Analysis of Peer-Assisted Video-on-Demand Systems with Scalable Video Streams

Kianoosh Mokhtarian
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

Mohamed Hefeeda
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

ABSTRACT

In recent years, peer-to-peer (P2P) and peer-assisted streaming have emerged as promising models for low-cost multimedia distribution to large scale user communities. In this paper, we study streaming of *scalable video streams* over these systems. Scalable video streams are composed of multiple layers and can easily be adapted according to the characteristics and needs of receivers. Thus, they can efficiently support a wide spectrum of heterogeneous peers participating in a P2P streaming system. We present an analytical model for forecasting the long-term behavior of a P2P streaming system with scalable video streams. Our analysis takes as inputs the characteristics of a dynamic P2P streaming system and the video streams. It then analytically computes the expected throughput of the streaming system and the expected video quality delivered to peers. The analysis also provides an upper bound on the maximum number of peers that can be admitted to the system at once (i.e., in flash crowd scenarios), while ensuring a certain video quality. We present a general analysis framework that can be customized to various practical P2P streaming systems with different characteristics. Then, we show the detailed analysis of a typical P2P streaming system and we explain how other systems can be analyzed using our model. We validate our analysis by comparing its results to those obtained from simulations, which confirm the accuracy of our analysis. Our analysis and simulations enable administrators of P2P streaming systems to predict the throughput and the video quality that can be delivered to users.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: [modeling techniques];
C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*distributed applications*

General Terms

Performance

Keywords

1. INTRODUCTION

The demand for multimedia services has seen a rapid growth in the past few years, which is even expected to accelerate in near future as confirmed by many market research reports [4, 16]. To meet portions of this demand, peer-assisted and peer-to-peer streaming systems have been successfully designed and deployed for large-scale user communities [1–3,5,19]. In these systems, peers contribute bandwidth and storage to serve other peers. However, the contributions from the peers are often less than the capacity needed to serve high quality video streams. To make up for the shortage in capacity, a number of dedicated servers are usually used in these streaming systems, which we refer to as seed servers. Because of the existence of seed servers, these streaming systems are sometimes called peer-assisted systems. We use the terms peer-assisted and P2P streaming systems interchangeably in this paper.

In current P2P streaming systems, a video is encoded at a certain bitrate, typically ranging from 300 kbps to 1 Mbps [6]. To support a wider range of receivers, a lower-bitrate video is preferred, but it provides a low quality for everyone. This problem may be solved by encoding and distributing multiple versions of the video, which is called *simulcasting*. However, a video has to be encoded many times for different combinations of decoding capabilities, connection bandwidths, and viewing resolutions. Moreover, switching among versions is not easy, because: (i) for every switching, a client has to wait for the next Intra-coded frame (I-frame) of the new version, and (ii) streams of different versions could be asynchronous [9]. In addition, P2P streaming with multiple video versions divides users to separate networks which may result in reduced connectivity and less efficient utilization of peers' resources. As an alternative, Multiple Description Coding (MDC) can encode a video into multiple descriptions, where the quality of the video will be proportional to the number of descriptions received. However, MDC techniques have considerable bitrate overhead and are computationally complex [9].

In contrast, a *scalable* video stream can be encoded once and a wide range of heterogenous clients can benefit from the video. In addition, heterogeneous clients receiving different layers can still share common layers and participate in the same overlay network, leading to a larger pool of re-

sources. Moreover, scalable coding has a lower overhead and is simpler than MDC [9]. Recent scalable video coding techniques, e.g., H.264/SVC [14], have further improved this coding efficiency and significantly outperformed previous scalable videos [18]. Accordingly, they have received an increasing adoption in practice [7, 13].

In this paper, we study streaming of scalable videos over P2P networks. We develop an analytical model to forecast the behavior of a given P2P system for streaming scalable videos. Our analysis takes as input the characteristics of the network, including the distribution of upload and download bandwidths of peers, peer arrival and failure rates, and video bitrates. In addition, the analysis takes into account the serving capacity of auxiliary seed servers in the system, which need to be deployed to complement the limited upload bandwidth of peers. For example, a home user with DSL or cable connection typically has an upload bandwidth of a few hundred kilobits per second, whereas an average-to-good quality video stream requires 1–2 Mbps. Taking the characteristics of the P2P network and the capacity of auxiliary seed servers as inputs, our goal is to answer the following questions:

Q1. What is the expected throughput of a given P2P streaming system in the long term? The throughput directly impacts the video quality delivered to peers.

Q2. To increase the throughput to a desired level, how much capacity we need to provide in terms of auxiliary seed servers?

Q3. How many peers a system can support in case of a flash crowd arrival of peers?

We present an analytical model to answer the above important questions. Our model is general and can be applied to P2P streaming systems with different characteristics and network conditions. We present numerical analysis of example systems and derive various insights on their performance. In addition, we validate our analytical results through extensive simulations. Our simulation results confirm the accuracy and usefulness of our analytical model.

This paper is organized as follows. Related works are summarized in Section 2. In Section 3, the considered P2P streaming model is described. Our analytical model is presented in Section 4. In Section 5, we conduct an analysis using this model for a sample network. Then, we validate our analysis using simulations, and we use the analysis to study the performance of a P2P streaming system in Section 6. Finally, we conclude the paper in Section 7.

2. RELATED WORK

While the problem of peer-to-peer or peer-assisted video streaming has been studied by numerous previous works from various angles, such as overlay construction, distribution of data availability information, and piece scheduling algorithms, the problem of theoretically analyzing the behavior of P2P streaming systems has received less attention. Current analytical works addressing this problem assume that the video streams are encoded in nonscalable manner. Nonscalable video streams have fixed bitrates and cannot be adapted easily. For example, Tu et al. [17] present an analytical framework for studying the performance of P2P streaming systems. This framework mathematically analyzes the

pattern of capacity growth of a given dynamic P2P system over time. It then estimates the time at which the capacity of the peer population suffices for continuing the operation of the network, which is actually the time at which the servers can be turned off. A similar analysis is done in [8] based on a stochastic fluid model, which finds the maximum streaming bitrate that can be supported in a P2P system. However, the works in [8, 17] only consider nonscalable streams with a constant bitrate. They also divide peers into a number of classes based on their bandwidth contributions. In addition, in [8] only live streaming scenarios are considered. The work in [17] can analyze on-demand streaming scenarios as well, but the time granularity of this analysis, i.e., the length of the smallest time unit, is the length of a complete streaming session. In our analytical model, we consider on-demand streaming of scalable video streams with variable bitrates, and we do not make any restrictive assumptions on the bandwidth of peers. Moreover, we analyze the capacity of the system over time with a fine granularity, which is the length of a video segment of a few minutes at most. This enables us to capture the dynamics of the network more accurately. For example, the time a peer stays in the network for seeding is often a fraction of the video length, which cannot be captured when we analyze the system based on a time unit equal to a complete video length.

Liu et al. [10] analyze peer-assisted streaming systems by constructing an appropriate tree structure for the network and provisioning resources for it. In contrast, we do not assume any particular structure imposed on the network formed by peers. That is, each peer can decide its neighbor list, based on the information it receives from trackers and/or other peers. Thus, our approach can be used to analyze mesh-based and receiver-driven systems, which have been repeatedly reported to be more efficient and robust than tree-based approaches. Since mesh-based systems are employed by most of today's P2P streaming systems [1–3, 6, 19], our analytical model is more practical and useful than previous ones. Another performance analysis on P2P streaming systems is conducted by Small et al. in [15]. However, similar to [10], the authors consider constructing a specific P2P topology for the overlay network, unlike our analysis which does not make any particular assumption on the network topology formed by peers.

Zhou et al. [20] and Parvez et al. [12] analyze the performance of P2P streaming systems with nonscalable videos when two different chunk selection strategies are employed by peers. These analyses are motivated by the observation that a poor streaming performance, particularly in terms of startup delay, is experienced when we adopt BitTorrent-like P2P downloads for streaming. In BitTorrent, a rarest-first strategy is often employed by peers to select which data chunks to download, which diversifies the chunks available at peers. This practice improves the overall download throughput by avoiding situations where many peers cannot find or they compete for a rare chunk. However, it is not appropriate for streaming, since streaming requires a peer to have pieces sequentially from the beginning of the file. Sequential download of pieces, on the other hand, is appropriate for streaming, but is not efficient for BitTorrent-like downloads. The authors of [20] analyze the performance of P2P streaming systems when these two strategies, rarest-first and sequential, are employed by peers. They then propose a hybrid strategy to be employed by peers for P2P live

streaming, which reduces the significant startup delay—the reduced delay can still be considerable when we are to ensure a high probability for continuous playback under this download model. The authors of [12] present a similar analysis, which is more general than [20] and consider on-demand streaming as well. However, the work in [12] also assumes that all peers have equal bandwidths and all peer-to-peer connections have the same throughput, i.e., no peer heterogeneity. Nevertheless, the argument that sequential download reduces the system throughput relies on the assumption that peers select their partners randomly. However, this assumption is no longer correct if peers request and serve data to each other more carefully to balance the load between the supply and the demand of peers. For example, each peer requests data from other peers that are closer to it in terms of playback time, and a sending peer always gives priority to requests coming from closer peers. This example is consistent with the P2P streaming models assumed in [12,20], but it makes the rarest-first and the sequential strategy equal in terms of the overall download throughput. In our analysis, we assume a sequential download strategy for peers.

3. SYSTEM OVERVIEW AND NOTATIONS

The considered P2P streaming architecture consists of trackers, seed servers, and peers. Peers join the system by contacting one of the trackers. The tracker receives periodic update reports from peers, informing it about their available data and capacity. A number of seed servers exist to inject the content into the network and to serve requests when there is not enough capacity in the peer population. Peers download the video in a streaming form, meaning that the video is downloaded sequentially at a bitrate equal to the bitrate of the video.

In the considered P2P system, peers are expected to use their upload bandwidth for serving lower layers of the videos first, in order to avoid the situation that some peers are starving (for not receiving lower layers) while other peers are receiving the highest quality. Moreover, peers serve as many layers as they can upload. For example, if all layers have a rate of 100 kbps and a peer has 250 kbps upload bandwidth, it will upload the two lowest layers at rate 100 kbps and the third one at 50 kbps.

We divide the video file into S small segments. Let P_s denote the set of peers that are currently watching the video at segment s where $1 \leq s \leq S$. A peer can only serve data to peers that are at earlier segments. In other words, a peer in P_s can be served by any peer in $P_{s'}$, $s < s' \leq S$, given that the requested layer is available at the sending peer. Peers in P_s cannot be served by any peer in $P_{s'}$ where $1 \leq s' \leq s$.

In addition to receiving from other peers, a peer may request to receive the stream from seed servers. Seed servers are often loaded with a volume of requests larger than their serving capacity. Thus, it is important to efficiently manage this capacity and carefully choose which requests to serve. In our previous work [11], we analyzed two methods for serving peers by seed servers and matching them to other peers: **random** and **contribution-based**. In the former case, peers are matched together and/or served by seed servers randomly. In the latter case, this task is done based on the contribution of peers and the bitrates/qualities of video substreams. We have shown how to use the contribution-based method to gain an overall benefit approaching the optimal benefit that can be gained in the system. In this paper, we

Table 1: List of symbols used in this paper.

Symbol	Description
C	Total capacity of seed servers.
L	Number of layers of the video.
T	Length of the video in number of segments.
T_{seg}	Video segment length (in seconds).
T_{seed}	The time each peer stays in the network for seeding (after finished watching).
$e[l]$	The capacity served for layer l to peers at a given video segment.
$f(e, l, r_l, n)$	The capacity that can be served for layer l (at rate r_l) by n peers demanding this layer, when an input capacity e is served to them.
$o[l]$	The capacity served for layer l by peers at a given video segment.
$q_l(t)$	Video quality (e.g., Y-PSNR) added by the l-th layer of the video at segment t.
$r_l(t)$	Bitrate of the l-th layer ($1 \leq l \leq L$) of the video at segment t.
$\gamma(t)$	The probability that a peer that is at segment t fails/leaves.
$\mathbf{D}, \mathbf{U}$	Random variables representing the download and upload bandwidth of peers. $\mathbf{D}$ and $\mathbf{U}$ are not independent.
$\mathbf{\Lambda}$	Random variable representing the number of arrivals in a time unit.

present our analysis for both random and contribution-based methods.

The notations that we use in the analysis are summarized in Table 1. Bold symbols represent random variables.

4. GENERAL ANALYTICAL MODEL

In this section, we present the proposed model for analyzing P2P streaming systems. We first present a high level overview of the analysis, followed by step-by-step details. We derive general formulas which can yield the throughput of the P2P system as a function of the characteristics of the network. These general formulas can be readily customized for a given set of network characteristics, and allow us to capture a wide range of network specifications. This mainly consists of replacing the general variables used in this section, e.g., random variables representing peer bandwidth distributions, by values specific to the P2P system being analyzed. In the next section we present an example of this customization.

4.1 Overview

Our analysis takes the following variables as inputs, and it outputs the throughput of the network in various forms, such as the average video quality delivered to peers.

- The joint distribution of upload and download bandwidth of peers. We define random variables $\mathbf{D}$ and $\mathbf{U}$ as the download and upload bandwidth, which are not independent; they should be considered jointly when analyzing the capacity of peers for uploading video layers. Let $Pr(x_1 \leq \mathbf{D} \leq x_2, y_1 \leq \mathbf{U} \leq y_2)$ where $x_1, x_2, y_1, y_2 \in \mathbb{R}$ denote the probability that a peer's download and upload bandwidth values are in the range $[x_1, x_2]$ and $[y_1, y_2]$, respectively.

- The expected peer seeding duration, T_{seed}.

- The seed servers capacity C, number of layers L, and bitrate of each video layer $r_l(t)$ at each segment t.

- The distribution of peer arrivals. $\mathbf{N}$ is the random variable representing the number of arrivals per segment and $Pr(\mathbf{N} = n)$ denotes the probability of having n arrivals in T_{seg} seconds.

- The peer failure rate, $\gamma(t)$, denoting the probability that a peer at segment t leaves the network; $\gamma(t)$ is a function of t to capture realistic peer behaviors, e.g., peers that have stayed longer are less likely to leave.

At a high level, the analysis proceeds as follows. Recall that a peer in P_s can stream to peers in $P_{s'}(s' < s)$ only. Accordingly, starting from the last segment of the video to the first $(s = S, S - 1, \ldots, 1)$, in each step of the analysis we consider the set of peers that are watching the same video segment s, i.e., peers in P_s. We estimate the output capacity of these peers as a function of the capacity served to them. Using this function, we can compute the capacity that is going to be served to these peers and the capacity that they will serve to peers in $P_1, \ldots, P_{s-1}$. We then proceed to segment $s - 1$ and so on. In our analysis, we treat capacity values as arrays rather than single numbers, where $e[1], \ldots, e[L]$ represent the capacity served for layers $1, \ldots, L$ to P_s, and $o[1], \ldots, o[L]$ represent the capacities served by P_s. We will later see how to divide a single capacity value C between $e[l]$ values for the layers.

To analyze the set of peers at a given segment, we define the function $o = f(e, l, r_l, n)$ as the output capacity of peers for layer l when this layer is served to them, i.e., n peers demanding the layer at the segment, with an input capacity e. The bitrate of the video layer at that segment is r_l.

In the remainder of this section, we will first obtain the function $f(\cdot)$ in Section 4.2. This function, as just described, captures the distribution of only one video segment to a given number of peers. We employ this function in Section 4.3 to analyze the distribution of the complete video in a dynamic network.

4.2 Distribution of one Video Segment

We define function $o = f(e, l, r_l, n)$ as the output capacity for layer l when serving this layer with a capacity e to n peers demanding it at a segment. r_l is the bitrate of the layer at the segment. Clearly, if e is equal to or larger than the demand of n peers, i.e., $e \geq r \times n$, all the n peers can be served using e. On the other hand, if $e < r \times n$, we can serve only a subset of peers. The estimated output capacity o depends on how these subsets are chosen. As discussed in Section 3, we consider two cases for this purpose: (i) when the serving is done by giving preference to peers with higher contribution, and (ii) when the serving is done randomly, as in many current P2P streaming systems. We distinguish three different $f(e, l, r, n)$ functions: $f_{opt}(\cdot)$ for case (i), $f_{rnd}(\cdot)$ for case (ii), and $f_{all}(\cdot)$ for the case where $e \geq r_l \times n$, in which it does not matter how to serve peers since the demand is smaller than the input capacity.

Among these three forms, we first obtain $o = f_{opt}(e, l, r_l, n)$. For this case, peers are sorted based on the bitrate that they can stream layer l to other peers, and are selected one by one for being served with layer l. The bitrate $u_{p,l}$ at which a peer

p streams layer l to other peers is estimated according the system model in Section 3. This model describes how peers are expected to share their upload bandwidth for serving different layers, i.e., lower layers first, and as many layers as possible. Thus, $u_{p,l}$ can be calculated as follows:

$$u_{p,l} = \min \left\{ u_p - \sum_{i=1}^{l-1} u_{p,i}, \ r_l \right\} \ 1 \leq l \leq L_p, \qquad (1)$$

where L_p is the number of layers that peer p can download, and u_p is its total upload bandwidth. If the upload bandwidth of p is higher than the total bitrate of the demanded layers, the bandwidth remained from Eq. (1) is equally divided among layers:

$$u_{p,l} = \min \left\{ u_p - \sum_{i=1}^{l-1} u_{p,i}, \ r_l \right\} + \frac{u_p - \sum_{i=1}^{L_p} r_i}{L_p} \qquad (2)$$

Using Eqs. (1) and (2) and the distribution of upload bandwidths ($\mathbf{U}$), we can calculate the distribution of $\mathbf{U}_l$: the random variable representing the upload rate of peers for layer l.

Peers with higher $u_{p,l}$ values are selected for being served with layer l. The number of peers that can be served using capacity e is $k = \lfloor e/r \rfloor$. Since for calculating $f_{opt}(\cdot)$, we know that $k < n$, the value to be calculated, o, consists of "the total capacity that can be streamed by the k peers that have the highest $\mathbf{U}_l$-capacity values among the total n peers". We estimate this capacity by a probabilistic analysis as follows. Assume that in a list of peers sorted according to the serving capacity for layer l, the $k + 1$-st peer has a $u_{p,l}$ value equal to x. Thus, the first k peers all have a $u_{p,l}$ value of $u_{p,l} \geq x$. Using this, we can obtain the expected value for the total capacity of the first k peers for layer l as:

$$E[k \times \mathbf{U}_l \mid \mathbf{U}_l > x, \mathbf{D} \geq R_l] =$$
$$k \times \frac{\int_x^\infty Pr(\mathbf{U}_l = y, \mathbf{D} \geq R_l) \ y \ dy}{Pr(\mathbf{U}_l \geq x, \mathbf{D} \geq R_l)}, \qquad (3)$$

where R_l is the cumulative bitrate of the first l layers. The two conditions considered in this expected value capture the fact that the involved peers in this calculation already have an upload bandwidth higher than the $k + 1$-st peer, and a download bandwidth that can afford the first l layers; clearly, peers that are not capable of downloading the first l layers cannot be an uploader for that layer.

To obtain the expected capacity of the first k peers, we need to integrate Eq. (3) with respect to x, and consider all possible combinations of peers in an arbitrary sorted list of peers. For the candidate $k + 1$-st peer that we considered in Eq. (3), there are $\binom{n}{1}$ choices. Then, there are $\binom{n-1}{k}$ choices for selecting the first k peers among the remaining ones. Hence, the expected value of $f_{opt}(e, l, r_l, n)$ is calculated as:

$$f_{opt}(e, l, r_l, n) = \int_0^\infty \binom{n}{1} \binom{n-1}{k} \underbrace{Pr(\mathbf{U}_l = x \mid \mathbf{D} \geq R_l)}_{\text{I}} \times$$
$$\underbrace{Pr(\mathbf{U}_l > x \mid \mathbf{D} \geq R_l)^k}_{\text{II}} \times \underbrace{Pr(\mathbf{U}_l < x \mid \mathbf{D} \geq R_l)^{n-k-1}}_{\text{III}} \times$$
$$\underbrace{E[k \times \mathbf{U}_l \mid \mathbf{U}_l > x, \mathbf{D} \geq R_l]}_{\text{Eq. (3)}} \ dx, \qquad (4)$$

where term I is the probability that the candidate $k + 1$-st peer has a capacity of x for serving layer l, term II is the

probability that there are k peers with higher capacity than x, and term III is the probability that there are $n - k - 1$ peers with lower such capacity. The product of these terms indicates the probability that the peer we assumed as the $k + 1$-st peer is actually the $k + 1$-st among a total of n.

Note that, rather than the expected value of o, one might argue that we should target the complete distribution of o since it could lead to more detailed results. For that distribution, however, we need to calculate the sum of k distributions of $\mathbf{U}_l$. In other words, Eq. (4) changes to:

$$Pr(o \geq x_0) = \int_0^\infty \binom{n}{1}\binom{n-1}{k} Pr(\mathbf{U}_l = x \mid \mathbf{D} \geq R_l) \times$$

$$Pr(\mathbf{U}_l > x \mid \mathbf{D} \geq R_l)^k \times Pr(\mathbf{U}_l < x \mid \mathbf{D} \geq R_l)^{n-k-1} \times$$

$$\underbrace{Pr(k \times \mathbf{U}_l \geq x_0 \mid \mathbf{U}_l > x, \mathbf{D} \geq R_l)}_{\text{I}} \, dx, \tag{5}$$

where term I denotes the sum of k independent values that follow the distribution of $\mathbf{U}_l$. Calculation of Eq. (5) requires convolving the distribution of $\mathbf{U}_l$ with itself k times, which is not feasible even for uniform distributions, the simplest ones. Specifically, in Eq. (4) we took advantage of the interesting property of additivity of expected values, i.e., $E[\mathbf{X} + \mathbf{Y}] = E[\mathbf{X}] + E[\mathbf{Y}]$, and in particular, $E[k \times \mathbf{X}] = k \times E[\mathbf{X}]$, which cannot be benefited from in Eq. (5).

We now consider the case where peers are served randomly. In this case we have:

$$f_{rnd}(e, l, r_l, n) = E[k \times \mathbf{U}_l \mid \mathbf{D} \geq R_l] = k \times E[\mathbf{U}_l \mid \mathbf{D} \geq R_l]$$

$$= k \times \frac{\int_0^\infty Pr(\mathbf{U}_l = x, \mathbf{D} \geq R_l) \, x \, dx}{Pr(\mathbf{D} \geq R_l)}, \tag{6}$$

Finally, we note that if the number of requesting peers is less than or equal to the number of peers that can be served, i.e., $k \geq n$, we have:

$$f_{all}(e, l, r_l, n) = n \times E[\mathbf{U}_l \mid \mathbf{D} \geq R_l], \tag{7}$$

since we can simply serve all peers that are at the segment.

4.3 Distribution of the Complete Video

We now use the formulas obtained for $f_{opt}(\cdot)$, $f_{rnd}(\cdot)$, and $f_{all}(\cdot)$ to calculate the capacity of the whole network. We first extend these functions to capture the dynamics of the network including arrivals and failures at one segment. We then employ these functions to generalize our analysis from the distribution of one segment to that of the whole video.

Peers dynamics. To accommodate peer arrivals and departures (or failures) that may happen at different time instances, we define $o = \hat{f}(e, l, r_l, \mathbf{N}, \epsilon)$ as the output capacity for serving a segment where the number of peers at the segment is represented by the random variable $\mathbf{N}$ and the failure probability for peers at the segment is ϵ. $\hat{f}(e, l, r_l, \mathbf{N}, \epsilon)$ is calculated as follows. Let $\mathbf{N}_l$ be a random variable representing the number of peers at a segment demanding layer l. The probability distribution of $\mathbf{N}_l$, given that the number of arrivals is $\mathbf{N}$, is given as:

$$Pr(\mathbf{N}_l = n \mid \mathbf{N}) =$$

$$\sum_{i=n}^\infty Pr(\text{there are } i \text{ peers}) \times Pr(n \text{ of the } i \text{ demand } l) =$$

$$\sum_{i=n}^\infty Pr(\mathbf{N} = i)\binom{i}{n} Pr(\mathbf{D} \geq R_l)^n Pr(\mathbf{D} < R_l)^{i-n}. \tag{8}$$

Accordingly, we calculate $\hat{f}(e, l, r_l, \mathbf{N}, \epsilon)$ as:

$$\hat{f}(e, l, r_l, \mathbf{N}, \epsilon) = \sum_{n=0}^k Pr(\mathbf{N}_l = n) \times f_{all}(e, l, r_l, n) \times (1 - \epsilon) +$$

$$\sum_{n=k+1}^\infty Pr(\mathbf{N}_l = n) \times f_{opt|rnd}(e, l, r_l, n) \times (1 - \epsilon). \tag{9}$$

Distribution of video segments over time. Consider the steady state of the network, where some peers are watching the video at segments $s = 1, \ldots, S$ and some peers have finished watching and are seeding for up to T_{seed} time units. We assume that the throughput of the network in the steady state is smaller or equal to the demand of the peer population. Otherwise, we reduce the bandwidth or even turn off the seed servers; note that the peer population is usually not capable of supporting its demand, since peers upload capacities are typically far less than their demand. We estimate the steady-state throughput of the network for both cases of random and contribution-based peer serving.

If serving randomly, the expected capacity of k arbitrary peers at segment t for serving layer l is $k \times E[\mathbf{U}_l \mid \mathbf{D} \geq \sum_{i=1}^l r_i(t)]$, as addressed in Eq. (6). If this value is less than $k \times r_l(t)$, which is the bandwidth required to serve layer l to k peers, then layer l of segment t will eventually vanish in the network since a random peer receiving it cannot serve it completely, e.g., if two peers receive the complete layer at a given time instance then only one peer can receive it at the next time instance—partial layers are of no use as they cannot be decoded. In other words, the number of peers that can be served with layer l of segment t constantly decreases and the steady-state capacity for serving that layer converges to zero. On the other hand, if the aforementioned expected value is greater or equal to $k \times r_l(t)$, eventually all peers at t that demand layer l can receive it, even if we initially seed this layer to a few random peers only. Accordingly, using Eq. (6) we can determine the video layers that survive and continue being streamed between peers, and those that do not. Using this, the video quality that is expected to be served in the network can be readily obtained, as we do in our sample analysis in Section 5.

On the other hand, if peers are served according to their upload capacity, the fate of a particular layer is not necessarily either vanishing or being served to everybody, as it is with random peer serving. As an extreme example, if at every segment we only have 1 peer capable of serving layer l, and all other peers are incapable of that, we can still have 1 peer at each segment receiving layer l while the others receive only up to the $l - 1$-st layer. We can have more peers receiving layer l by using seed servers. For instance, we can have more than 2 peers receiving the layer if we seed only one layer bitrate, since the peer we seed to will itself serve to others a portion of the layer. We now determine the number of peers that can receive layer l in the steady state as a function of the bitrate seeded.

Recall that the number of peer arrivals is represented by the random variable $\mathbf{\Lambda}$, and the failure probability at a segment t is $\gamma(t)$. Considering peers that may have left/failed at earlier segments than t, the number of peers at segment t can be represented by the random variable $\mathbf{\Lambda}_t$ as:

$$\mathbf{\Lambda}_t = \mathbf{\Lambda} \times \left(1 - \sum_{i=0}^{t-1} \gamma(i)\right). \tag{10}$$

Once the distribution of layer l of segment t is started, at each time instance i we have:

$$x_{t,l}^i = \hat{f}\Big(\underbrace{x_{t,l}^{i-1} \times \frac{T + T_{seed}}{T} + C_{t,l}}_{\text{I}} \;,\; l \;,\; r_l(t) \;,\; \mathbf{\Lambda}_t \;,\; \gamma(t) \Big),$$

(11)

where $x_{t,l}^i$ $(i \geq 1)$ is the layer-l throughput of peers at segment t at time instance i of the system; $x_{t,l}^0$ is defined as zero. Term I in Eq. (11) represents the capacity served for layer l to peers at segment t, which consists of $C_{t,l}$, the capacity served by seed servers for layer l of segment t, and $x_{t,l}^{i-1}$, the capacity served by other peers. The coefficient $\frac{T + T_{seed}}{T}$ is to take into account the expected capacity shared by seeding peers—this coefficient has a few subtle details capturing the dynamics of the network and the probability that a peer at segment t stays for seeding, which we do not include here as they do not have a considerable effect. Note that we are given a total seed server capacity C, not $C_{t,l}$ values. The way we divide C among $C_{t,l}$ values depends on the method used for allocation of seed servers. We first divide C over all video segments according to the $\mathbf{\Lambda}_t$ value of each segment t and the bitrate/quality of the video at that segment $(r(t), q(t))$. Then, we divide the capacity given to each segment t over different layers, based on the server allocation method, the demand for the layers (according to $\mathbf{D}$), and the bitrate/quality of the layers. Since conducting our analysis for a given network consists of several iterations over Eq. (11) for different segments and layers, we perform the above step in each iteration.

We first note that $x_{t,l}^i$ values converge over time, i.e., for $i = 1, 2, \ldots$, in either case of random or contribution-based serving of peers. This is intuitively true, and is specifically because $x_{t,l}^i$ values are monotonic with i, i.e., either constantly non-increasing or constantly non-decreasing. Otherwise, for instance, we could have the case that 5 peers, either randomly selected or just the top 5 high-capacity peers, can serve 8 peers, but 8 peers (selected the same way) can serve only 7 peers, which is not possible. As $x_{t,l}^i$ is monotonic and it is finite, it converges to a specific value. We illustrate a sample of this convergence in the next section.

Denoting the converged value of $x_{t,l}^i$ simply by $x_{t,l}$, we can now determine the steady-state number of peers that are expected to receive layer l of segment t. This number, denoted by $m_{t,l}$, can be obtained as:

$$m_{t,l} = \frac{x_{t,l}}{r_l(t)}.$$

(12)

Using the $m_{t,l}$ values we can estimate the steady-state throughput of the system along a variety of metrics. For example, the total video bitrate served in the network can be calculated as:

$$\text{Bitrate served} = \sum_{t=1}^{T} \sum_{l=1}^{L} m_{t,l} \times r_l(t).$$

(13)

Similarly, the average video quality served to peers is as:

$$\text{Average quality} = \frac{1}{\sum_{t=1}^{T} \mathbf{\Lambda}_t} \sum_{t=1}^{T} \Big(\mathbf{\Lambda}_t \sum_{l=1}^{L} m_{t,l} \times q_l(t) \Big),$$

(14)

in which the use of $\mathbf{\Lambda}_t$ captures the fact that the expected number of peers at different video segments is not the same.

4.4 Analysis of Flash Crowd Scenarios

Using our estimate of the throughput of the system, we can forecast the capacity of the system for supporting sudden extensive loads such as a flash crowd. In a flash crowd event, a large number of peers arrive at the system at about the same time. Taking advantage of the flexibility of scalable streams, a system may sustain in these circumstances and support the new peers by temporarily lowering the video quality delivered in the network until the system recovers from the shock and returns to a stable state. Our analysis can determine the robustness of a P2P system against flash crowds, which we define as the maximum number of peers that can be admitted to the system at once. This maximum can be achieved when we serve only a minimum video quality to all peers, i.e., the base layer. Thus, having estimated the total throughput of the network, we can calculate the number of minimal substreams that can be served using this throughput, i.e., an upper bound on the maximum number of peers that can be served when a flash crowd arrives. We assume that a peer keeps the video file in its buffer until it finishes watching the video; this is realistic since the peer is going to seed afterwards. Thus, all peers can serve the base layer of the first few segments of the video. Hence, the maximum flash crowd size N_{FC} that can be supported is calculated as:

$$N_{FC} = \frac{1}{r_1(1)} \Big(C + \sum_{t=1}^{T} \sum_{l=1}^{L} m_{t,l} \times r_l(t) - \underbrace{\sum_{t=1}^{T} \mathbf{\Lambda}_t \times r_1(t)}_{\text{I}} \Big),$$

(15)

where term I represents the bitrate served for delivering the base layer to peers that were already in the network. Eq. (15) calculates the number of base layers that can be served to newly arrived peers—the bitrate of the base layer of the first few segments of the video is assumed to be nearly equal to that of the first segment, $r_1(1)$. Note that it is the serving of the first few segments that is most important for sustaining a flash crowd. After the first few segments, peers arrived in the flash crowd also contribute to uploading, and the throughput of the system increases accordingly.

4.5 Summary

In this section, we have presented our general analytical model for studying the behavior of P2P streaming systems. We divide a video file into S segments, and analyze the bandwidth consumed and the bandwidth contributed by peers that are watching each video segment s for $s = S, S-1, \ldots, 1$. This part of the analysis is done using Eq. (9). The preliminary functions needed for calculation of Eq. (9) are derived in its preceding equations. Then, we plug the single-segment analysis function obtained in Eq. (9) into Eq. (11), which analyzes the distribution of each video segment/layer over time. Using Eqs. (11) and (12), we can obtain the number of peers that are expected to receive each video segment/layer of the video stream in the steady state. These numbers are then used to predict the total bitrate that is going to be served in the system in the long term (Eq. (13)), the average video quality delivered to peers (Eq. (14)), the capability of the system for supporting flash crowds of peers (Eq. (15)), or others user-defined metrics for capturing the performance of the system.

5. ANALYSIS OF AN EXAMPLE SYSTEM

In this section, we show the details of employing the proposed analytical model for analyzing a sample P2P streaming system. This is to demonstrate how our general analysis in Section 4 can be used in practice. P2P streaming systems with different characteristics than considered in this section can be analyzed in a similar manner.

5.1 Analysis

Analysis Setup. The P2P streaming network we analyze is specified as follows. Suppose we are distributing a video file consisting of L layers, each at a fixed rate of r kbps. The video length is S segments. We consider both random and contribution-based serving of peers. The download bandwidth of peers is uniformly distributed in $[0, M]$ kbps, and the upload bandwidth of each peer a constant fraction α of its download bandwidth, which makes it uniformly distributed in $[0, \alpha M]$ kbps. Peers arrive according to a Poisson distribution with an average arrival rate of λ, and stay after they finish watching the video for seeding for up to T_{seed} time units. A fraction of peers leave the system before watching the complete video and seeding. These departures happen with probability $\hat{\gamma}_t$ for peers at segment t ($1 \leq t \leq S$). Note that the general analysis framework presented in the previous section is not limited to any of the above assumptions, e.g., the bitrate of video layers can be different, and the bandwidth of peers can follow an arbitrary distribution.

We customize the model and equations derived in the previous section for the network described above. First, we need to calculate the functions $f_{all}(\cdot)$, $f_{rnd}(\cdot)$, and $f_{opt}(\cdot)$, which take as input the bandwidth e kbps for serving the r-kbps layer l to n peers. For this purpose, we need to obtain the closed form expression for $E[\ \mathbf{U}_l \mid \mathbf{D} \geq R_l\ ]$, the expected value of the upload capacity for layer l by a peer that can download layer l, since this value is repeatedly used in the following equations. This is done using the distribution of bandwidths as:

$$E[\ \mathbf{U}_l \mid \mathbf{D} \geq R_l\ ] = E[\ \mathbf{U}_l \mid \mathbf{U} \geq \alpha rl\ ],$$

where $E[\ \mathbf{U}_l \mid \mathbf{U} \geq z\ ]$ for any arbitrary z is calculated according to the way a peer shares its upload bandwidth among layers (Eqs. (1) and (2)) and the distribution of the upload bandwidth of peers, which is uniform in our case. More specifically, denoting by $U_l(u)$ the layer-l upload bandwidth of a peer whose total upload bandwidth is u, according to Eqs. (1) and (2) we have:

$$U_l(u) = \begin{cases} \frac{u}{L} & rL \leq u \leq \alpha M \\ r & rl \leq u < rL \\ u - r(l-1) & r(l-1) \leq u < rl \\ 0 & 0 \leq u < r(l-1), \end{cases}$$

which leads to calculation of the desired expected value $E[\ \mathbf{U}_l \mid \mathbf{U} \geq z\ ]$ as:

$$E[\ \mathbf{U}_l \mid \mathbf{U} \geq z\ ] = \frac{\int_z^\infty Pr(\mathbf{U} = u) U_l(u)\, du}{Pr(\mathbf{U} \geq z)} = \frac{\int_z^{\alpha M} U_l(u)\, du}{\alpha M - z}$$

$$= \begin{cases} \frac{1}{\alpha M - z} \int_z^{\alpha M} \frac{u}{L}\, du & rL \leq z \leq \alpha M \\[2mm] \frac{1}{\alpha M - z} \int_z^{rL} r\, du + \\ \quad E[\ \mathbf{U}_l \mid \mathbf{U} \geq rL\ ] & rl \leq z < rL \\[2mm] \frac{1}{\alpha M - z} \int_z^{rl} (u - r(l-1))\, du + \\ \quad E[\ \mathbf{U}_l \mid \mathbf{U} \geq rl\ ] & r(l-1) \leq z < rl \\[2mm] \frac{1}{\alpha M - z} \int_z^{r(l-1)} 0\, du + \\ \quad E[\ \mathbf{U}_l \mid \mathbf{U} \geq r(l-1)\ ] & 0 \leq z < r(l-1). \end{cases} \quad (16)$$

The first row in Eq. (16) corresponds to the case where the upload bandwidth of a peer is higher than the total bitrate of video layers, in which a bandwidth u is evenly divided among the L layers. The second row adds to this case the possibility that the upload bandwidth is less than the bitrate of all layers, but higher than rl, the bitrate of the first l layers. In this case, the bitrate at which layer l is streamed to other peers is just equal to r. The third row adds the case that the peer's upload bandwidth is less than the bitrate of the first l layers, but higher than the first $l-1$ of them. In this case, it can upload layer l at a rate less than r. Finally, the last row considers the case where the peer cannot upload layer l at all.

Now that we obtained $E[\ \mathbf{U}_l \mid \mathbf{U} \geq z\ ]$ for any arbitrary z, we can calculate functions $f_{all}(\cdot)$ and $f_{rnd}(\cdot)$ using Eqs. (6) and (7):

$$f_{all}(e, l, r, n) = n \times E[\ \mathbf{U}_l \mid \mathbf{U} \geq \alpha rl\ ]$$
$$f_{rnd}(e, l, r, n) = \lfloor \frac{e}{r} \rfloor \times E[\ \mathbf{U}_l \mid \mathbf{U} \geq \alpha rl\ ]$$

Next, we calculate $f_{opt}(\cdot)$ using Eq. (4):

$$f_{opt}(e, l, r_l, n) = \int_0^{\alpha M} Pr(\mathbf{U} = u \mid \mathbf{U} \geq \alpha rl) \times$$
$$Pr(\mathbf{U} \geq u \mid \mathbf{U} \geq \alpha rl)^k \times$$
$$Pr(\mathbf{U} < u \mid \mathbf{U} \geq \alpha rl)^{n-k-1} \times$$
$$E[\mathbf{U}_l \mid \mathbf{U} \geq \alpha rl, \mathbf{U} \geq u]\, du, \quad (17)$$

in which all probabilities can be easily derived according to the uniform distribution of $\mathbf{U}$ in $[0, \alpha M]$.

The next function we need to calculate is $Pr(\mathbf{N}_l = n \mid \epsilon)$, the probability distribution of the number of peers at a segment which are demanding layer l, when the failure probability for peers at that segment is ϵ. This is done using Eq. (8) as follows:

$$Pr(\mathbf{N}_l = n \mid \epsilon) =$$
$$\sum_{i=n}^{\infty} f_{\text{Poisson}}(\lceil \frac{i}{1-\epsilon} \rceil; \lambda) \binom{i}{n} (\frac{\alpha M - rl}{\alpha M})^n (\frac{rl}{\alpha M})^{i-n} \quad (18)$$

We can now analyze the distribution of each video segment through Eqs. (9) and (11), since we have already obtained all its preliminaries in the above equations. Thus, we have:

$$x_{t,l}^i = \hat{f}(\ x_{t,l}^{i-1} \times \frac{T + T_{seed}}{T} + C_{t,l}, l, r, \underbrace{\lambda(1 - \sum_{i=1}^{t-1} \hat{\gamma}_i)}_{\text{I}}, \hat{\gamma}_t\ ), \quad (19)$$

where term I denotes the expected number of peers (with Poisson distribution) at segment t, i.e., peers whose expected number was λ at the beginning of their joining, and left the
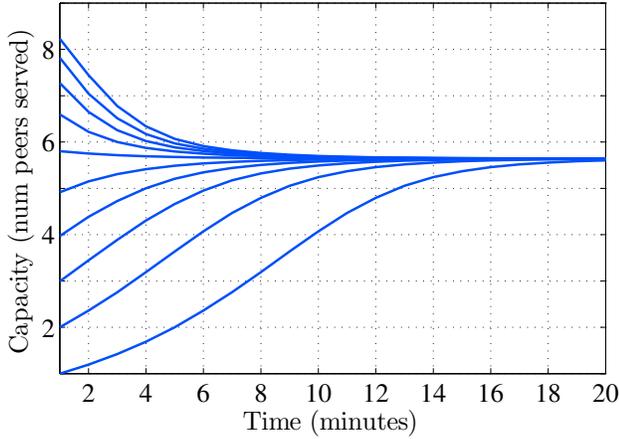
(a) The streaming capacity when we only seed at the first time step. The curves from bottom to top correspond to initial seed values from 1 layer rate to 10 layer rates.
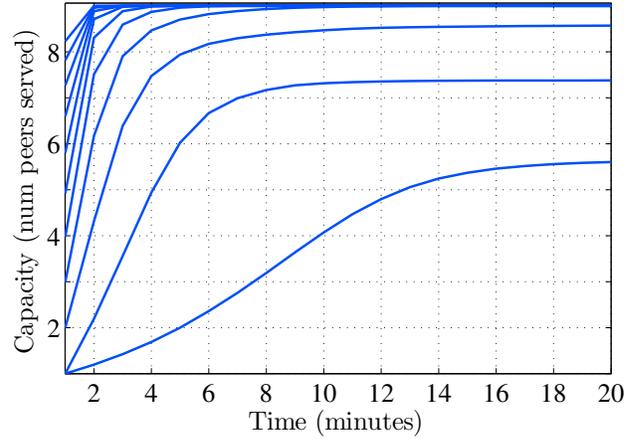
(b) The streaming capacity when we seed for longer periods. The bottom curve is for when we seed with 1 layer bitrate at the first time step only, and the other 10 curves correspond to longer seeding periods with capacities 1 to 10 layer bitrates.

Figure 1: The streaming capacity of peers at one segment for one layer.

system with probability $\hat{\gamma}_i$ at each time instance i. Denoting the value converged over time by $x_{t,l}$, the steady-state number of peers receiving layer l of segment t, i.e., $m_{t,l}$, is calculated using Eq. (12).

5.2 Numerical Results (Answering Q_1 and Q_2)

Since each video segment is analyzed individually, let us first focus on the distribution of one segment. We would like to see how many peers at a given segment can be served with a particular layer l in the long term. We then use these results as a basis for analyzing the distribution of the complete video. To carry out a numerical analysis, we fix the value of some parameters introduced in the previous subsection as follows. We suppose the distribution of a 1-hour video which we divide into 60 1-minute segments. The video has $L = 10$ layers, each at rate $r = 200$ kbps. The maximum possible download bandwidth of a peer is $M = 10$ Mbps and the upload bandwidth is $\alpha = 20\%$ of its download bandwidth. Peers arrive at a rate of $\lambda = 10$ peers per minute with a Poisson distribution, and after watching the video, they stay for seeding for up to $T_{seed} = 30$ minutes. 25% of peers leave the system before they finish watching and seeding, which happens with an exponential probability distribution with most of the departures taking place in the first few minutes of watching.

Step 1: Distribution of one video segment. We first note that the number of peers served with layer l of a given segment t converges over time, as discussed in Section 4.3. This can be observed in Figure 1(a), which shows a sample result. In this figure, considering a window of peers at one segment over time, the number of peers that are served with layer 5 in each time step is plotted. Each curve, from bottom to top, corresponds to an initial seed capacity, starting from 1 layer bitrate and increasing to 10. The failure probability for peers at the segment is assumed 20%. In Figure 1(a), independent of the initial seed, the number of peers served with layer 5 of the segment converges to 5.6 on average, though it takes more time to boost the capacity when the initial seed

is small while larger initial seeds lead to faster convergence. If we do not stop seeding after the first time step and keep seeding over time, indeed a higher bitrate is served to the peers. This increase in the throughput for streaming layer 5 of the segment is illustrated in Figure 1(b). In this figure, the increase in the number of served peers when the seeding capacity is increased is more significant for smaller capacity values, which is particularly observable as the gap between the three lowest curves. This is because in those cases, the entire seeded bitrate is consumed by peers, and since peers will also contribute to serving, the increase in the throughput is more than the seeded bitrate. On the other hand, if the seed is more than the demand of the peers, only part of it is consumed, and the number of peers served will converge to the expected number of peers demanding the layer. This is observed in Figure 1(b) for seeding capacities of more than 3 layer bitrate. In particular, 7 out of the 11 curves gather on top of the figure. Though Figure 1(b) corresponds to the serving of layer 5 only, serving of other layers also demonstrate similar behaviors. For lower layers (1 to 4), the most bottom curve converges to higher values than 5.6, because the total upload rate of lower layers by the peers is higher. Moreover, because of the same reason, more curves gather on top of the figure. For higher layers (6 to 10), the reverse of these behaviors is observed: the most bottom curve converges to a smaller value than 5.6, and less curves gather on top of the figure.

In summary, using our analysis and given the seeding capacity, we can determine for each layer the value to which the number of served peers at a segment converges, i.e., the $m_{t,l}$ value introduced in Section 4.3, which is the steady-state number of peers served with layer l of segment t. We use these values for analyzing the distribution of the complete video in the next step.

Step 2: Deriving the throughput for the complete video. Using the previous step, we obtain $m_{t,l}$ for all valid t, l values. Consequently, we can calculate the throughput of the network in terms of various metrics. For example, we es-
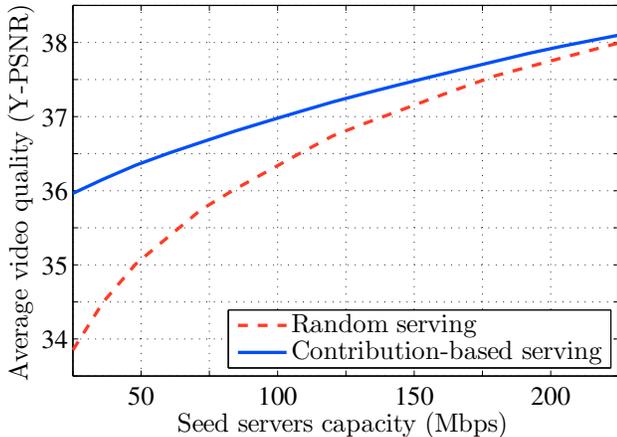
Figure 2: Validation of the analytical model. Comparing the average video quality delivered to peers for different seeding capacities.

Figure 3: Average video quality delivered to peers for different seeding capacities.

timate the average video quality delivered to peers using Eq. (14) for different values of seeding capacity C. This is plotted in Figure 2 for both cases of contribution-based and random serving. **Figure 2 answers questions Q1 and Q2 discussed in Section 1:** it estimates the throughput of the system in terms of video quality, and it computes the capacity needed for providing a desired level of video quality. For example, if we provide 25 Mbps seeding capacity, the average video quality delivered to peers will be 35.9 dB. If we increase this capacity to 125 Mbps, the average quality will increase to 37.3 dB. This increase continues as we provider higher capacities, and it eventually approaches a value of approximately 38.5 dB, which is the maximum quality demand of peers—note that although the video stream can provide a quality of up to 40 dB, not all peers are demanding this quality level.

6. EVALUATION AND VALIDATION OF THE ANALYSIS

In this section, we first validate our analytical model by comparing its results to those obtained from simulation. Then, we show how we can use the analysis for studying different performance aspects of P2P streaming systems.

6.1 Validation Using Simulation

We validate our analytical model by comparing its results to those of running the considered P2P streaming network in a simulator for P2P on-demand streaming of scalable videos.

Simulation setup. We simulate distribution of a 1-hour video which consists of $L = 10$ layers, each at rate $r = 200$ kbps. The video segment length is $T_{seg} = 1$ minute. The download bandwidth of peers is uniformly distributed in $[0, 10]$ Mbps and the upload bandwidth of a peer is $\alpha = 20\%$ of its download bandwidth. Peers arrive according to a Poisson distribution with an average arrival rate of $\lambda = 10$ peers per minute, and may stay for seeding for up to $T_{seed} = 30$ minutes after they finished watching the video. 25% of peers leave the system before they finish watching and seeding, where most of the departures take place in the first
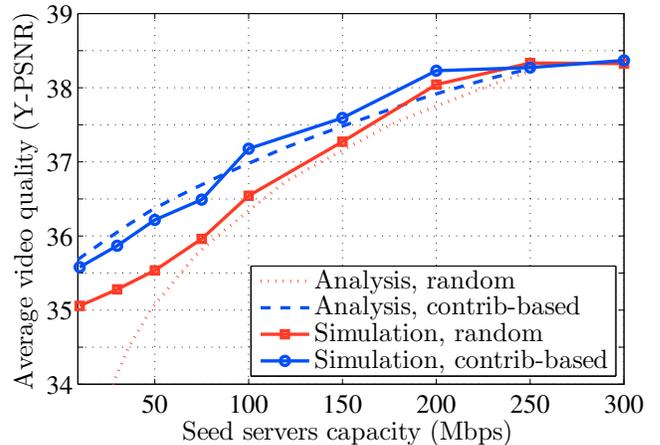
few minutes of watching. According to the arrival and failure/departure rates, the video length, and the peer seeding time, the number of peers in the network varies with time, but on average there are 500–600 at any time. Each simulation runs for 5 hours of simulation time. The simulation follows an event-driven procedure that gathers events and once every 10 seconds applies all of them. As a sender, each peer disconnects its connection to a receiver according to a Bernoulli probability distribution with an expected value of 2 minute. This is done so as to make enough dynamics in the network and to simulate peer failures in addition to the considered arrivals and departures. Moreover, we make peers disobey our assumption about the way they share their upload bandwidth among different layers in Eqs. (1) and (2), by having each $u_{p,l}$ value deviate by up to 50% from its supposed value. This is used to capture peers and network dynamics in real systems. Furthermore, a peer may receive different number of layers from different segments, which, if varies non-smoothly, results in a poor quality. Thus, to make the simulation more realistic, we avoid quality fluctuations at receiver side by having each peer take a simple heuristic that does not allow more than 1 layer change in the number of layers in two consecutive video segments, i.e., the heuristic at each peer drops the enhancement layers that violate this criterion. The video segment length and the simulation time step are 10 seconds.

We run the simulation for different values of seed server capacity, and compare the results with those of our analysis. This comparison is illustrated in Figure 3, which shows the average video quality in the steady state obtained by the analytical and the simulation study for different values of seeding capacity. Figure 3 shows that estimations of the analysis are reasonably close to the results obtained by simulations, except for small seeding capacities. The deviation observed for these values in the case of random serving is because the analytical model assumes a certain value as the upload bandwidth of a peer for a particular layer in Eqs. (1) and (2). This assumption becomes far from reality when the seeding capacity is small and it is not carefully allocated, because in this case only a few video layers are exchanged between peers and the upload bandwidths of peers for higher layers
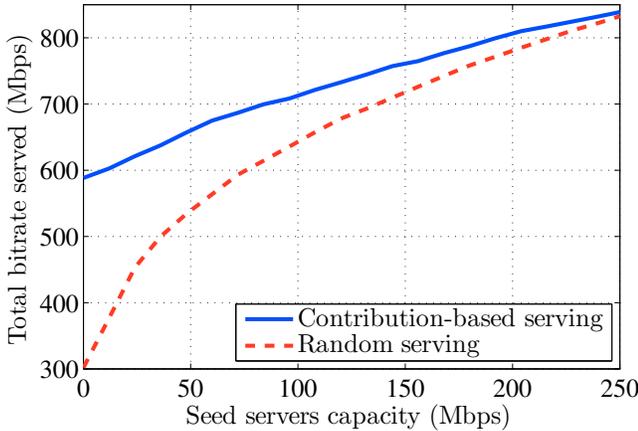
Figure 4: Total video bitrate served in the network for different methods of allocating the auxiliary seeding capacity.

Figure 5: Maximum number of peers that the system can admit when a flash crowd of peers arrives.

become significantly smaller than expected; and those for lower layers are larger than expected. Except for this particular case, the simulation explained in this section with dynamic P2P streaming systems confirms the accuracy of our analysis.

6.2 Analyzing Seed Server Allocation Methods

Using our analytical model, we compare the two methods of random and contribution-based serving of peers introduced in Section 3. We first compare these two approaches in terms of the video quality that they can deliver to peers. We compute the average delivered quality in these two cases using our analytical model, and plot the results in Figure 3. The figure demonstrates the impact of the seed server and peer matching policy. The contribution-based policy clearly outperforms the random policy. In some cases, up to 50 Mbps higher seeding capacity is needed for the random policy to provide the same quality level as the contribution-based policy. This difference is specially significant for limited seed server capacities, which is the typical case in practice. The difference diminishes for very large seeding capacities, as the capacity approaches a value that suffices for fully serving all peers with their demanded substreams.

Next, we compare the total video bitrate served in the network. This value is the sum of the bitrates of all substreams that the peers receive in the steady state. Figure 4 illustrates this value for the two cases of contribution-based and random serving. Similar to the average video quality, the gap between the bitrate served in these two cases is more significant (up to 300 Mbps) for smaller seeding capacity values. That is, careful serving with limited seeding capacity leads to a much stronger boost in the capacity of the network. Most notably, when there is no seeding, e.g., seed servers are turned off in the steady state, the total bitrate that peers can serve to each other in case of contribution-based serving is nearly twice its value with random serving. As another example, the random approach can boost a 50 Mbps seeding capacity to a throughput of 540 Mbps in the P2P system, whereas this value is 660 Mbps (22% higher) with careful serving.
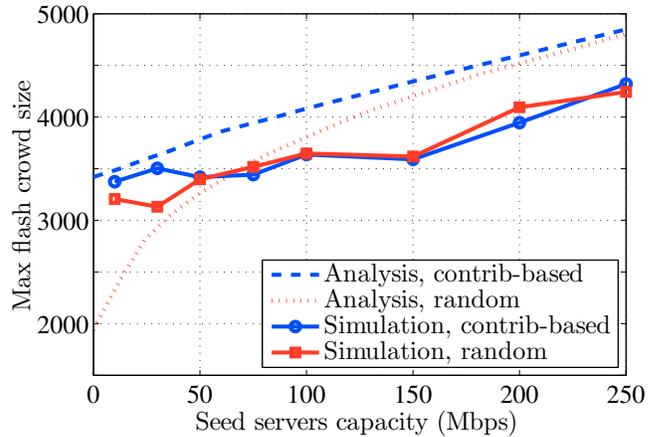
Therefore, our analysis can be used to analyze the impact of different methods for allocating seed servers and/or matching peers on the throughput of P2P systems. We presented two sample methods, serving randomly and serving towards maximizing the overall video quality delivered. Nevertheless, other methods and heuristics for prioritizing peers and their requests can as well be accommodated in the analytical model. For example, in Eq. (4) one might take into account not only the upload capacity of the requesting peer, but also the demanded substreams in order to minimize the gap between peers' received layers and their demanded layers.

6.3 Analyzing Flash Crowd Scenarios (Answering Q₃)

One of the important concerns for administrators of multimedia streaming systems is the flash crowd event, in which a large volume of peers arrive in the system at approximately the same time. This imposes an extensive load on the system and may significantly impact its performance. In such events, the system may not be able to serve all the newly arrived peers, and may have to reject many peers. In these scenarios, an important question is how many peers the system can admit in case of a flash crowd of peers.

Using the flash crowd analysis in Section 4.4, we obtain the maximum number of peers (arriving together) that can be served by the system for different values of seeding capacity. **The results of this test are plotted in Figure 5, which answer question Q3 discussed in Section 1**. In this figure, curves labeled "Analysis" show the upper bound on the flash crowd size that can be supported by the system, using the total serving capacity of peers and seed servers. Curves labeled "Simulation" depict the number of peers supported in a flash crowd in a simulated P2P system. To obtain these numbers, we make a set of 10,000 peers arrive in the system at the same time. Shortly after the arrivals, we measure the number of peers out of these 10,000 that could receive a video stream (the base layer). Note that the analytical results for the flash crowd test only provide an upper bound on the number of peers that can be served. Accordingly, there is a gap between simulation results and analysis results for this test. Moreover, although with a smaller domain, the

142

results of simulation also do not exhibit a smooth behavior. This can be attributed to the high degree of dynamics and instability introduced to the simulated system by the large crowd of peers. Nevertheless, Figure 5 shows that for smaller seed server capacities, the two serving methods make a significant difference in terms of the flash crowd size that can be supported. For example, to support 3,500 peers we need approximately 70 Mbps capacity for seed server in case of random serving, whereas we can support this number of peers with almost no extra seeding if the serving has been done carefully until the flash crowd arrives.

7. CONCLUSIONS

We have presented an analytical model to forecast the performance of P2P streaming systems with scalable videos. Given the characteristics of a P2P streaming system, our model computes the long-term throughput of the network in terms of the delivered video quality and the total served bitrate. Our analysis can also asses the benefit of deploying a given amount of seeding resources, and help determining the cost-benefit tradeoff for provisioning a higher seeding capacity to sustain a desired level of video quality. This is an important benefit of the proposed analytical model, since most current P2P streaming systems need to deploy auxiliary seed servers in order to offset the capacity shortage created by the asymmetry between peers' upload and download bandwidths. In addition, the proposed analysis can estimate an upper bound on the capability of a P2P network for supporting a flash crowd of peers, which is the maximum number of peers that can be admitted to the system at once. We have validated our analytical model using extensive simulations with realistic parameters of dynamic P2P streaming systems. The results of our simulations confirm the accuracy of the proposed model.

8. REFERENCES

[1] PPLive. http://www.pplive.com/en/index.html.

[2] SopCast. http://www.sopcast.org/.

[3] TVAnts. http://www.tvants.com/.

[4] Global IPTV market analysis (2006-2010). Technical report, RNCOS, August 2006. http://www.rncos.com/Report/IM063.htm.

[5] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. Rodriguez. Exploring VoD in P2P swarming systems. In *Proc. of IEEE INFOCOM'07*, pages 2571–2575, Anchorage, AK, May 2007.

[6] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross. A measurement study of a large-scale P2P IPTV system. *IEEE Transactions on Multimedia*, 9(8):1672–1687, December 2007.

[7] O. Hillestad, A. Perkis, V. Genc, S. Murphy, and J. Murphy. Adaptive H.264/MPEG-4 SVC video over IEEE 802.16 broadband wireless networks. In *Proc. of Packet Video Workshop (PV'07)*, pages 26–35, Lausanne, Switzerland, November 2007.

[8] R. Kumar, L. Yong, and K. Ross. Stochastic fluid theory for P2P streaming systems. In *Proc. of IEEE INFOCOM'07*, pages 919–927, Anchorage, AK, May 2007.

[9] B. Li and J. Liu. Multirate video multicast over the Internet: An overview. *IEEE Network*, 17(1):24–29, February 2003.

[10] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang. Performance bounds for peer-assisted live streaming. In *Proc. of ACM Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'08)*, pages 313–324, Annapolis, MD, June 2008.

[11] K. Mokhtarian and M. Hefeeda. Efficient allocation of seed servers in peer-to-peer streaming systems with scalable videos. In *Proc. of IEEE International Workshop on Quality of Service (IWQoS'09)*, pages 1–9, Charleston, SC, July 2009.

[12] N. Parvez, C. Williamson, A. Mahanti, and N. Carlsson. Analysis of Bittorrent-like protocols for on-demand stored media streaming. In *Proc. of ACM Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'08)*, pages 301–312, Annapolis, MD, June 2008.

[13] T. Schierl, T. Stockhammer, and T. Wiegand. Mobile video transmission using scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1204–1217, September 2007.

[14] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, September 2007.

[15] T. Small, B. Liang, and B. Li. Scaling laws and tradeoffs in peer-to-peer live multimedia streaming. In *Proc. of ACM Multimedia Conference*, pages 539 – 548, Santa Barbara, CA, October 2006.

[16] The Insight Research Corporation. Streaming media, IPTV, and broadband transport: Telecommunications carriers and entertainment services 2006-2011. http://www.insight-corp.com/execsummaries/iptv06execsum.pdf, 2006.

[17] Y. Tu, J. Sun, M. Hefeeda, Y. Xia, and S. Prabhakar. An analytical study of peer-to-peer media streaming systems. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 1(4):354–376, November 2005.

[18] M. Wien, H. Schwarz, and T. Oelbaum. Performance analysis of SVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1194–1203, September 2007.

[19] X. Zhang, J. Liu, B. Li, and T. Yum. CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. In *Proc. of IEEE INFOCOM'05*, volume 3, pages 2102–2111, Miami, FL, March 2005.

[20] Y. Zhou, D. Chiu, and J. Lui. A simple model for analyzing P2P streaming. In *IEEE International Conference on Network Protocols (ICNP'07)*, pages 226–235, Beijing, China, October 2007.