Optimal Content Placement for Peer-to-Peer Video-on-Demand Systems[1]

Bo (Rambo) Tan
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
Email: botan2@illinois.edu

Laurent Massoulié
Technicolor Paris Research Lab
Issy-les-Moulineaux Cedex 92648, France
Email: laurent.massoulie@technicolor.com

***Abstract*—In this paper, we address the problem of content placement in peer-to-peer systems, with the objective of maximizing the utilization of peers' uplink bandwidth resources. We consider system performance under a many-user asymptotic. We distinguish two scenarios, namely "Distributed Server Networks" (DSN) for which requests are exogenous to the system, and "Pure P2P Networks" (PP2PN) for which requests emanate from the peers themselves. For both scenarios, we consider a *loss network* model of performance, and determine asymptotically optimal content placement strategies in the case of a limited content catalogue. We then turn to an alternative "large catalogue" scaling where the catalogue size scales with the peer population. Under this scaling, we establish that storage space per peer must necessarily grow unboundedly if bandwidth utilization is to be maximized. Relating the system performance to properties of a specific random graph model, we then identify a content placement strategy and a request acceptance policy which jointly maximize bandwidth utilization, provided storage space per peer grows unboundedly, although arbitrarily slowly, with system size.**

I. Introduction

The amount of multimedia traffic accessed via the Internet, already of the order of exabytes (10^{18}) per month, is expected to grow steadily in the coming years. A peer-to-peer (P2P) architecture, whereby peers contribute resources to support service of such traffic, holds the promise to support such growth more cheaply than by scaling up the size of data centers. More precisely, a large-scale P2P system based on resources of individual users can absorb part of the load that would otherwise need to be served by data centers.

In the present work we address specifically the Video-on-Demand (VoD) application, for which the critical resources at the peers are storage space and uplink bandwidth. Our objective is to ensure that the largest fraction of traffic is supported by the P2P system. More precisely, we look for content placement strategies that enable content downloaders to maximally use the peers' uplink bandwidth, and hence maximally offload the servers in the data centers. Such strategies must adjust to the distinct popularity of video contents, as a more popular content should be replicated more frequently.

We consider the following mode of operation: Video requests are first submitted to the P2P system; if they are accepted, uplink bandwidth is used to serve them at the video streaming rate (potentially via parallel substreams from different peers). They are rejected if their acceptance would require disruption of an ongoing request service. Rejected requests are then handled by the data center. Alternative modes of operation could be envisioned (e.g., enqueueing of requests, service at rates distinct from the streaming rate, joint service by peers and data center,...). However the proposed model is appealing for the following reasons. It ensures zero waiting time for requests, which is desirable for VoD application; analysis is facilitated, since the system can be modeled as a *loss network* [6], for which powerful theoretical results are available; and finally, as our results show, simple placement strategies ensure optimal operation in the present model.

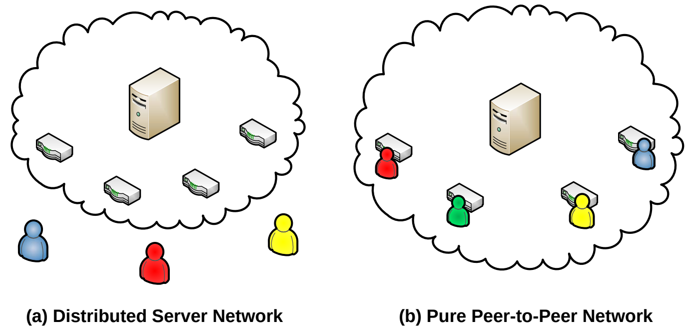

Fig. 1: Two architectures of P2P VoD systems

In the P2P system we are considering, there are two kinds of peers: boxes and pure users. Their difference is that boxes do contribute resources (storage space and uplink bandwidth) to the system, while pure users do not. This paper focuses on the following two architectures (illustrated in Figure 1):

- **Distributed Server Network (DSN):** Requests to download contents come only from pure users, and can be regarded as external requests.
- **Pure P2P Network (PP2PN):** There are no pure users in the system, and boxes do generate content requests, which can be regarded as "internal".

The rest of the paper is organized as follows: We review related work in Section II and introduce our system model in Section III. For the Distributed Server Network scenario, the

[1] Several results developed in the present paper have made the object of a "brief announcement" [10].

so-called "proportional-to-product" content placement strategy is introduced and shown to be optimal in a large system limit in Section IV. A distinct placement strategy is introduced and proved optimal for the Pure P2P Network scenario in Section V. These results apply for a catalogue of contents of limited size. An alternative model in which catalogue size grows with the user population is introduced in Section VI, where it is shown that the "proportional-to-product" placement strategy remains optimal in the DSN scenario in this large catalogue setting, for a suitably modified request management technique.

II. Related Work

The number and location of replicas of distinct content objects in a P2P system have a strong impact on such system's performance. Indeed, together with the strategy for handling incoming requests, they determine whether such requests must either be delayed, or served from an alternative, more expensive source such as a remote data center. Requests which cannot start service at once can either be enqueued (we then speak of a waiting model) or redirected (we then speak of a loss model).

Previous investigations of content placement for P2P VoD systems were conducted by Suh et al. [9]. The problem tackled in [9] differs from our current perspective, in particular no optimization of placement with respect to content popularity was attempted in this work. Performance analysis of both queueing and loss models are considered in [9]. Valancius et al. [14] considered content placement dependent on content popularity, based on a heuristic linear program, and validated this heuristic's performance in a loss model via simulations.

Tewari and Kleinrock [12], [13] advocated to tune the number of replicas in proportion to the request rate of the corresponding content, based on a simple queueing formula, for a waiting model, and also from the standpoint of the load on network links. They further established via simulations that Least Recently Used (LRU) storage management policies at peers emulated rather well their proposed allocation.

Wu et al. [15] considered a loss model, and a specific time-slotted mode of operation whereby requests are submitted to randomly selected peers, who accommodate a randomly selected request. They showed that in this setup the optimal cache update strategy can be expressed as a dynamic program. Through experiments, they established that simple mechanisms such as LRU or Least Frequently Used (LFU) perform close to the optimal strategy they had previously characterized.

Kangasharju et al. [5] addressed file replication in an environment where peers are intermittently available, with the aim of maximizing the probability of a requested file being present at an available peer. This differs from our present focus in that the bandwidth limitation of peers is not taken into account, while the emphasis is on their intermittent presence. They established optimality of content replication in proportion to the *logarithm* of its popularity, and identified simple heuristics approaching this.

Boufkhad et al. [3] considered P2P VoD from yet another viewpoint, looking at the number of contents that can be simultaneously served by a collection of peers.

Content placement problem has also been addressed towards other different optimization objectives. For example, Almeida et al. [1] aim at minimizing total delivery cost in the network, and Zhou et al. [16] target jointly maximizing the average encoding bit rate and average number of content replicas as well as minimizing the communication load imbalance of video servers.

Cache dimensioning problem is considered in [7], where Laoutaris et al. optimized the storage capacity allocation for content distribution networks under a limited total cache storage budget, so as to reduce average fetch distance for the request contents with consideration of load balancing and workload constraints on a given node. Our paper takes a different perspective, focusing on many-user asymptotics so the results show that the finite storage capacity per node is never a bottleneck (even in the "large catalogue model", it also scales to infinity more slowly than the system size).

There are obvious similarities between our present objective and the above works. However, none of these identifies explicit content placement strategies at the level of the individual peers, which lead to minimal fraction of redirected (lost) requests in a setup with dynamic arrivals of requests.

Finally, there is a rich literature on loss networks (see in particular Kelly [6]); however our present concern of optimizing placement to minimize the amount of rejected traffic in a corresponding loss network appears new.

III. Model Description

We now introduce our mathematical model and related notations. Denote the set of all boxes as $\mathcal{B}$. Let $|\mathcal{B}| = B$ and index the boxes from 1 to B. Box b has a local cache $\mathcal{J}_b$ that can store up to M contents, all boxes having the same storage space M. We further assume that each box can simultaneously serve U concurrent requests, where U is an integer, i.e., each box has an uplink bandwidth equal to U times the video streaming rate. In particular we assume identical streaming rates for all contents.

The set of available contents is defined as $\mathcal{C}$. Let $|\mathcal{C}| = C$ and index contents from 1 to C. Thus a given box b will be able to serve requests for content c for all $c \in \mathcal{J}_b$.

In a Pure P2P Network, when box b has a request for a certain content c, which is coincidentally already in its cache, a "local service" is provided and no download service is needed, hence the service to this request consumes no bandwidth resource. The effect of local service on deriving an optimal content placement strategy will be discussed in detail in Section V.

In a Distributed Server Network, however, local service will never occur since all the requests are external with respect to the system resources[2].

For a new request that needs a download service, an attempt is made to serve this request by some box holding content c, while ensuring that previously accepted requests can themselves be assigned to adequate boxes, given the cache content and bandwidth resources of all boxes. This potentially involves "repacking" of requests, i.e., reallocation of all the bandwidth resources in the system ("box-serving-request" mapping) to accommodate this new download demand pattern. If such repacking can be found, then the request is accepted; otherwise, it is rejected from the P2P system.

It will be useful in the sequel to characterize the concurrent numbers of requests that are amenable to such repacking. Let $\mathbf{n} = \{n_c\}_{c\in\mathcal{C}}$ be the vector of numbers n_c of requests per content c. A matching of these requests to server boxes is feasible iff. there exist nonnegative integers z_{cb} (number of concurrent downloads of content c from box b) such that

$$\begin{aligned} \sum_{b:c\in\mathcal{J}_b} z_{cb} &= n_c, \ \forall\ c\in\mathcal{C}; \\ \sum_{c:c\in\mathcal{J}_b} z_{cb} &\le U, \ \forall\ b\in\mathcal{B}. \end{aligned} \tag{1}$$

A more compact characterization of feasibility follows by an application of Hall's theorem [2] (detailed in our technical report [11]), giving that $\mathbf{n}$ is feasible if and only if

$$\forall\ \mathcal{S}\subseteq\mathcal{C},\ \sum_{c\in\mathcal{S}} n_c \le U\,|\{b\in\mathcal{B}:\ \mathcal{S}\cap\mathcal{J}_b\neq\emptyset\}|\,. \tag{2}$$

We now introduce statistical assumptions on request arrivals and durations. New requests for content c occur at the instants of a Poisson process with rate ν_c. We assume that the video streaming rate is normalized to 1, and is the same for all contents. We further assume that all videos have the same duration, again normalized at 1. Under these assumptions, the amount of work per time unit brought into the system by content c equals ν_c.

With the above assumptions at hand, assuming fixed cache contents, the vector $\mathbf{n}$ of requests under service is a particular instance of a general stochastic process known as a loss network model. Loss networks were introduced to represent ongoing calls in telephone networks, and exhibit rich structure. In particular, the corresponding stochastic process is reversible, and admits a closed-form stationary distribution. For the Distributed Server Network model, the stationary distribution reads:

$$\pi(\mathbf{n}) = \frac{1}{Z}\prod_{c\in\mathcal{C}}\frac{\nu_c^{n_c}}{n_c!}\mathcal{I}_{\{\mathbf{n}\text{ is feasible}\}}. \tag{3}$$

In words, the numbers of requests n_c are independent Poisson random variables with parameter ν_c, conditioned on feasibility of the whole vector $\mathbf{n}$.

Our objective is then to determine content placement strategies so that in the corresponding loss network model, the fraction of rejected requests is minimal. The difficulty in doing this analysis resides in the fact that the normalizing constant Z is cumbersome to evaluate. Nevertheless, simplifications occur under large system asymptotics, which we will exploit in the next sections.

We conclude this section by the following remark. For simplicity we assumed in the above description that a particular content is either fully replicated at a peer, or not present at all, and that a request is served from only one peer. It should however be noted that we can equally assume that contents are split into sub-units, which can be placed onto distinct peers, and downloaded from such distinct peers in parallel sub-streams in order to satisfy a request. This extension is detailed in [11].

[2] In fact the external users issuing requests could keep local copies of previously accessed content, and hence experience "local service" upon re-accessing the same content. But we do not need consider this as this happens outside the perimeter of our system.

IV. Optimal Content Placement in Distributed Server Networks

We first describe a simple adaptive cache update strategy driven by demand, and show why it converges to a "predetermined" content placement called "proportional-to-product" strategy. We then establish the optimality of this "proportional-to-product" placement in a large system asymptotic regime.

A. The Proportional-to-Product Placement Strategy

A simple method to adaptively update the caches at boxes driven by demand is described as follows: Whenever a new request comes, with probability ϵB (ϵ is chosen such that $\epsilon B \le 1$), the server picks a box b uniformly at random, and attempts to push content c into this box's cache. If c is already in there, do nothing; otherwise, remove a content selected uniformly at random from the cache.

Since external demands for content c are according to a Poisson process with rate ν_c, we find that under the above simple strategy, content c is pushed at rate $\epsilon\nu_c$ into a particular box which is not caching content c. Recall that each box stores M distinct contents, and let j denote a candidate "cache state", which is a size M subset of the full content set $\mathcal{C}$. For convenience, let $\mathcal{J}$ denote the collection of all such j.

With the above strategy, the caches at each box evolve independently according to a continuous-time Markov process. The rate at which cache state j is changed to j', where $j' = j + \{c\} \setminus \{d\}$ for some contents $d\in j$, $c\notin j$, which we denote by $q(j,j')$, is easily seen to be $q(j,j') = \epsilon\nu_c/M$. Indeed, content d is evicted with probability $1/M$, while content c is introduced at rate $\epsilon\nu_c$.

It is easy to verify that the distribution $p(\cdot)$ given by

$$p(j) = \frac{1}{Z}\prod_{c\in j}\nu_c, \quad j\in\mathcal{J}, \tag{4}$$

for some normalizing constant Z, verifies the follwing equation:

$$p(j)q(j,j') = p(j')q(j',j), \quad j,j'\in\mathcal{J}. \tag{5}$$

The latter relations, known as the local balance equations, readily imply that $p(\cdot)$ is a stationary distribution for the above Markov process; since the process is irreducible, this is the unique stationary distribution.

Thus, we can conclude that under this cache update strategy, the random cache state at any box eventually follows this stationary distribution. This is what we refer to as the **"proportional-to-product" placement strategy**, and it is the one we advocate in the Distributed Server Network scenario.

Remark 1: The customized parameter ϵ should not be too large, otherwise the burden on the server will be increased due to use of "push". Neither should it be too small, otherwise the Markov chain will converge too slowly to the steady state.

Under the cache update strategy, the distribution of cache contents needs time to converge to the steady state. If we have a priori information about content popularity, we can use a centralized sampling strategy as an alternative way to directly generate proportional-to-product content placement in one go. One method works as follows: Select successively M contents at random in an i.i.d. fashion, according to the probability distribution $\{\hat{\nu}_c\}$, where $\hat{\nu}_c = \nu_c / \sum_{c' \in \mathcal{C}} \nu_{c'}$ is the normalized popularity. If there are duplicate selections of some content, re-run the procedure. It is readily seen that this yields a sample with the desired distribution[3]. $\diamond$

B. Large Network Asymptotics and Optimization Framework

We now consider the asymptotic regime called **"many user–fixed catalogue" scaling**: The number of boxes B goes to infinity. The system load, defined as

$$\rho \triangleq \frac{\sum_{c \in \mathcal{C}} \nu_c}{BU}, \tag{6}$$

is assumed to remain fixed, which is achieved in the present section by assuming that the content collection $\mathcal{C}$ is kept fixed, while the individual rates $\{\nu_c\}$ scale linearly with B. We also assume that the normalized content popularities $\{\hat{\nu}_c\}$ remain fixed as B increases. It thus holds that $\nu_c = \hat{\nu}_c \rho BU$ for all $c \in \mathcal{C}$. Note that although boxes are pure resources rather than users, scaling of $\{\nu_c\}$ with B to infinity actually indicates a "many-user" scenario.

To analyze the performance of our proposed proportional-to-product strategy, we require that the cache contents are sampled at random according to this strategy and are subsequently kept fixed. This can either reflect the situation where we use the previously introduced sampling strategy, or alternatively the situation where the cache update strategy has already made the distribution of cache states converge to the steady state, and occurs at a slower time scale than that at which new requests arise and complete.

Note that, as B grows large, the right-hand side in the feasibility constraint (2) verifies, by the law of large numbers,

$$|\{b \in \mathcal{B} :\ \mathcal{S} \cap \mathcal{J}_b \neq \emptyset\}| \sim B \sum_{j: j \cap \mathcal{S} \neq \emptyset} m_j,$$

where

$$m_j = \frac{1}{Z} \prod_{c \in j} \hat{\nu}_c,$$

[3]We describe an alternative exact sampling strategy in [11] which can be faster than the one just described when very popular contents are present.

and Z is a normalizing constant. Indeed, under proportional-to-product placement, each box holds a size M content set j with probability m_j, and this happens independently over boxes.

We are now in a particular setup of a loss network indexed by a large parameter B, where the set of request types, $\mathcal{C}$, is left unchanged, the corresponding request rate ν_c scaling linearly in B, and the set of feasibility constraints (2) is also unchanged, except that the right-hand side scales again linearly with B. This particular setup has been identified as the "large capacity network scaling" in Kelly [6]. There, it is shown that the loss probabilities in the limiting regime where $B \to \infty$ can be characterized via the analysis of an associated variational problem. We now describe the corresponding results of [6] relevant to our present purpose.

Consider the problem of finding the mode of the stationary distribution (3), which corresponds to maximizing $\sum_{c \in \mathcal{C}} (n_c \log \nu_c - \log n_c!)$ over feasible $\mathbf{n}$. Then, approximate $\log n_c!$ by $n_c \log n_c - n_c$ according to Stirling's formula and replace the integer vector $\mathbf{n}$ by a real-valued vector $\mathbf{x}$. This leads to the following optimization problem:

[OPT 1]

$$\max_{\mathbf{x}} \quad \sum_{c \in \mathcal{C}} (x_c \log \nu_c - x_c \log x_c + x_c) \tag{7}$$

$$s.t. \quad \forall\, \mathcal{S} \subseteq \mathcal{C},\ \sum_{c \in \mathcal{S}} x_c \leq \sum_{j: j \cap \mathcal{S} \neq \emptyset} m_j BU \tag{8}$$

$$\text{over} \quad \mathbf{x} \geq 0.$$

The corresponding Lagrangian is given by:

$$\begin{aligned} L(\mathbf{x}, \mathbf{y}) \quad = \quad & \sum_{c \in \mathcal{C}} (x_c \log \nu_c - x_c \log x_c + x_c) \\ & + \sum_{\mathcal{S} \subseteq \mathcal{C}} y_{\mathcal{S}} \Big(\sum_{j: j \cap \mathcal{S} \neq \emptyset} m_j BU - \sum_{c \in \mathcal{S}} x_c \Big), \end{aligned}$$

where $\{y_{\mathcal{S}}\}_{\mathcal{S} \subseteq \mathcal{C}}$ are Lagrangian multipliers. The KKT conditions for this convex optimization problem comprise the original constraints and the following ones:

$$\frac{\partial L}{\partial x_c}(\mathbf{x}^*, \mathbf{y}^*) = \log \nu_c - \log x_c^* - \sum_{\mathcal{S}: c \in \mathcal{S}} y_{\mathcal{S}}^* = 0,\ \forall\ c \in \mathcal{C} \tag{9}$$

$$y_{\mathcal{S}}^* \Big(\sum_{j: j \cap \mathcal{S} \neq \emptyset} m_j BU - \sum_{c \in \mathcal{S}} x_c^* \Big) = 0,\ y_{\mathcal{S}}^* \geq 0,\ \forall\ \mathcal{S} \subseteq \mathcal{C}, \tag{10}$$

where $(\mathbf{x}^*, \mathbf{y}^*)$ is a solution to the optimization problem. From equation (9), we further get

$$x_c^* = \nu_c \exp\Big(- \sum_{\mathcal{S}: c \in \mathcal{S}} y_{\mathcal{S}}^*\Big),\ \forall\ c \in \mathcal{C}. \tag{11}$$

Then the result that we will need from Kelly [6] is the following: In the limit $B \to \infty$, the steady state probability of accepting request for c, denoted by A_c, verifies

$$\lim_{B \to \infty} A_c = \exp\Big(- \sum_{\mathcal{S}: c \in \mathcal{S}} y_{\mathcal{S}}^*\Big),\ \forall\ c \in \mathcal{C}, \tag{12}$$

where $y_{\mathcal{S}}^*$ are the Lagrangian multipliers of the previous optimization problem.

C. Optimality of Proportional-to-Product Content Placement

Note that the global acceptance probability, denoted by A_{sys}, which also reads $A_{sys} = \sum_{c\in\mathcal{C}} \hat{\nu}_c A_c$, cannot exceed $\min(1, 1/\rho)$. Indeed, it is clearly no larger than 1. It cannot exceed $1/\rho$ either, otherwise the system would treat more requests than its available resources.

We now prove that the proportional-to-product content placement not only achieves the optimal global acceptance probability $A_{sys} = \min(1, 1/\rho)$, but also achieves fair individual acceptance probabilities, i.e., $A_c = A_{sys}$ for all c. More precisely, we have the following theorem:

Theorem 1: Let $m_j = \prod_{c\in j} \hat{\nu}_c/Z$ for all $j \subseteq \mathcal{C}$ s.t. $|j| = M$, where Z is the normalizing constant. Then, $\lim_{B\to\infty} A_c = \min\{1, 1/\rho\},\ \forall\ c \in \mathcal{C}$, for fixed ρ and $\mathcal{C}$. $\diamond$

Before giving the proof, we comment on the result. One point to note is that the optimality of the asymptotic acceptance probability does not depend on M, as long as $M \geq 1$. Thus for this particular scaling regime, storage space is not a bottleneck. As we shall see in the next two sections, increasing M **does** improve performance if either local services occur, as in the Pure P2P Network scenario (Section 4), or if the catalogue size C scales with the box population size B, a case not covered by the classical literature on loss networks, and to which we turn in Section VI-B.

Proof: First, we consider $\rho \geq 1$. From equation (12), assuming $A_c = 1/\rho$, we have

$$\forall\ \mathcal{S} \subseteq \mathcal{C},\ \sum_{\mathcal{S}: c\in\mathcal{S}} y^*_{\mathcal{S}} = \log\rho. \tag{13}$$

Putting equation (13) into (11) leads to

$$\forall\ c \in \mathcal{C},\ x^*_c = \nu_c/\rho. \tag{14}$$

Thus, inequality (8) in OPT 1 becomes

$$\forall \mathcal{S} \subseteq \mathcal{C},\ \sum_{c\in\mathcal{S}} \nu_c \leq \rho \sum_{j: j\cap\mathcal{S}\neq\emptyset} m_j BU. \tag{15}$$

Since $\nu_c = \rho BU \cdot \hat{\nu}_c$ and $\sum_{c\in\mathcal{C}} \hat{\nu}_c = 1$, inequality (15) further becomes, upon expliciting the normalization constant Z:

$$\forall\ \mathcal{S} \subseteq \mathcal{C},\ \sum_{c\in\mathcal{S}} \hat{\nu}_c \cdot \sum_{\substack{\mathcal{G}:\ \mathcal{G}\subseteq\mathcal{C} \\ |\mathcal{G}|=M}} \prod_{c\in\mathcal{G}} \hat{\nu}_c \leq \sum_{c\in\mathcal{C}} \hat{\nu}_c \cdot \sum_{\substack{\mathcal{G}:\ \mathcal{G}\cap\mathcal{S}\neq\emptyset \\ \mathcal{G}\subseteq\mathcal{C} \\ |\mathcal{G}|=M}} \prod_{c\in\mathcal{G}} \hat{\nu}_c. \tag{16}$$

Two types of product terms (mapped to subsets $\mathcal{K} \subseteq \mathcal{C}$) appear on both sides:

I. $\prod_{c\in\mathcal{K}} \hat{\nu}_c$: $|\mathcal{K}| = M + 1,\ \mathcal{K} \cap S \neq \emptyset$.
II. $(\prod_{c\in\mathcal{K}} \hat{\nu}_c) \cdot \hat{\nu}_{c'}$: $c' \in \mathcal{K} \cap S,\ |\mathcal{K}| = M$.

To show whether inequality (16) hold, we only have to prove that given any $\mathcal{S} \subseteq \mathcal{C}$, for each product term (related to a $\mathcal{K}$) which appears in one inequality corresponding to a certain $\mathcal{S}$, its multiplicity on the left hand side is no more than that on the right hand side.

1. **For a product term of Type I:**
   - On the LHS: Since $\prod_{c\in\mathcal{K}} \hat{\nu}_c = \prod_{c\in\mathcal{G}} \hat{\nu}_c \cdot \hat{\nu}_{c'}$ for some $\mathcal{G} \subseteq \mathcal{C}$ and $c' \in \mathcal{S} \cap \mathcal{K}$, where $\mathcal{G}$ is a size M content set, $c' \notin \mathcal{G}$, and $\mathcal{K} = \mathcal{G} + \{c'\}$. It is easy to see that we have $|\mathcal{S} \cap \mathcal{K}|$ different choice of c' in a $\mathcal{K}$, so the multiplicity of this product term on the LHS equals $|\mathcal{S} \cap \mathcal{K}|$.
   - On the RHS: When $|\mathcal{S} \cap \mathcal{K}| \geq 2$, for any $c' \in \mathcal{K}$, $\mathcal{K}\backslash\{c'\}$ is a size M content set of which the intersect with $\mathcal{S}$ is not empty, hence the multiplicity equals $|\mathcal{K}|\ (= M+1)$. When $|\mathcal{S}\cap\mathcal{K}| = 1$, the exception to the above case is that if $c' \in \mathcal{S}\cap\mathcal{K}$, then $\mathcal{K} \setminus \{c'\}$ is a size M content set which has no intersect with $\mathcal{S}$ and is actually impossible to appear in the second summation term (over all size M content sets $\mathcal{G}$ s.t. $\mathcal{G}\cap\mathcal{S} \neq \emptyset$) in inequality (16). Thus, the multiplicity equals $|\mathcal{K}| - 1\ (= M)$.

   From above, we can see that the multiplicity of the product term on the LHS is always no more than that on the RHS.
2. **For a product term of Type II:**
   $\mathcal{K}$ is actually already a size M content set $\mathcal{G}$ s.t. $\mathcal{G}\cap\mathcal{C} \neq \emptyset$. Therefore, it is easy to see that on both sides, the multiplicities of this product term are both 1.

Now we can conclude that inequality (16) holds for all $\mathcal{S} \subseteq \mathcal{C}$, and continue to check the complementary slackness. Given $\rho \geq 1$, one simple solution to equation (13) reads:

$$\forall\ \mathcal{S} \subseteq \mathcal{C},\ y^*_{\mathcal{S}} = \log\rho \cdot \mathcal{I}_{\{\mathcal{S}=\mathcal{C}\}}. \tag{17}$$

Besides, inequality (16) is tight for $\mathcal{S} = \mathcal{C}$ (we even do not need to check this when $\rho = 1$). Therefore, complementary slackness is always satisfied with solution (17).

So far we have proved that the KKT condition holds when $\rho \geq 1$. When $\rho < 1$, there is an additional factor $1/\rho > 1$ on the RHS of inequality (16). Since the old version of inequalities (16) is proved to hold, the new version automatically holds, but none of them is tight now. However, since when $\rho < 1$, we have $A_c = 1,\ \forall\ c \in \mathcal{C}$, so that $y^*_{\mathcal{S}} = 0,\ \forall\ \mathcal{S} \subseteq \mathcal{C}$, which means complementary slackness is always satisfied (similar to $\rho = 1$). This completes the proof. ■

V. Optimal Content Placement in Pure Peer-to-Peer Networks

In the Pure P2P Network scenario, when box b has a request for content c which is currently in its own cache, a "local service" will be provided and no download bandwidth in the network will be consumed. To simplify our analysis, each request for a specific content is assumed to originate from a box chosen uniformly at random (this in particular assumes identical tastes of all users).

This means that the effective arrival rate of the requests for content c which generates traffic load actually equals $\tilde{\nu}_c \triangleq \nu_c(1 - \tilde{m}_c)$, where $\tilde{m}_c$ is defined as the fraction of boxes who have cached content c. Let $\rho_c \triangleq \rho\hat{\nu}_c$ denote the traffic load generated by requests for content c, and λ_c denote the fraction of the system bandwidth resources used to serve requests for content c. Obviously, $\sum_{c\in\mathcal{C}} \lambda_c \leq 1$. The traffic load absorbed by the P2P system either via local services or via service from another box is then upper-bounded by

$$\tilde{\rho} = \sum_{c\in\mathcal{C}} \rho_c \tilde{m}_c + [\rho_c(1-\tilde{m}_c)] \wedge \lambda_c, \tag{18}$$

where "$\wedge$" denotes the minimum operator.

We will use this simple upper bound to identify an optimal placement strategy in the present Pure P2P Network scenario. To this end, we shall establish that our candidate placement strategy asymptotically achieves this performance bound, namely absorbs a portion $\tilde{\rho}$ as $B \to \infty$.

To find the optimal strategy, we introduce a variable $x_c \triangleq [\rho_c(1-\tilde{m}_c)] \wedge \lambda_c$ for all c. Note further that the fraction λ_c is necessarily bounded from above by $\tilde{m}_c$, as only those boxes holding c can devote their bandwidth to serving c. It is then easy to see that the quantity $\tilde{\rho}$ in (18) is no larger than the optimal value of the following linear program:

[OPT 2]

$$\begin{aligned} \max_{\tilde{\mathbf{m}},\boldsymbol{\lambda},\mathbf{x}} \quad & \sum_{c\in\mathcal{C}} (\rho_c \tilde{m}_c + x_c) \\ s.t. \quad & \forall\ c \in \mathcal{C},\ 0 \leq \tilde{m}_c \leq 1,\ 0 \leq \lambda_c \leq \tilde{m}_c; \\ & \forall\ c \in \mathcal{C},\ 0 \leq x_c \leq \lambda_c,\ x_c \leq \rho_c(1-\tilde{m}_c); \\ & \sum_{c\in\mathcal{C}} \tilde{m}_c = M,\ \sum_{c\in\mathcal{C}} \lambda_c \leq 1. \end{aligned}$$

The following theorem gives the structure of an optimal solution to OPT 2.

Theorem 2: Assume that $\{\hat{\nu}_c\}$ are ranked in descending order. The following solution solves OPT 2:

- For $1 \leq c \leq M-1$, $\tilde{m}_c = 1$, $\lambda_c = x_c = 0$.
- For $M \leq c \leq c^*$, $\tilde{m}_c = \lambda_c = x_c = \rho_c/(1+\rho_c)$, where c^* satisfies that
$$\sum_{c=M}^{c^*} \frac{\rho_c}{1+\rho_c} \leq 1, \text{ but } \sum_{c=M}^{c^*+1} \frac{\rho_c}{1+\rho_c} > 1.$$
- For $c = c^*+1$, $\tilde{m}_c = \lambda_c = x_c = 1 - \sum_{c=M}^{c^*} \tilde{m}_c$.
- For $c^*+2 \leq c \leq C$, $\tilde{m}_c = \lambda_c = x_c = 0$. ◇

The proof consists in checking that the KKT conditions are met for the above candidate solution. Details are given in [11].

The above optimal solution suggests the following placement strategy:

"Hot-Warm-Cold" Content Placement Strategy

Divide the contents into three different classes according to their popularity ranking (in descending order):

- **Hot:** The $M-1$ most popular contents. Each box reserves $M-1$ cache slots for them to make sure that requests for these contents are always met via local service.
- **Warm:** The contents with indices from M to c^*+1 (or c^* if $\sum_{c=M}^{c^*} \tilde{m}_c = 1$). For these contents, a fraction $\tilde{m}_c$ of all the boxes will store content c in their remaining one cache slots, where the value of $\tilde{m}_c$ is given in Theorem 2. All requests for these contents (except c^*+1 if it is classified as "warm") can be served, at the expense of all bandwidth resources.
- **Cold:** The other less popular contents are not cached.

Remark 2: The requests for the c^* most popular contents ("hot" contents and "warm" contents except content c^*+1) incur zero loss, while the requests for the $C-c^*-1$ least popular contents incur 100% loss. There is a partial loss in the requests for content c^*+1 if $\sum_{c=M}^{c^*} \tilde{m}_c < 1$.

Note that the placement for "warm" contents looks like the "water-filling" solution in the problem of allocating transmission powers onto different OFDM channels to maximize the overall achievable channel capacity in the context of wireless communications. ◇

Under this placement strategy, the maximum upper bound on the absorbed traffic load reads

$$\tilde{\rho} = \sum_{c=1}^{c^*} \rho_c + (\rho_{c^*+1}+1)\left(1 - \sum_{c=M}^{c^*} \frac{\rho_c}{1+\rho_c}\right).$$

We then have the following corollary:

Corollary 1: Considering the large system limit $B \to \infty$, with fixed catalogue and associated normalized popularities $\{\hat{\nu}_c\}$ as considered in Subsection IV-B, the proposed "hot-warm-cold" placement strategy achieves an asymptotic fraction of absorbed load equal to the above upper bound $\tilde{\rho}$, and is hence optimal in this sense. ◇

Proof: With the proposed placement strategy, hot (respectively, cold) contents never trigger accepted requests, since all incoming requests are handled by local service (respectively, rejected). For warm contents, because each box holds only one warm content, it can only handle requests for that particular warm content. As a result, the processes of ongoing requests for distinct warm contents evolve independently of one another. For a given warm content c, the corresponding number of ongoing requests behaves as a simple one-dimensional loss network with arrival rate $\nu_c(1-\tilde{m}_c)$ and service capacity $\tilde{m}_c BU$. For $c = M, \ldots, c^*$, one has $\tilde{m}_c = \rho_c/(1+\rho_c)$ where $\rho_c = \nu_c/(BU)$, so both the arrival rate and the capacity of the corresponding loss network equal $\tilde{m}_c BU$. The asymptotic acceptance probability as $B \to \infty$ then converges to 1 and the accepted load due to both local service and services from other boxes converges to ρ_c. For content c^*+1 (if $\tilde{m}_{c^*+1} > 0$), the corresponding loss network has arrival rate $\nu_{c^*+1}(1-\tilde{m}_{c^*+1})$ and service capacity $\tilde{m}_{c^*+1}BU$. Then, in the limit $B \to \infty$, the accepted load (due to both local services and services from other boxes) reads $\rho_{c^*+1}\tilde{m}_{c^*+1} + \tilde{m}_{c^*+1}$ (which is actually smaller than ρ_{c^*+1}). Summing the accepted loads of all contents yields the result. ■

VI. Large Catalogue Model

Keeping the many-user asymptotic, we now consider an alternative model of content catalogue, which we term the "large catalogue" scenario. The set of contents $\mathcal{C}$ is divided into a fixed number of "content classes", indexed by $i \in \mathcal{I}$. In class i, all the contents have the same popularity (arrival rate) ν_i. The number of contents within class i is assumed to scale in proportion to the number of boxes B, i.e., class i contains $\alpha_i B$ contents for some fixed scaling factor α_i. We further define $\alpha \triangleq \sum_i \alpha_i$. With the above assumptions, the system traffic load ρ in equation (6) reads

$$\rho = \frac{1}{U} \sum_{i \in \mathcal{I}} \alpha_i \nu_i. \tag{19}$$

The primary motivation for this model is mathematical convenience: by limiting the number of popularity values we limit the "dimensionality" of the request distribution, even though we now allow for a growing number of contents. It can also be justified as an approximation, that would result from batching into a single class all contents with a comparable popularity. Such classes can also capture the movie type (e.g. thriller, comedy) and age (assuming popularity decreases with content age).

We use $\hat{\upsilon}_i$ to denote the normalized popularity of content class $i \in \mathcal{I}$ and it reads $\sum_{i \in \mathcal{I}} \hat{\upsilon}_i = 1$. It is reasonable to regard each $\hat{\upsilon}_i$ as fixed. $\hat{\nu}_i \triangleq \hat{\upsilon}_i/(\alpha_i B)$ represents the normalized popularity of a specific content in class i, which decreases as the number of contents in this class $\alpha_i B$ increases, since users now have more choices within each class. In practice, an online video provider company which uses the Distributed Server Network architecture adds both boxes and available movies of each type to attract more user traffic, under a constraint of a maximum tolerable traffic load ρ.

Returning to the Distributed Server Network model of Section IV, we consider the following questions: What amount of storage is required to ensure that memory space is not a bottleneck? Is the proportional-to-product placement strategy still optimal under the large-catalogue scaling?

A. Necessity of Unbounded Storage

We first establish that bounded storage will strictly constrain utilization of bandwidth resources. To this end we need the following lemma:

Lemma 1: Consider the system under large catalogue scaling, with fixed weights α_i and cache size M per box. Define $M' \triangleq \lceil 2M/\alpha \rceil$. Then

(i) More than half of the contents are replicated at most M' times, and

(ii) For each of these contents, the loss probability is at least $E(\inf_i \nu_i, M'U) > 0$, where $E(\cdot, \cdot)$ is the Erlang function [6] defined as:

$$E(\nu, C) \triangleq \frac{\nu^C}{C!} \left[\sum_{n=1}^{C} \frac{\nu^n}{n!} \right]^{-1}.$$

◇

Proof: We first prove part (i). Note that the total number of content replicas in the system equals BM. Thus, denoting by f the fraction of contents replicated at least $M'+1$ times, it follows that $f\alpha B(M'+1) \le BM$, which in turn yields

$$f \le \frac{M}{\alpha\left(\lceil 2M/\alpha \rceil + 1\right)} \le \frac{M}{2M+\alpha} < \frac{1}{2},$$

which implies statement (i).

To prove part (ii), we establish the following general property for a loss network with call types $j \in \mathcal{J}$, corresponding arrival rates ν_j, and maximal number of competing calls C_j for type j. Denoting by $p_j(\boldsymbol{\nu})$ the loss probability with arrival rates $\boldsymbol{\nu}$, we then want to prove

$$p_j(\boldsymbol{\nu}) \ge E(\nu_j, C_j). \tag{20}$$

To show this, one constructs jointly the trajectory of a loss network under arrival rates $\boldsymbol{\nu}$ (System 1), and a loss network with only arrivals of type j calls at rate ν_j (System 2). Denoting by $X_j(t)$ and $X'_j(t)$ the number of active calls of type j respectively in Systems 1 and 2, starting with empty initial conditions, this joint construction can be performed in such a coupled way that external requests of type j occur at the same time in both systems, the monotonicity $X'_j(t) \ge X_j(t)$ holds at any time instant t, and when a downward jump (completion of a call) occurs for X_j, it also occurs for X'_j.

We further let $A(t)$ denote the number of type j external call requests, $L(t)$, $L'(t)$ the number of type j call rejections, and $D(t)$, $D'(t)$ the number of type j call completions, respectively in Systems 1 and 2, during time interval $[0,t]$. Then, the coupled construction ensures the two inequalities

$$X(t) \le X'(t), \quad D(t) \le D'(t),$$

where $X(t) = A(t) - L(t) - D(t)$ and $X'(t) = A(t) - L'(t) - D'(t)$. Combining these two inequalities yields $L(t) \ge L'(t)$. Upon dividing this inequality by $A(t)$ and letting t tend to infinity, one retrieves the announced inequality (20) by the ergodic theorem. ■

This readily implies the following corollary:

Corollary 2: Under the assumptions in Lemma 1, the overall rate of rejected requests is at least $\frac{\alpha B}{2} E(\min_i \nu_i, M'U)$, which is $\Omega(B)$ when M is bounded. Indeed, for bounded M, M' is also bounded, and $E(\min_i \nu_i, M'U)$ is bounded away from 0. ◇

Thus, even when the system load ρ is strictly less than 1, with bounded M there is a non-vanishing fraction of rejected requests, hence a suboptimal use of bandwidth.

B. Efficiency of Proportional-to-Product Placement

We consider the following **"Modified Proportional-to-Product Placement"**: Each of the M cache slots at a given box b contains a randomly chosen content. The probability of selecting one particular content c is $\nu_i/(\rho BU)$ if it belongs to class i. In addition, we assume that the selections for all such MB cache slots are done independently of one another.

Remark 3: This content placement strategy can be viewed as a "balls-and-bins" experiment. All the MB cache slots are regarded as balls, and all the $|\mathcal{C}|$ ($= \sum_i \alpha_i B$) contents are regarded as bins. We throw each of the MB balls at random among all the $|\mathcal{C}|$ bins. Bin c (corresponding to content c which belongs to class i) will be chosen with probability $\nu_i/(\rho BU)$. Alternatively, the resulting allocation can be viewed as a bipartite random graph connecting boxes to contents. $\diamond$

Note that this strategy differs from the "proportional-to-product" placement strategy proposed in Section IV, in that it allows for multiple copies of the same content at the same box. However, by the birthday paradox, we can prove the following lemma which shows that up to a negligible fraction of boxes, the above content placement does coincide with the proportional-to-product strategy.

Lemma 2: By using the above content placement strategy, at a certain box, if $M \ll \sqrt{(\min_i \alpha_i)B}$, then the probability that all the M cached contents are different is close to 1. $\diamond$

In order to use the birthday paradox to prove it, one can think of picking a content for a cache slot as a two-step process: with probability $\alpha_i\nu_i/\sum_j \alpha_j\nu_j$, a content in class i is chosen; then, conditional on class i, a specific content is chosen uniformly at random among all the $\alpha_i B$ contents in class i. The detailed proof is included in [11].

To prove that under this particular placement, inefficiency in bandwidth utilization vanishes as $M \to \infty$, we shall in fact consider a slight modification of the "request repacking" acceptance rule considered so far for determining which contents to accept:

[Counter-Based Acceptance Rule]

A parameter $L > 0$ is fixed. Each box b maintains at all times a counter Z_b of associated requests. For any content c, the following procedure is used by the server whenever a request arrives: A random set of L distinct boxes, each of which holds a replica of content c, is selected. An attempt is made to associate the newly arrived request with all L boxes, but the request will be rejected if its acceptance would lead any of the corresponding box counters to exceed LU. $\diamond$

Remark 4: In this acceptance rule, associating a request to a set of L boxes does not mean that the requested content will be downloaded from all these L boxes. As before, the download stream will only come from one of the L boxes, and we do not specify which one is to be picked.

It is readily seen that the above rule defines a loss network. Moreover, it is a stricter acceptance rule than the previous one. Indeed, it can be verified that when all ongoing requests have an associated set of L boxes, whose counters are no larger than LU, then the feasibility condition (2) holds a fortiori. $\diamond$

We introduce an additional assumption, needed for technical reasons.

Assumption 1: A content which is too poorly replicated is never served. Specifically, **a content must be replicated at least $M^{3/4}$ times to be eligible for service.** $\diamond$

Our main result in this context will be the following theorem:

Theorem 3: Consider fixed M, α_i, ν_i, and corresponding load $\rho < 1$. Then for suitable choice of parameter L, with high probability (with respect to placement) as $B \to \infty$, the loss network with the above "modified proportional-to-product placement" and "counter-based acceptance rule" admits a content rejection probability $\phi(M)$ for some function $\phi(M)$ decreasing to zero as $M \to \infty$. $\diamond$

The interpretation of this theorem is as follows: The fraction of lost service opportunities, for an underloaded system ($\rho < 1$), vanishes as M increases. Thus, while Corollary 2 showed that $M \to \infty$ is necessary for optimal performance, this theorem shows that it is also sufficient: there is no need for a minimal speed (e.g. $M \geq \log B$) to ensure that the loss rate becomes negligible.

Due to space limitation, we give a brief proof below. A more detailed proof can be found in out technical report [11].

Proof: Let N_c denote the number of replicas of content c of class i. Then, N_c admits a binomial distribution with parameters $(MB, \frac{\nu_i}{\rho BU})$. We call content c a "good" content if $|N_c - \mathbb{E}[N_c]| < M^{2/3}$, i.e., $|N_c - \frac{\nu_i M}{\rho U}| < M^{2/3}$. Using the Chernoff bound, it follows that

$$\Pr(\text{content } c \text{ is good}) \geq 1 - 2e^{-\Theta(M^{1/3})}. \quad (21)$$

Denote by X_i the number of good contents in class i. By regarding X_i as a function of independent variables $\xi_1, \ldots, \xi_{MB}$, each of which corresponds to the choice of a content to be placed in a particular cache slot at a particular box, and then applying a corollary of Azuma-Hoeffding inequality (see e.g. Section 12.5.1 in [8] or Corollary 6.4 in [4]), one obtains

$$\Pr\left(|X_i - \mathbb{E}[X_i]| \geq (MB)^{2/3}\right) \leq 2e^{-2(MB)^{1/3}}.$$

Seeing inequality (21), one can further find

$$\begin{aligned} &\Pr\left(X_i \geq \left(1 - 2e^{-\Theta(M^{1/3})}\right) \cdot \alpha_i B - (MB)^{2/3}\right) \\ \geq\ & 1 - 2e^{-2(MB)^{1/3}}, \end{aligned} \quad (22)$$

where $M \ll \sqrt{B}$ is sufficient for the lower bound on X_i shown in the above probability to be $\Theta(B)$.

Furthermore, we call a replica "good" if it is a replica of a good content, and use C_i to denote the number of good replicas of class i. We also call a box "good" if the number of good replicas of class i held by this box lies within $\frac{\alpha_i\nu_i M}{\rho U} \pm O(M^{2/3})$. Let $\mathcal{E}_i$ represent an event that the number X_i of good contents within class i satisfies

$$X_i \geq \left(1 - 2e^{-\Theta(M^{1/3})}\right) \alpha_i B - (MB)^{2/3}, \quad (23)$$

which has a probability of at least $1 - 2e^{-\Omega((MB)^{1/3})}$, according to inequality (22). Conditional on $\mathcal{E}_i$, to constitute a box, sample without replacement from the determined content replicas. Denoting the number of good replicas of class i

stored in a particular box (say, box b) by ζ_i, it can be found that conditional on $\mathcal{E}_i$, ζ_i is stochastically bounded between Bin $\left(M, \frac{\alpha_i\nu_i}{\rho U}\left(1-O(M^{-1/3}+M^{2/3}B^{-1/3})\right)\right)$ and Bin $\left(M, \frac{\alpha_i\nu_i}{\rho U}\left(1+O(M^{-1/3})\right)\right)$. Thus, for all $i \in \mathcal{I}$,

$$\begin{aligned} & \Pr\left(\left|\zeta_i - \frac{\alpha_i\nu_i M}{\rho U}\right| \geq O(M^{2/3})\right) \\ \leq\; & \Pr\left(\left|\zeta_i - \frac{\alpha_i\nu_i M}{\rho U}\right| \geq O(M^{2/3}) \middle| \mathcal{E}_i\right)\Pr(\mathcal{E}_i) + \Pr(\mathcal{E}_i^c) \\ \stackrel{(a)}{\leq}\; & 2e^{-\Theta(M^{1/3})}\cdot\Pr(\mathcal{E}_i) + (1-\Pr(\mathcal{E}_i)) \leq 2e^{-\Theta(M^{1/3})}, \end{aligned}$$

where (a) is due to the Chernoff bound (applied to the two binomial bounds on ζ_i). Then, by definition,

$$\Pr(\text{box } b \text{ is good}) \geq 1 - 2|\mathcal{I}|e^{-\Theta(M^{1/3})}.$$

Now, using a similar approach to bound the number of good boxes, say Y, as we did for the number of good contents of each class, we can find

$$\Pr\left(Y \geq B\left(1-2|\mathcal{I}|e^{-\Theta(M^{1/3})}\right)\right) \geq 1 - 2e^{-2(MB)^{1/3}}.$$

Finally, consider the performance of the loss network defined by the "Counter-Based Acceptance Rule." We introduce an auxiliary system to establish an upper bound on the rejection rate. In the auxiliary system, upon arrival of a request for content c, L different requests are mapped to L distinct boxes holding a replica of c, but here they are accepted or rejected individually rather than jointly. Letting Z_b (respectively, Z'_b) denote the number of requests associated to box b in the original (respectively, auxiliary) system, one readily sees that $Z_b \leq Z'_b$ at all times and all boxes. Furthermore, for each box b, the process Z'_b evolves as a one-dimensional loss network with arrival rate no larger than $\overline{\nu} = O(LM^{-1/12}) + \rho LU$. The explanation for the this upper bound is as follows: The requests for each non-good content a box caches have a rate of at most $O(LM^{-3/4})$, since by Assumption 1, the requests for a content are ignored if it is replicated less than $M^{3/4}$ times. Besides, there are at most $O(M^{2/3})$ non-good contents in a good box. On the other hand, the rate of requests for a good content is upper bounded by $\frac{\rho LU}{M}(1+O(M^{-1/3}))$, and there are at most $\alpha_i\nu_i M/\rho U + O(M^{2/3})$ good content replicas of class i cached in a good box.

Therefore, we can upper bound the loss probability of Z'_b by $E(\rho LU + O(LM^{-1/12}), LU)$, which can actually be further upper bounded by $e^{-\Theta(L)}$, under the assumption that $\rho < 1$.

Given some small $\epsilon \in (0, 1/3)$, it can be further shown that, with high probability, for a fraction of at least $1 - O(M^{-\epsilon})$ of good contents, each of them has at least a fraction $1 - O(M^{-1/3+\epsilon})$ of its replicas stored in good boxes. Additionally, for each of such contents, the probability of accepting a corresponding request is lower bounded by $\left(1-O(M^{-1/3+\epsilon})\right)^L \cdot \left(1-Le^{-\Theta(L)}\right)$, and such contents almost accounts for the whole content catalogue as $B, M \to \infty$. Letting $L \to \infty$ but keeping $L \ll M^{-(1/3-\epsilon)}$, this lower bound goes to 1 and hence we can conclude that the requests for almost all the contents will have near-zero loss. ■

VII. Conclusion

In peer-to-peer video-on-demand systems, the information of content popularity can be utilized to design optimal content placement strategies, which minimizes the fraction of rejected requests in the system, or equivalently, maximizes the utilization of peers' uplink bandwidth resources. We focused on P2P systems where the number of users is large. For the limited content catalogue size scenario, we proved the optimality of a proportional-to-product placement in the Distributed Server Network architecture, and proved optimality of "Hot-Warm-Cold" placement in the Pure P2P Network architecture. For the large content catalogue scenario, we also established that proportional-to-product placement leads to optimal performance in the Distributed Server Network. Many interesting questions remain. To name only two, more general popularity distributions (e.g. Zipf) for the large catalogue scenario could be investigated; the efficiency of adaptive cache update rules such as the one discussed in Section IV-A, or classical alternatives such as LRU, in conjunction with a loss network operation, also deserves more detailed analysis.

References

[1] J. M. Almeida, D. L. Eager, M. K. Vernon, and S. J. Wright. Minimizing delivery cost in scalable streaming content distribution systems. *IEEE Transactions on Multimedia*, 6(2):356–365, Apr. 2004.

[2] B. Bollobás. *Modern Graph Theory*. Springer, New York, 1998.

[3] Y. Boufkhad, F. Mathieu, F. de Montgolfier, D. Perino, and L. Viennot. Achievable catalog size in peer-to-peer video-on-demand systems. In *Proc. of IPTPS*, 2008.

[4] M. Draief and L. Massoulie. *Epidemics and rumours in complex networks*. Cambridge University Press, 2010.

[5] J. Kangasharju, K. W. Ross, and D. A. Turner. Optimizing file availability in peer-to-peer content distribution. In *Proc. of IEEE INFOCOM*, 2007.

[6] F. Kelly. Loss networks. *The Annals of Applied Probability*, 1(3):319–378, 1991.

[7] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis. On the optimization of storage capacity allocation for content distribution. *Computer Networks*, 47:409–428, 2003.

[8] M. Mitzenmacher and E. Upfal. *Probability and computing: randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[9] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello. Push-to-peer video-on-demand system: Design and evaluation. *IEEE Journal on Selected Areas in Communications*, 25(9):1706–1716, 2007.

[10] B. R. Tan and L. Massoulie. Brief announcement: Adaptive content placement for peer-to-peer video-on-demand systems. In *Proc. of ACM PODC*, Jul. 2010.

[11] B. R. Tan and L. Massoulie. Optimal content placement for peer-to-peer video-on-demand systems. *Technical Report*, Jan. 2011. (http://arxiv.org/abs/1004.4709).

[12] S. Tewari and L. Kleinrock. On fairness, optimal download performance and proportional replication in peer-to-peer networks. In *Proc. of IFIP Networking*, 2005.

[13] S. Tewari and L. Kleinrock. Proportional replication in peer-to-peer networks. In *Proc. of IEEE INFOCOM*, 2006.

[14] V. Valancius, N. Laoutaris, L. Massoulie, C. Diot, and P. Rodriguez. Greening the internet with nano data centers. In *Proc. of CoNEXT)*, pages 37–48, 2009.

[15] J. Wu and B. Li. Keep cache replacement simple in peer-assisted VoD systems. In *Proc. of IEEE INFOCOM Mini-Conference*, 2009.

[16] X. Zhou and C.-Z. Xu. Efficient algorithms of video replication and placement on a cluster of streaming servers. *Journal of Network and Computer Applications*, 30(2):515–540, Apr. 2007.