demand Streaming, Set-Top Box, Queueing Model, Performance Evaluation.**

I. INTRODUCTION

In recent years, multimedia content (especially video) distribution has largely been moved to the Internet. This calls for tremendous backend streaming resources with important processing power and, most critically, huge bandwidth capacities. Traditional VoD streaming systems that rely on point-to-point streaming and the client-server paradigm embody all these scalability and cost issues. Peer-to-Peer (P2P) technology is often used to either off-load content servers, or to create completely decentralized content sharing and distribution communities. In both cases, peers share their available resources, such as their up-link channel and some limited cache capacity to cooperatively serve one another.

Service providers can take advantage of the cache capacity available at peers (in both memory and non-volatile storage) to pro-actively place certain contents on them. The hope is that such contents may become useful to other peers, thus mitigating the load on servers. On this matter, past relevant research works have analyzed such P2P-based VoD streaming architectures that use the storage capacities of end-users' set-top boxes (STBs) to deliver scalable VoD streaming services [1]. Nowadays, different kinds of devices can be used as STBs. The most conventional ones are receivers for dedicated channels, such as cable networks or terrestrial broadcasts. Simple functionalities can be even provided by home gateways, such as DSL or cable modems with a hard drive and extended features. Different types of STBs are already used for various IPTV and VoD platforms such as Tivo, Apple TV and Roku player[2].

P2P-based streaming solutions based on the pro-active placement of content on STBs consist of fragmenting the different video contents into complementary sub-streams and then spreading them over subsets of STBs. By this way a VoD service request generated by a STB can be served by other contributing peers[3] in the network, provided that they have the necessary sub-streams and up-link capacity to do so.

Up to now, to best of our knowledge, most research works focused on the design of strategies for replicating and distributing movies over the network established by STBs to minimize costs at server side. Regarding to this issue, several recent works [2], [3] have analytically demonstrated that the optimal number of replicas of each movie must be proportional to its relative popularity within the content library. Some other works [4], [5] have reported that the proportional replication may have poor performance for unpopular movies, and for this reason the replication strategy should be greedier for this kind of contents. On the other hand, due to the evolution of diverse users' interest it is necessary that the number of replicas of videos are pro-actively adapted to the video popularity. By this reason, several works have proposed different algorithms to control replication ratios [2], [3], [5], [6]. We would like to emphasize, however, that the optimization of replicas mechanisms has been typically performed either considering very crude system performance indicators or analyzing asymptotic behavior.

Independently of the selected strategy for replication and distribution of the contents, there is the need of a tool to efficiently predict performance of large size peer-assisted

[2] http://www.tivo.com/, http://www.apple.com/appletv/, http://www.roku.com

[3] In the remainder of this paper, the terms "STB", "peer" and "node" will be used interchangeably

VoD services, modeling all the relevant aspects that may have an impact on system performance (user bandwidth heterogeneity, possible user unavailability, adoption scalable encoding schemes, etc.) typical of a STB-based network [7], [8]. In this paper we try to fill this gap proposing a simple, yet easily extensible model, which can be successfully employed by developers to validate/orient their design choices.

The contribution of this work is then two-fold. First, we construct a new simple and scalable analytical queuing theoretical model, which is able to represent/capture the fundamental dynamics of STB-based systems. Second, we show how our model can be easily extended to evaluate dynamic, heterogeneous and hybrid scenarios. The model with all its extensions is validated by comparing its performance results against those obtained from a simulated environment. The good fitting with simulation results obtained in our validation proves that the model we propose is a reliable and flexible tool to design and dimension a STB-based VoD system.

The rest of the paper is structured as follows. Section II reviews the most relevant research works. Section III provides an in-depth description of our target VoD architecture. Section IV offers a thorough description of our analytical model and its extensions. Section V presents different multisource streaming strategies. In Section VI the reliability of our model is evaluated. Finally, Section VII draws important conclusions from this work.

II. RELATED WORKS

Nowadays, multimedia content distribution over the Internet deeply relies on data centers and content delivery networks which come with big drawbacks such as deployment and management complexity, power consumption, and lack of scalability. Several works have been proposed to replace these technologies with P2P networks built by STBs placed at users' home, i.e. at the edge of the Internet. In particular for VoD streaming, many proposals can be found in the literature. In the following, we resume the current state of the art.

Toast [9] was the first work in which a peer-assisted VoD architecture based on STBs was developed. This system modified the BitTorrent protocol to enable STBs to stream data to each other, while still using VoD servers to provide content chunks when no peer could provide requested data in a timely fashion. To improve the efficiency of the system, authors of [10] proposed to add a pre-fetching module to the BitTorrent-like protocol used in the STB-based scenario.

To further improve the quality of service of these systems, many works introduced a pro-active content placement on idle STBs, thus letting the resulting system exploit already dedicated baseline resources, i.e. contents distributed over the STBs network, to satisfy current and future user demand. Suh et al. [11] proposed the first push-to-peer VoD system where data is pro-actively pushed from the content provider to peers. Some striping-based policy was introduced, but no optimization of content placement was attempted. Performance analysis of both queuing and loss models were

considered.

More recent works have better focused on the problem of pro-actively distribute content replicas over the P2P network established by the STBs. In [2] authors optimize caching under a stochastic supply-demand model showing that the optimal cache profile is proportional to the popularity of the contents. Similarly, in [3] authors consider a loss network model of performance, and determine asymptotically optimal content placement strategies in the case of a limited content catalogue, and then they turn to an alternative "large catalogue" scaling, where the catalogue size scales with the peer population. However, although previous works and conventional wisdom suggest using the proportional-to-popularity replication strategy, authors in [4] reported this policy may lead to poor performance, especially for unpopular movies.

On the other hand, due to the evolution of diverse users' interests across a rich repository of videos, it is not uncommon to have demand-supply imbalance across videos. This requires the number of replicas of videos to be pro-actively adapted to the video popularity in order to guarantee good service quality. Several works have proposed different algorithms to control replication ratios. These algorithms are characterized by two approaches: *passive adjustment*, in which peers adjust the replication ratio in the P2P-VoD system using replacement algorithms [3], [6], and *active adjustment*, where the server actively uploads data for which the replicas are not at the desired replication ratio [2], [5].

The model proposed in this paper complements the works described above, aiming at predicting the effects of design choices on system performance expressed through blocking probability.

III. SYSTEM ARCHITECTURE

Many possible architectures have been proposed for supporting STB-based P2P VoD applications. For each architecture different algorithms for content replication, distribution, and seeking of video contents have been defined. Here, we describe a simple broad P2P VoD solution which could be adapted to most STB-based architectures.

We consider a system involving B STBs connected to each other creating an actual P2P network $\mathcal{B}$ working at application layer. Each STB is expected *i)* to be placed at the edge of the network, i.e. at user's home, possibly behind an xDSL or an FTTH interface, *ii)* to be equipped with some limited storage device equivalent to a few GBytes capacity, and *iii)* to be always on and reachable.

Available contents are listed in a catalogue $\mathcal{C}$. Users are free to browse the catalogue and choose movies to stream. Each new title c, when uploaded onto the system, is replicated X_c times over the network created by STBs, i.e. $\mathcal{X}_c$ (with $X_c = |\mathcal{X}_c|$) represents the set of chosen STBs which will host a copy of c. We assume that the popularity of each title λ_c is somehow a priori known, so that X_c is proportional to c's expected demand. Given the small storage capacity of each STB, the design of the scheme that drives replicas placement

represents a difficult task that have been deeply considered in literature [12]. However, the placement criteria for the replicas of title c onto $\mathcal{B}$ is orthogonal to the video delivery problem, and it is not considered further.

The system relies on a tracker server T_S, which is responsible of tracking availability of contents in the P2P network by keeping an up-to-date picture of the movies hosted by each STB. Whenever a user u desires to consume a video, T_S provides her STB, S_u, the set of available "active" STBs, $\mathcal{X}_c$, which are hosting the required title. Then S_u arbitrarily selects one STB, S_u^t, out of $\mathcal{X}_c$ to download the content. S_u^t can satisfy the request only if it has sufficient available upload bandwidth to transmit the video at its declared encoding rate. If this latter requirement is satisfied, S_u^t starts delivering the video to S_u. Otherwise, S_u moves to next STB listed in $\mathcal{X}_c$ and asks again for the desired content. This procedure is repeated until the request generated by S_u is successfully satisfied and user can consume the video.

It can happen that in overloaded conditions, peers are not able to satisfy new incoming requests, inducing requesting peer S_u to wait for large amount of time before finding some peer available for the upload. Once S_u has expired the number of possible request attempts X_c, it moves to the backup server, B_S, to finally get the video. B_S is indeed expected to have limitless resources and infinite upload capacity, thus being able to deliver whatever title indexed in $\mathcal{C}$ and demanded by new incoming requests. However, resorting to B_S has to be seen as the last possible alternative to the system, being the P2P network established by the STBs expected to fully handle the amount incoming requests independently.

Observe that this design can be easily extended to consider the case in which movies are split in complementary sub-streams.

IV. MODEL DESCRIPTION

In this Section we present our analytical model. For simplicity, we start considering a simple system in which files are not split into sub-streams and every content download request is served by only one peer. We will show in Section IV-E how this assumption can be easily released.

A. Preliminary System Assumptions

We start considering a purely P2P system consisting of a network of B STBs (or reliable peers, i.e. where no churning happens), in which bandwidth is provided only by STBs, then we will show how our model can be easily adapted to model hybrid systems in which video distribution is supported also by servers that provide the extra bandwidth needed to guarantee optimal QoS to the users. In the following we present the main assumptions our model relies on:

1) we assume all STBs having a local cache that can store up to Z contents. Therefore, the total storage capacity provided from the pool of nodes is equal to $B \cdot Z$.
2) We initially target a homogeneous scenario in which all STBs are provided with the same upload capacity U.

Then, in Sec. IV-G we will show how our model can be generalized to a heterogeneous scenario.

3) We expect all video contents to be encoded with similar quality, i.e. the nominal video rate for all contents can be assumed to be the same, equal to d bit/s. As described in Sec. IV-G also this assumption can be partially released.
4) Furthermore, we assume that the duration of contents follows a general distribution with average $E[L]$.
5) Contents available in the catalogue for streaming follow a Zipf popularity distribution.

B. Content Popularity

C different contents are offered by the VoD service to its users, every content c (for $1 \le c \le C$) has an associated popularity λ_c, which represents the aggregate rate at which users request content c. The numerical results presented in this paper refer to a scenario in which contents are classified into a limited number of I classes, $\mathcal{C}_i$, according to their relative popularity. All contents in category $\mathcal{C}_i$ have the same associated popularity λ_i. Furthermore, we assume that the popularity associated to contents in different categories is distributed according to a Zipf with parameter α:

$$\lambda_c = \Lambda \frac{\frac{1}{i^\alpha}}{\sum_{j=1}^{I} \frac{1}{j^\alpha}} \qquad \forall c \in \mathcal{C}_i \qquad (1)$$

We remark that the Zipf distribution is widely accepted as a realistic model of video content popularity in Internet as shown in [13]. However, even if our model will be described and evaluated considering a Zipf-based popularity distribution, it is going to work also under general content popularity distributions.

C. Basic Model

We assume that content c is replicated X_c times in the system (i.e. it is stored in X_c different STBs). Thus, whenever a request for the specific content c is issued by a user, the system can redirect it to any of the X_c STBs storing such content. Streaming immediately starts from one of the STBs storing the desired content, provided that enough bandwidth is available at the selected STB to sustain a minimal transfer rate d equal to the nominal video rate. If servers are involved to support the content distribution, requests that cannot be satisfied by STBs pool are redirected to one of the B_S servers (see Sec. IV-H), otherwise they are blocked.

We model every individual STB storing the tagged content c as a simple queue. Two different queuing systems well adapt to the two different situations in which: i) the system is designed in such a way to maximize the instantaneous rate of concurrent downloads, so to exploit at the best the upload bandwidth of STBs (such as in push based systems), or ii) when, instead, the system is designed to sustain a download rate matched to the nominal video rate in any bandwidth situation (such as in pull based systems).

In the first case we can model every STB as a single server, limited processor sharing queue (M/G/1/K Processor Sharing). The limit of the processor sharing queue is set in such a way to

ensure that the system can always guarantee a minimum download rate equal to the nominal video rate to every ongoing download, i.e. $K = \lfloor U/d \rfloor$. In the second case we model the STB by means of a multi-server queue with no waiting line (M/G/K/K), where again $K = \lfloor U/d \rfloor$. Observe that, in both cases we make the assumption that the arrival rate at STBs follows a homogeneous Poisson process. This assumption, which is not completely verified in practice, constitutes one of the main sources of approximation of our model.

Considering the b-th STB that is hosting content c, we denote with λ_b the average request rate hitting b and with $\overline{L_b}$ the average duration of contents stored in b. We then define the load for each STB with $\rho_b = \lambda_b \overline{L_b}$, so that we can easily evaluate $P_o(\rho_b)$, the probability that a content cannot be served by a STB for both queuing models described above:

$$P_o(\rho_b) = \begin{cases} \dfrac{\frac{\rho_b^K}{K!}}{\sum_{j=0}^{K}\frac{\rho_b^j}{j!}} & \text{M/G/K/K} \\[4mm] \dfrac{\rho_b^K(1-\rho_b)}{1-\rho_b^{K+1}} & \text{M/G/1/K PS} \end{cases} \qquad (2)$$

Now, assuming the behavior of individual STBs to be independent, we can derive an estimate for the probability that an incoming download request for a content c cannot be served by any of the STBs storing the desired content as:

$$P_{block}(c) = \prod_{b \in \mathcal{B}_c} P_o(\rho_b) \qquad (3)$$

where $\mathcal{B}_c$ denotes the set of STBs storing content c.

Observe that, in general, both λ_b and $\overline{L_b}$ come to depend on the considered content displacement policy. To simplify our model and to obtain results that depend just on the replication factors X_c and not from the particular displacement, in our model we set $\lambda_b = \overline{\lambda_{b|c}} = E[\lambda_b | c \in b]$ and $\overline{L_b} = \overline{L_{b|c}} = E[L_b | c \in b]$ for all the STBs hosting content c, where $\overline{\lambda_{b|c}}$ denotes the average arrival rate of content requests to a generic STB, given that this STB is hosting a content c. $\overline{L_{b|c}}$ denotes the average length of contents stored in each STB, given that c is one of the stored contents. So doing, we obtain an expression for $P_o(\rho_b)$, which depends only on the stored content c (we denote this expression with $\overline{P}_o(\rho_b|c)$). As a consequence, the expression (3) reduces to:

$$P_{block}(c) = \left(\overline{P}_o(\rho_b|c)\right)^{X_c} \qquad (4)$$

where X_c is the cardinality of the set of STBs storing the desired content c returned to users by tracker T_S.

We emphasize that the evaluation of $\overline{\lambda_{b|c}}$ is not immediate. It can be obtained by an **Erlang Fixed Point Approximation** described in details in our technical report [14].

D. Evaluation of $\overline{\lambda_{b|c}}$

We can approximately evaluate $\overline{\lambda_{b|c}}$ as:

$$\overline{\lambda_{b|c}} = \frac{\lambda_c}{X_c} + \frac{B-1}{BZ} \cdot \frac{\Lambda - \lambda_c}{C} + \frac{1}{Z}\sum_{c=1}^{C}\lambda_c \sum_{n=1}^{X_c}\lambda_c\left(\overline{P}_o(\rho_b|c)\right)^n$$

where $\frac{\lambda_c}{X_c}$ represents the arrival rate to b of fresh requests for content c (i.e., requests for which box b is the first box on which they are routed to), $\frac{B-1}{BZ} \cdot \frac{\Lambda - \lambda_c}{C}$ represents the average rate of fresh requests for the other contents stored in b, and the third term represents the average rate of redirected requests (i.e. requests that have been previously blocked at some STB). Observe that we have assumed the rate of requests to be essentially equally distributed among all the boxes in the system. Now observe that $\overline{P}_o(\rho_b|c)$, depends on the average arrival rate of requests at boxes which can be given in terms of quantities $\overline{\lambda_{b|c}}$ as $\frac{1}{Z}\sum_c \overline{\lambda_{b|c}}$. A standard fixed point numerical procedure is used to solve the resulting chicken and egg loop.

E. Extension to the Multi-Stripes Case

The simple system described in previous section is able to exploit STBs upload bandwidth only under the condition $U \geq d$. To increase the efficiency of the system, the video content can be partitioned in H sub-streams, and different sub-streams can be retrieved by different STBs. Each sub-stream contains a portion of the video stream and must be downloaded at rate $d_s = d/H$. An easy way to generate H different sub-streams from a video file is to cut it into small fragments of fixed size (chunks) and to assign to sub-stream h those fragments whose order n satisfies $|n|_H = h$[4].

Every sub-stream of a video c is replicated X_c times and stored by disjoint sets of X_c STBs. When a new download request for content c arrives, for each sub-stream, one STB, hosting the considered sub-stream is selected as server if enough bandwidth is available.

Also in this case we can model the individual STB dynamics with an M/G/1/K Processor Sharing model or an M/G/K/K model, having placed $K = \lfloor HU/d \rfloor$. Note that in this case the hitting rate of requests at STB represents the sub-streams request rate hitting STB b. Thus, again from (2) we obtain the expression of the probability that a STB hosting a stream of the desired content has not enough available bandwidth to sustain the new download.

Content download can successfully start only when all the sub-streams can be successfully downloaded by STBs. As a consequence, the probability that the new incoming is blocked is given by:

$$P_{block}(c) = 1 - \left(1 - \left(\overline{P}_o(\rho_b|c)\right)^{X_c}\right)^{H} \qquad (5)$$

F. Extension to the Case of Non Reliable Peers

When peers are not fully reliable, it may happen that some of the peers hosting a copy of content c are not available, i.e. not "on" or reachable, inducing the system to resemble a dynamic P2P network. Thus, the number Y_c of available peers

---

[4] Symbol $|.|_M$ denotes the module M operator.

hosting content c at time t becomes a random variable. Under the assumption that the behavior of different peers is independent, and that every peer is available with a probability p_{av}, Y_c is a binomial distributed random variable.

As a consequence, in the basic case in which no multi-striping is allowed the expression (3) reduces to:

$$P_{block}(c) = \sum_{y=0}^{X_c} \binom{X_c}{y} \left(\overline{P_o}(\rho_b|c)\right)^y p_{av}{}^y (1-p_{av})^{X_c-y} \qquad (6)$$

The extension to the case in which multi-striping is allowed can be done along the same lines.

G. Extension to Heterogeneous Cases

In this section we show how our model can be extended to the case in which either the upload bandwidths at STBs or the encoding rate of videos are heterogeneous. We first consider the case in which M different classes of STBs have different upload bandwidth. Let U_m represent the upload bandwidth of class m. In this case the probability that a content request cannot be served by a STB comes to depend on the class m:

$$P_o(\rho_b|m) = \begin{cases} \dfrac{\frac{\rho_b^{K_m}}{K_m!}}{\sum_{j=0}^{K_m}\frac{\rho_b^j}{j!}} & \text{M/G/K/K} \\[4mm] \dfrac{\rho_b^{K_m}(1-\rho_b)}{1-\rho_b^{K_m+1}} & \text{M/G/1/K PS} \end{cases} \qquad (7)$$

where $K_m = \lfloor U_m/d \rfloor$.

Then, an expression of $P_{block}(c)$ can be obtained conditionally over the vector $\boldsymbol{r} = [r_c^1, \dots, r_c^M]$, whose m-th component is the number STBs belonging to class m returned by the tracker T_S:

$$P_{block}(c|r) = \prod_{m=1}^{M} \left(\overline{P_o}(\rho_b|c,m)\right)^{r_c^m} \qquad (8)$$

At last the expression of $P_{block}(c)$ is obtained unconditioning over $\boldsymbol{r}$.

To extend, instead, our model to the case in which files with different download rates coexist we have to resort to multi-class extensions of the simple queuing expressions in (2). Multi-class generalizations of both the M/G/K/K PS and M/G/1/K systems are available in literature under the constraint that service times are phase-type distributed [15]. We can exploit those results to obtain an expression for $P_o(\rho_b)$, from which proceeding the same lines as before, we can evaluate the blocking probability.

H. Hybrid Systems

In hybrid systems, a set of distribution servers $\mathcal{B}_S$, expected to be equipped with infinite resources, supports the process of delivery of contents. Requests that cannot be served by STBs are redirected to a server $B_S \in \mathcal{B}_S$. In such a case, it is important to characterize the bandwidth requested to servers $\mathcal{B}_S$, which can be evaluated in the basic scenario as:

$$B_{\mathcal{B}_S} = \sum_{c=1}^{C} \lambda_c d P_{block}(c) \qquad (9)$$

where we recall that λ_c represents content c aggregate request rate and $P_{block}(c)$ is computed according to (4). Previous expression can be easily extended to the multi-striping case, evaluating the average rate at which striping requests cannot be served by STBs. It results:

$$B_{\mathcal{B}_S} = H \sum_{c=1}^{C} \lambda_c d_s \overline{P_{o,H}}(\rho_b|c)^{X_c} \qquad (10)$$

where $\overline{P_{o,H}}(\rho_b|c)$ represents the probability that a stripe of content c cannot be served by STBs.

V. Multi-Striping Strategies – Advanced Techniques

The simple multi-striping technique described in the previous section can be made more effective by employing advanced coding techniques such as Layered Coding (LC) and Multiple Description Coding (MDC).

A. Layered Coding

The latest layered coding standard, known as SVC [16], relies on a coding scheme that splits the video-stream in a base-layer and successively refinable enhancement layers. The base-layer only is needed to be delivered successfully to guarantee a basic quality decoding of the video, while receiving enhancement layers will further increase the quality, but with larger available bandwidth requirements. The main drawback of this family of schemes is that enhancement layers cannot be decoded when the base layer is missing. This means that chunks carrying base layer data must be transmitted with higher priority. In the context of P2P VoD, SVC strategy issues the problem that streaming rates and storage requirements cannot be the same for the different stripes of one specific video-content.

B. Multiple Description Coding

Also MDC (Multiple Description Coding) schemes split the video-stream in multiple layers (aka descriptions), but differently from SVC, all descriptions carry the same amount of information and the quality of decoded video increases with the number of successfully received layers. In this way there is no base-layer to decode the video, and whatever alone received layer is enough to reproduce a basic quality video-stream. The ideas behind this promising technology are to provide error resilience to media streams while enabling some automatic rate-adaptive streaming.

C. Model Extension

Our model can be, in principle, easily extended to analyzing the performance of system employing both LC and MDC. For example, when a MDC coding scheme is employed, our model easily provides the probability that h stripes of content c (with $0 \le h \le H$) are retrieved by users:

$$P_h(c) = \binom{H}{h} \left(\overline{P_{o,H}}(\rho_b|c)^{X_c}\right)^h \left(1 - \left(\overline{P_{o,H}}(\rho_b|c)^{X_c}\right)\right)^{H-h} \qquad (11)$$

Observe that when MDC technique is employed, the quality of experience perceived by users comes to depend on the number of stripes h received.

VI. PERFORMANCE EVALUATION

In this section we evaluate the actual reliability of our model, starting from the basic, presented in Sec. IV-C, to all its extensions, presented in Sec. IV-E to IV-H.

TABLE I
RESUME OF STB PARAMETERS ADOPTED IN G_{FTTH} AND G_{ADSL} SCENARIOS

	G_{FTTH}	G_{ADSL}
U	10 Mb/s	1.0 Mb/s
SC	40 GB	4 GB
d	3 Mb/s	0.5 Mb/s
L	2 GB	0.35 GB
Z	20	12
K	3	2

A. Model Validation

In order to evaluate the goodness of the proposed model, we considered two different scenarios. In the first one, called G_{FTTH}, the VoD service relies on a FTTH (Fiber-To-The-Home) network, where STBs take advantage of fiber links with symmetric download and upload bandwidth capacities. Even if not diffused in all countries, fiber network services start being proposed by many Internet Service Providers, and fiber connections are often proposed coupled with VoD streaming services [17]. The second scenario we adopt, called G_{ADSL}, considers instead a more common setup, in which each STB relies on an ADSL interface with poor upload capacity. The two scenarios also differ for the storage capacity available at each STB, which is expected to be proportional to the up-link available bandwidth. In both scenarios, 2k active STBs are involved and the content library is composed by 400 titles. Each title has been added to a given class associated to a certain popularity index as shown in Sec. IV-B. Each content is therefore replicated over the STBs network a number of times proportional to the popularity index of the category it belongs to. In particular, we adopted a Zipf popularity function with $\alpha = 0.25$ and 10 different content classes.

G_{FTTH} - Each STB is equipped with a storage capacity (SC) of 40GB and down-link/up-link capacities equal to 10Mbit/s. Given its large available resources, the system is able to support high definition streams with an average encoding rate of 3Mbps. Given an average length of 90min, to every title in the content library corresponds a storage footprint (L) of 2GB, therefore each STB can host 20 content replicas. Considering these system parameters, every STB can contribute to the system streaming up to 3 movies at the same time.

G_{ADSL} - In this case the overall system capacity is limited due to the low up-link speed that constitutes the main bottleneck. Indeed, each STB has 1Mbit/s and 5Mbit/s upload and download capacities respectively. The storage capacity at each STB is limited to 4GB. Given a 90min average length of

movies and an encoding rate of 0.5Mbps, a stream has a storage footprint roughly corresponding to 0.35GB. In this case, each STB contributes to the streaming service transmitting up to 2 movies in parallel and can store up to 12 copies.

G_{FTTH} and G_{ADSL} represent two benchmark scenarios that we selected for the validation of proposed model. In Tab. VI-A we summarize the parameter settings for each STB in the two described scenarios.
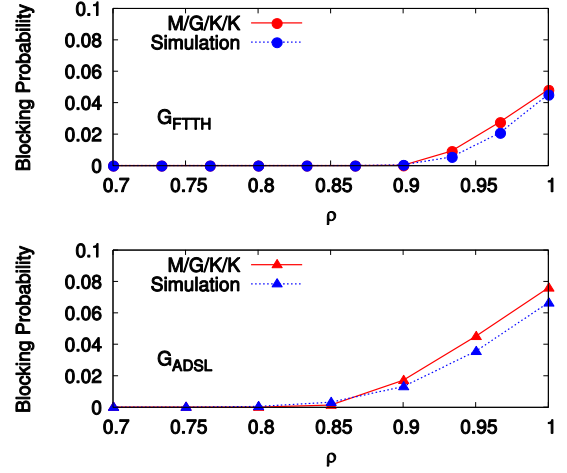

Fig. 1. Comparison of the blocking probability between analytical model and simulation when increasing the system load for G_{FTTH} (top) and G_{ADSL} (bottom).
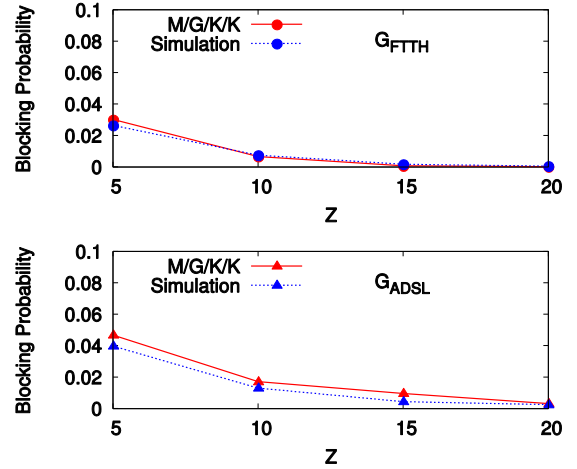

Fig. 2. Comparison of the blocking probability between analytical model and simulation when increasing the storage capacity of each STB, for G_{FTTH} (top) and G_{ADSL} (bottom). System load $\rho = 0.8$.

To enforce the validation process, we employed a custom discrete event-driven simulator that has been developed in C language to analyze the behavior of a peer-assisted streaming system.

B. Basic Model Evaluation

We start our evaluation comparing the M/G/K/K model represented in Eq. (2) with simulation results obtained through our custom simulator. Fig. 1 reports the blocking probability (averaged over all the 10 content classes) when increasing the

system load which has been defined as $\rho = \lambda \bar{L}$. Focusing on the top plot that refers to G_{FTTH} scenario, observe how the curve representing the model perfectly fits the simulation result. The bottom plot instead refers to G_{ADSL} scenario, in which STBs are equipped with more limited resources. Even if in this case the two curves do not collimate, the analytical model shows the same trend of the simulation curve, but with a small pessimistic estimate, making it possible, however, to successfully dimension a P2P VoD system. All simulation results reported in this paper present on average a 95% confidence interval with error equal to 22%. For the sake of clearness, plots just report average values, so that the good fitting between analytical and simulation curves is better highlighted.
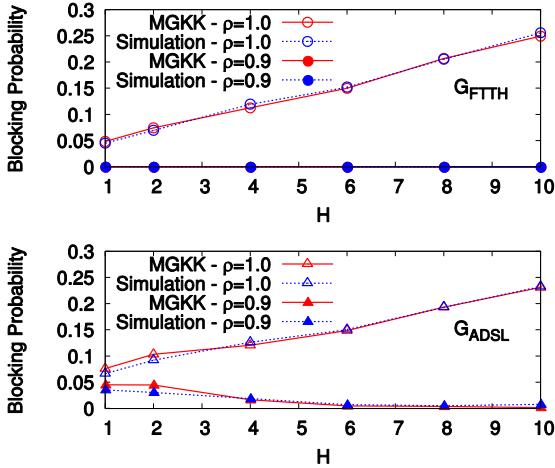

Fig. 3. Comparison of the blocking probability between analytical model M/G/K/K and simulation results when increasing the number of stripes H. System loads $\rho = 0.9, 1.0$ are considered for both G_{FTTH} (top) and G_{ADSL} (bottom) scenarios.
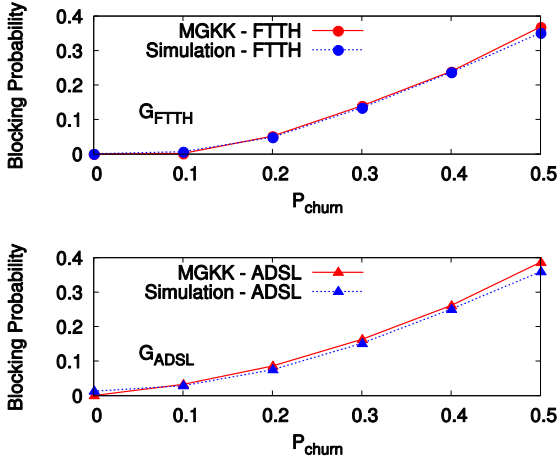

Fig. 4. Comparison of the blocking probability between analytical model M/G/K/K and simulation results when increasing the churning probability in the system P_{churn}. System load $\rho = 0.9$ has been considered for both G_{FTTH} (top) and G_{ADSL} (bottom) scenarios.

As an example of system dimensioning procedure, we analyse the impact on the blocking probability of parameter Z which represents the storage capacity available at each STB. It

is known, indeed, that increasing the size of hard disks represents a little expense, but this can lead to reasonable benefits in terms of performance. Fig. 2 reports the average blocking probability averaged over the 10 considered content classes for a fixed system load, $\rho = 0.8$. The plot confirms what expected, i.e. increasing the number of content replicas that each STB can store, we increase for each request the probability of retrieving and downloading desired content. This plot repeats what seen in Fig. 1: G_{FTTH} shows very good fitting between model M/G/K/K and simulation, while in G_{ADSL} case our model is good enough to provide a constant pessimistic estimate of the simulated system behavior.

C. Multi-Striping Extension Evaluation

In this subsection we evaluate the extension to our model that is evaluating multi-striping technique, as explained in Sec. IV-E. Fig. 3 depicts the aggregate blocking probability per content when increasing the number of stripes H in which contents have been splitted in. Top and bottom plots refer to G_{FTTH} and G_{ADSL} scenarios respectively. In order to provide a clearer view about benefits and drawbacks of implementing a multi-striping technique, we report results for two different system load conditions: indeed, when the system load is high, but not in critical conditions, i.e. $\rho \leq 0.9$, adopting a multi-striping policy clearly helps at improving performance and increasing the number of stripes H definitely reduces the blocking probability for each content. On the other hand, instead, when the system load is large, i.e. $\rho = 1.0$, the P2P network established by STBs is not able to sustain the request rate for each stripe, so that increasing the number of stripes has a negative impact on performance. Increasing the number of stripes indeed means, in critical system conditions, heightening the blocking probability for each stripe as well, thus reducing the capability of the system to provide all the pieces that are composing contents. Observe that our model fits very well simulation results, positively validating the reliability of this extension.

D. Non Reliable Peers Extension Evaluation

We now focus on the evaluation of the extension shown in Sec. IV-F, which models the case in which the P2P network established by STBs (peers) is affected by churning, i.e. when peers are free to enter and leave the system. Given the total number of replicas in the system X_c for each content c, if peers are not reliable, the actual number of copies available in the system Y_c will be a binomial random variable, such that $Y_c \leq X_c$.

This means that the blocking probability of requests will be strongly influenced by the impossibility of retrieving copies of contents. Fig. 4 depicts the average blocking probability over the 10 classes of contents when increasing the fraction of churning peers P_{churn} (where $P_{churn} = 1 - p_{av}$), i.e. when reducing the number of replicas for each content. System load ρ was equal to 0.9. Obviously, the raise of P_{churn} has strong impairment on performance: it is easy to see that when $P_{churn} = 0.3$, the blocking probability grows of one order of magnitude (e.g. in G_{ADSL}, from 0.02 to 0.2, roughly

speaking). Observe that we get really good fitting between the model and the simulation results for both G_{FTTH} (top) and G_{ADSL} (bottom) scenarios.

E. Heterogeneity Extension Evaluation

Focusing on Sec. IV-G, we evaluate the system response when we introduce some bandwidth heterogeneity into it. To this extent, we define P_{bw} that represents the fraction of peers whose up-link capacity is smaller with respect to the declared setup. For scenario G_{ADSL}, $B \cdot P_{bw}$ STBs will experience an up-link capacity reduced by a half from standard setup, i.e. 0.5Mbit/s. For scenario G_{FTTH}, instead, $B \cdot P_{bw}$ STBs will have an up-link capacity reduced by a third from standard setup, i.e. 6.6Mbit/s. Fig. 5 reports the average blocking probability when varying P_{bw}. Since increasing P_{bw} actually translates in lowering the system overall up-link capacity, performance are going to decrease. Again our model nicely fits simulation results, and even if fitting slowly decreases for larger values of P_{bw}, our model still provides an acceptable pessimistic estimation.

F. Processor Sharing Extension Evaluation

Given the successful validation of the model shown in section Sec. VI-B, we move on, focusing on the extension of our model that considers a push-based policy, i.e. implementing a processor sharing queue on each STB. In this configuration the download rate of peers is dynamically adapted so to exploit at every time the whole available server bandwidth, while a call admission control mechanism still guarantees that minimum download requirements are met.

Of course, the blocking probability is expected to decrease in this case, because the system uses in a more efficient way bandwidth resources. However observe pull-based policies are rather natural in the VoD context, and they allow simplifying the system control. Furthermore observe that push-systems may require to the underlay to transport unnecessary fraction of data every time a user stops watching a movie before its natural end. This configuration induces the P2P VoD service to resemble somehow a much greedier system, more similar to a BitTorrent-like P2P file-sharing application, in which peers are free to download movies at the maximum available speed. Of course, the blocking probability is expected to decrease in this case, but at the cost of using all available bandwidth, with the possibility of wasting data, and thus bandwidth, if users stop watching the movie before its end.

In Fig. 6 we directly compare pull-based (M/G/K/K) and push-based (M/G/K/1 PS) approaches for both scenarios G_{FTTH} and G_{ADSL}. Observe that the push-based policy clearly overcomes the pull-based one, reducing the blocking probability by a factor 2 for G_{FTTH}. However in case users have very limited resource, such as in the G_{ADSL} scenario, the beneficial effect of downloading at maximum possible rate becomes less significant (with respect to G_{FTTH}). Indeed, in the G_{ADSL} scenario system performance is intrinsically limited by the scarcity of bandwidth resources.

G. Scaling the Model

In order to provide a complete view about the flexibility of our model, we tested it against our simulator again, but varying different system parameters. Fig. 7 shows how the blocking probability, aggregated over content classes, scales with the number of STBs participating to the system, from $B = 2k$ to $B = 10k$. System load ρ was 1.0. Of course, in this case, the size of the downloadable contents C has been increased proportionally to B. Notice that enlarging the pool of STBs translates into a more uniform distribution of replicas over the P2P network, thus leading to a better share of resources and, then, to a lower blocking probability. Also in this case, we observe a good fitting between model and simulation curves for both G_{FTTH} (top) and G_{ADSL} (bottom) scenarios.
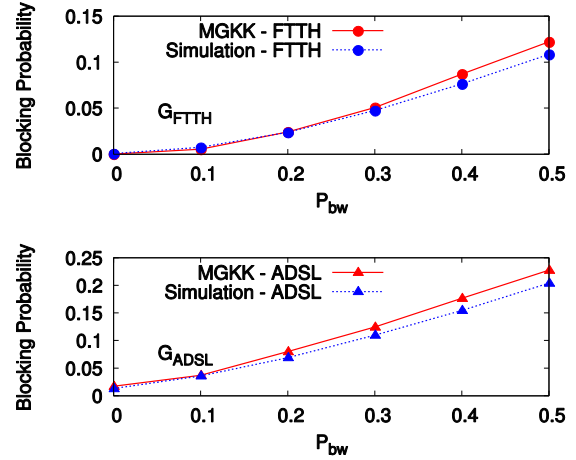

Fig. 5. Comparison of the blocking probability between analytical model M/G/K/K and simulation results when increasing the bandwidth heterogeneity in the system P_{bw}. System load $\rho = 0.9$ has been considered for both G_{FTTH} (top) and G_{ADSL} (bottom) scenarios.
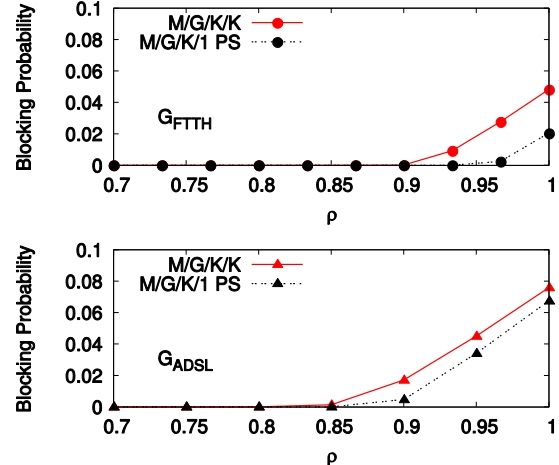

Fig. 6. Comparison of the blocking probability between analytical models M/G/K/K and M/G/K/1 PS when increasing the system load. Both G_{FTTH} (top) and G_{ADSL} (bottom) scenarios have been considered.

At last we would like to emphasize that our model permits to arbitrarily scale at no computational costs both the number of users in the system as well as the number of contents, and thus provide an efficient and flexible tool for performance

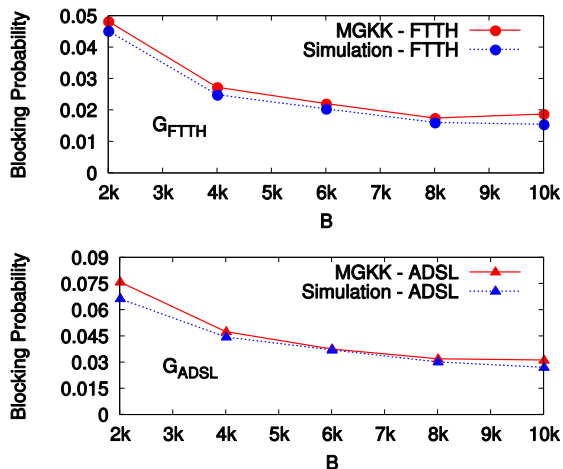evaluation and system design of large scale systems.

Fig. 7. Comparison of the blocking probability between analytical model M/G/K/K and simulation results when increasing the number of STBs in the system for both G_{FTTH} (top) and G_{ADSL} (bottom) scenarios. System load $\rho = 1.0$.

VII. CONCLUSION

In this paper, we have developed a simple, highly scalable and versatile queuing theoretic analytical model for performance evaluation of large scale STB-based VoD systems. Our model represents the evolution of available upload bandwidth at every STB by a simple M/G/K/K, under the assumption that content downloads proceed at constant nominal rate. We have shown how our model can be extended in several respects, considering the effect of user churning, of sub-streaming, and of bandwidth heterogeneity. Furthermore, our model can also be used to dimension the bandwidth of servers in peer-assisted hybrid scenarios exploiting backup servers. At last, we generalized our basic model to consider systems in which the download rate may speed up beyond the nominal rate to better exploit the system bandwidth. In the last case, upload bandwidth evolution at peers is represented by single servers Processor Sharing (PS) queues. The proposed model has been validated through a wide experimental campaign. We compared model predictions against those obtained from an event-driven simulator. In all cases the proposed model proves to be a reliable tool for the design of a P2P VoD system.

REFERENCES

[1] N. Laoutaris, P. Rodriguez, and L. Massoulie, "Echos: edge capacity hosting overlays of nano data centers," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 51-54, 2008.

[2] F. Liu, B. Li, B. Li, and H. Jin, "Peer-assisted on-demand streaming: Characterizing demands and optimizing supplies," *IEEE Transactions on Computers*, to be published. [Online]. Available: http://dx.doi.org/10.1109/TC.2011.222

[3] B. Tan and L. Massoulié, "Optimal content placement for peer-to-peer video-on-demand systems," in *Proceedings of IEEE INFOCOM 2011*, 2011, pp. 694-702.

[4] K. Wang and C. Lin, "Insight into the p2p-vod system: Performance modeling and analysis," in *Proceedings of 18th International Conference on Computer Communications and Networks*, ser. ICCCN '09, 2009, pp. 1-6.

[5] W. Wu and J. C. Lui, "Exploring the optimal replication strategy in p2p-vod systems: Characterization and evaluation," in *Proceedings of IEEE INFOCOM 2011*, 2011, pp. 1206-1214.

[6] S. A. Chellouche, D. Négru, Y. Chen, and M. Sidibe, "Home-box-assisted content delivery network for internet video-on-demand services," in *Proceedings of IEEE ICC 2012*, to be published.

[7] X. Wang, C. Zheng, Z. Zhang, H. Lu, and X. Xue, "The design of video segmentation-aided VCR support for P2P VoD systems," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 531-537, May 2008.

[8] L. Yu, L. Gao, J. Zhao, and X. Wang, "SonicVoD: a VCR-supported P2P-VoD system with network coding," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 576-582, May 2009.

[9] Y. R. Choe, D. L. Schuff, J. M. Dyaberi, and V. S. Pai, "Improving vod server efficiency with bittorrent," in *Proceedings of the 15th International Conference on Multimedia*, ser. MULTIMEDIA '07, 2007, pp. 117-126.

[10] V. Janardhan and H. Schulzrinne, "Peer assisted vod for set-top box based ip network," in *Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*, ser. P2P-TV '07, 2007, pp. 335-339.

[11] K. Suh, C. Diot, J. Kurose, L. Massoulié, C. Neumann, D. F. Towsley, and M. Varvello, "Push-to-peer video-on-demand system: Design and evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 1706-1716, 2007.

[12] A. Nafaa, S. Murphy, and L. Murphy, "Analysis of a large-scale vod architecture for broadband operators: A p2p-based solution," *IEEE Communications Magazine*, vol. 46, no. 12, pp. 47-55, 2008.

[13] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," *CoRR*, vol. abs/1202.0108, 2012.

[14] J. P. Muñoz-Gea, S. Traverso, and E. Leonardi, "Modeling p2p vod systems: methods and approximations," *Technical Report*, July 2012. (http://www.tlc-networks.polito.it/oldsite/traverso/papers/tr_07-10-2012.pdf)

[15] J. W. Roberts, "A service system with heterogeneous use requirements-applications to multi-service telecommunication systems," in *Proceedings of the International Conference on Performance Data Communication Systems and Their Applications*, 1981, pp. 423-431.

[16] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 17, no. 9, pp. 1103-1120, 2007.

[17] C. Jayasundara, A. Nirmalathas, E. Wong, and C. A. Chan, "Localized p2p vod delivery scheme with pre-fetching for broadband access networks," in *Proceedings of IEEE Globecom 2011*, 2011, pp. 1-5.

BIOGRAPHIES
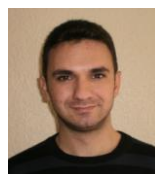
Juan Pedro Muñoz-Gea received the Master's Degree in Telecommunications Engineering in 2005 from Universidad Politécnica de Cartagena (UPCT), Spain. In 2006, he started working as a research assistant at UPCT where he obtained his PhD in 2011. Currently, he works as an assistant professor in the ICT Department at UPCT. His research interests are focused on overlay networks and peer-to-peer systems.

Stefano Traverso is a Post-Doc fellow at Politecnico di Torino, Italy. He holds a Master degree in Computer and Communication Networks Engineering (2008) and a PhD in Electronics and Communications Engineering (2012), both form Politecnico di Torino. He was a visiting PhD student at Telefonica I+D research center in Barcelona, Spain. His research interests are P2P applications and online social networks.

Emilio Leonardi (M'99-SM'09) received a PhD in Telecommunications Engineering in 1995 from Politecnico di Torino, where he is currently an associate professor. In 1995, he visited the Computer Science Department of UCLA; in summer 1999 he joined the Bell Laboratories/Lucent Technologies, Holmdel (NJ); in 2001, the Electrical Engineering Department of the Stanford University and finally in 2003, the Sprint Advanced Technologies Laboratories, Burlingame (CA). His research

interests are in the field of performance evaluation of wireless networks and P2P streaming systems.