2011 19th IEEE International Conference on Network Protocols

Analysis of Adaptive Streaming for Hybrid CDN/P2P Live Video Systems

Ahmed Mansy and Mostafa Ammar
School of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332
{amansy, ammar}@cc.gatech.edu

***Abstract*—Most commercial video streaming systems rely on Content Distribution Networks (CDNs) to distribute video content. HTTP adaptive streaming has been recently adopted by major video streaming providers and is now considered the standard technique used with CDN-based streaming systems. Despite the success of these systems, cost-effective scalability continues to be of concern in their design and deployment. To address this, recent work has proposed the use of hybrid CDN and Peer-to-peer (P2P) live streaming systems. The design of these systems aims to combine the scalability of P2P systems and the desirable performance properties of CDN-based systems. However, the use of adaptive streaming, has not been explored extensively in such hybrid systems.**

Designing and operating an adaptive hybrid streaming system is very challenging. Two design decisions are very critical in the operation of any such system. The first one is the bitrate adaptation strategy which specifies how different bitrates are assigned to different users while maximizing user satisfaction. The second is defining the operational guidelines for switching the system between the CDN and the P2P modes while efficiently utilizing the available resources. In this paper we present a model and analysis of a hybrid CDN-P2P adaptive live streaming system with the objective of answering these two design questions. We first present a stochastic fluid model to the hybrid streaming system with a single video bitrate and we obtain theoretical results to guide the system operation as described above. We then extend the analysis to the adaptive streaming case with multiple video bitrates. We model adaptive streaming as a linear optimization problem to obtain the best bitrate adaptation strategy. We validate our analysis using simulations. Our conclusion is that adaptive hybrid streaming can significantly improve the ability of the system to satisfy more users with higher video bitrates over CDN-based systems.

I. INTRODUCTION

Video is widely believed to dominate traffic of the Internet. According to Cisco's visual networking index report [1], Internet video traffic slightly exceeded one third of the total traffic of the Internet in 2010 and is expected to continue its growth to reach 57% by the year 2014. This portion of the Internet traffic is mainly consumed by streaming stored and live video content and does not even include P2P video file sharing. This growth has happened in part due to the increasing availability of broadband access networks that give users high bandwidth access to the Internet which enables them to stream video with high quality. Although stored video forms the bigger portion of video content on the Internet, live video streaming is growing in volume and importance specially with important events being broadcast over the Internet [2] and new live streaming services becoming free to the public (e.g., the new live service from YouTube [3]).

Content Distribution Networks (CDNs) are currently considered the main pillar of video distribution over the Internet. The purpose of CDNs is to improve user performance in terms of delay and throughput. CDNs accomplish this by deploying multiple nodes, usually called edge servers, distributed geographically in multiple ISPs. Each edge server implements a streaming server, and when a user requests a video stream it is redirected to the closest edge server to start the desired stream. In recent years, CDN-based video streaming has evolved to provide a new adaptive streaming service where a single video can be streamed in multiple qualities at the server and a video player can choose the best quality that fits the condition of the Internet connection of the user. Although much work has been done on adaptive streaming over the years [4], [5], it was not commercially popular until the widespread adoption of *HTTP adaptive streaming* technology which is now being used by many video streaming providers (e.g., Microsoft Smooth Streaming, Netflix, Adobe).

Another important source of video traffic on the Internet is Peer-to-Peer (P2P) networks. Some P2P video systems have recently succeeded in attracting significant numbers of users [6]–[8]. According to [1], the volume of P2P TV monthly traffic exceeded 280 peta bytes in 2009. Some adaptive streaming techniques have been proposed in P2P systems such as [9] and [10]. These approaches use layered streaming as opposed to HTTP streaming. Although layered streaming has existed for a long time, its complicated design and need for high processing power specially at the clients does not make it attractive for major commercial video providers.

Despite the popularity and success of CDN-based systems, some concerns arise about their cost-effective scalability specially when supporting high quality videos to a large population of users. In order to address these issues, some recent work has proposed hybrid streaming systems that combine CDNs and Peer-to-Peer technology [8], [11]. These systems promise to achieve the scalability of P2P networks and the desired low delay and high throughput of CDNs. LiveSky [11] is an operational commercial live streaming system with more than ten million users that adopts the hybrid CDN-P2P approach. Although hybrid systems have a significant potential for providing an attractive video streaming scheme, adaptive streaming has not been extensively explored in such systems.

Designing and operating adaptive hybrid streaming systems is very challenging. By definition, these systems come with

978-1-4577-1394-1/11/$26.00 ©2011 IEEE

two degrees of freedom in their operation; one is the *adaptive* property of the system where users can switch among different streams of different qualities for the same video, and the other one is the *hybrid* operation mode which means users may receive data either from the server or from other peers streaming the same video. That said, two decisions are very critical in the design of any adaptive hybrid streaming system. The first one is the bitrate adaptation strategy which specifies how different bitrates are assigned to different users while maximizing the overall user satisfaction. The second is defining the operational guidelines a system can use to switch between the CDN and the P2P modes while efficiently utilizing both server and peer upload capacity. Another challenging issue is the interaction between these two decisions and understanding how they affect each other.

In this paper we present an analysis of adaptive streaming in a hybrid live video system with the goal of providing answers to these two design questions and studying the interactions between them. We model a system that adopts the *HTTP adaptive streaming* technology which makes it very easy to integrate into real CDN-based systems. We first present a stochastic fluid model to the hybrid streaming system with a single video bitrate and we obtain a lower bound on the number of users that should receive the stream from a CDN server in order to be able to support delivering that video stream to other users. This lower bound can be described as the switching point between the CDN and the P2P modes. We then extend this analysis to the adaptive streaming case with multiple video bitrates. We model adaptive streaming as a linear optimization problem to obtain the best bitrate adaptation strategy. In order to compare our results to CDN-based adaptive systems, we also derive results for these systems similar to what we did to the hybrid system. We developed a discrete event simulator and used simulations to validate our analysis. Our results show that adaptive streaming in hybrid systems can significantly improve the ability to satisfy more users with higher video bitrates over adaptive CDN-based systems. We also show that adaptive hybrid systems can lead to significant savings in CDN resources as opposed to CDN systems.

Related work. P2PLive [7] is one of the most famous P2P TV systems in China where users can play tens of live video channels and hundreds of on-demand movies. Multiple adaptive streaming techniques have been proposed in P2P streaming systems. For example, the work in [9] uses layered video encoding to adaptively deliver different layers of the video to clients. Another approach was used in [10] where network coding is used to make SVC more feasible in adaptive streaming. Some recent work has shown the potential benefits of using hybrid video-on-demand systems [8]. Using a nine-month trace from the MSN video service, the work in [8] shows that a hybrid P2P-CDN system could significantly reduce server bandwidth costs. The most relevant work to ours is LiveSky [11]. LiveSky is a hybrid live streaming system, however it does not support adaptive streaming. Some work has been done also in analyzing hybrid streaming systems [12], [13], however, none of them studies adaptive streaming.

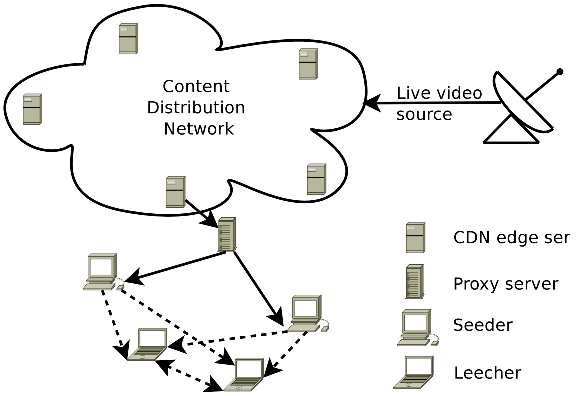


Fig. 1: System architecture

Paper outline. The rest of this paper is organized as follows. In section II we present a description of the system architecture and the main goals of our analysis throughout the paper. We present analysis for the single bitrate system in section III then in section IV we present analysis for the adaptive streaming system with multiple bitrates. We validate our analysis through simulation in section V. Evaluation results of a case study are presented in section VI. Finally, we conclude the paper in section VII.

II. SYSTEM DESCRIPTION

In this section we briefly describe how video streaming works in CDN-based systems, then we describe the components of the hybrid system and how they interact with each other.

A content distribution network consists of a set of core servers and a set of edge (surrogate) servers [14]. Core servers are responsible for managing the CDN, saving the content being distributed and when needed forwarding content to edge servers to serve requesting clients. Edge servers are the ones that actually serve client requests and they are usually distributed geographically to bring content close to as many users as possible. This helps to increase the CDN's system scalability and to improve response time for user requests.

User requests are forwarded to different edge servers in a CDN in the following manner. When a web client tries to retrieve an object for a web page it first uses DNS to resolve the server name of the URL of the object. If a CDN is used to distribute that web object, a popular way to direct a client to the best edge server is using DNS redirection. In that case, the requesting client is directed to an authoritative DNS name server that belongs to the CDN which in turn redirects the client to an edge server [15]. For a detailed description of how DNS redirection is done, the reader is referred to [16].

Broadcasting live video over CDNs is slightly different than distributing stored content. In the latter case, content is stored at edge servers hard drives and different caching techniques may be used to decide how to replace old (outdated) content with new content depending on client requests. On the other hand, in the live video case, a CDN usually has an entry point to live video where the video is encoded into a single bitrate or multiple bitrates and then the resulting streams (of different bitrates) are forwarded to edge servers when needed [17]. Note that multiple bitrates are used when adaptive streaming is enabled where clients are allowed to switch among streams of different qualities according to network and server conditions.

It is important to mention here that adaptive streaming in this paper does not mean that clients switch among different streams according to changes in their download rates. Client download rate is assumed to be variable among clients but constant over time which is true for wired connections that are not shared by multiple users. Changes in user download rates over time can be modeled as a composition of two events: departure of a user with the old bitrate, and the arrival of a new user with the new bitrate. Adaptivity in this paper comes from the fact that a CDN server has a limited capacity and it may not be able to satisfy all client requests. In that case, the server can follow one of two different strategies: 1)Serve clients with their desired bitrates in a first come first serve manner and reject any new clients when the server capacity is fully consumed. 2)Try to accommodate as many clients as possible by considering the possibility of delivering streams of lower bitrates to some clients in order to save some of the server capacity. Based on user arrival rates, video viewing duration, and the distribution of requested different video birates, we will show how the server can compute the best adaptation strategy.

A hybrid CDN/P2P streaming system usually uses the above described infrastructure of CDNs with the addition of new mechanisms for integrating peer coordination into the system. We propose the system architecture in figure 1 as our hybrid streaming system. We assume that each edge server in the CDN will have a proxy server attached to it. The proxy server can be either a software entity on the same edge server machine, or a separate machine that is connected to the edge server. When a client requests to start a live video stream, the request is directed to the proxy server through the edge server. The proxy server is responsible for the following tasks.

- Compute the best bitrate adaptation strategy.
- Keep a directory service in which it maintains a mapping between all video streams currently being transmitted and clients that are currently connected to the proxy.
- For any new streaming request, the proxy will have to make two decisions. The first one is to allocate a bitrate to the client based on the computed adaptation strategy; this bitrate could be the one originally requested by the client or could be a different, typically lower, bitrate. The second decision is whether to serve that request directly (CDN mode) or to redirect the requesting client to other clients (peers) that are streaming the same video (P2P mode). A client served in the CDN mode is called a *seeder* while a client in the P2P mode will be called a *leecher*.

Although peer selection strategies are not the focus of this paper, it is important to emphasize here that we assume that peers form random mesh networks. In the next sections we will show how the proxy server can make such decisions in a way to optimize the system performance.

III. SINGLE RATE SYSTEM MODEL

In this section we present a model for hybrid streaming using a single video bitrate, we present the adaptive streaming model with multiple bitrates in the next section. The model we develop is a stochastic fluid model similar to the one used in [18]. In our analysis, we do answer the following question to the single bitrate system: when the proxy server receives a new streaming request, should the server treat the incoming client as a *seeder* or as a *leecher*. In other words, we aim to find out how many *seeders* will be sufficient to provide a live stream to a certain number of *leechers*.

We analyze the system for the theoretical *unconstrained* case when peers can have an unlimited number of connections with other peers and for the more realistic *constrained* case when peers can only have a limited number of incoming and outgoing connections. Moreover, for these two cases we develop the analysis for systems with *churn* when clients come and go and also for *churnless* systems when the number of clients is fixed. Throughout the rest of this section we assume that the proxy server is providing a video stream with bit rate r bps, and the upload capacity of the proxy is C_{proxy} (bps).

For churn analysis, we assume that users join the system at random points in time, stay in the system for a random period, then leave the system. Previous studies of client behavior in real live streaming systems show that client arrival follows a Poisson process over short time scales [19]. Considering that our analysis can be applied to the system over short time scales, it is reasonable to assume that user arrival follows a Poisson process with rate λ. Users stay in the system for a period of time that follows a general probability distribution with mean $1/\gamma$. Define $N(t)$ as the number of users in the system at time t, then it is clear that $N(t)$ can be represented as the number of customers in a $M/G/\infty$ queue [20].

A. Unconstrained churnless system

Denote n_l as the number of leechers and n_s as the number of seeders in the system. Also define $u_i^{(l)}, u_j^{(s)}$ as the upload rates of leecher i and seeder j respectively. Note that the following two conditions must hold: $C_{proxy} \geq n_s r$, $\sum_{j=1}^{n_s} u_j^{(s)} \geq r$. The first inequality represents the server capacity constraint and the second one is necessary to guarantee that the set of seeders have the minimum upload capacity to support the video bitrate. In this system, the maximum achievable streaming rate for each client r_{max} is given by

$$r_{max} = \min\{U_s, \frac{U_s + \sum_{i=1}^{n_l} u_i^{(l)}}{n_l}\} \tag{1}$$

where U_s is the total upload rate of all seeders and $U_s = \sum_{j=1}^{n_s} u_j^{(s)}$. The proof of this result can be found in [18].

For analysis purposes, we assume that all leechers have the same upload rate of u_l and all seeders have the same upload rate of u_s. Alternatively, u_l and u_s can be considered as the average upload rates over all leechers and seeders respectively. We also assume that $r > u_l$ which means that the average upload rate of a single leecher is not enough to support sharing the whole stream with other peers. In that case r_{max} can be reduced to $r_{max} = \min\{n_s u_s, \frac{n_s u_s + n_l u_l}{n_l}\}$. When the system is in steady state condition, is it reasonable to assume that $n_s u_s > \frac{n_s u_s + n_l u_l}{n_l}$, then we conclude that a churnless system can support a streaming rate of

$$r \leq \frac{n_s u_s + n_l u_l}{n_l} \tag{2}$$

which gives the lower bound $n_s \geq \frac{n_l(r-u_l)}{u_s}$ on the number of seeders to support bitrate r.

B. Unconstrained system with churn

As we explained earlier, $N(t)$ is the total number of users in the system at time t (including seeders and leechers) and N follows a Poisson distribution with rate $\rho = \lambda/\gamma$.

We now compute the probability that the system will be able to support a streaming rate r in case of node churn. In order to do that and to simplify the analysis we assume that node churn happens only in leecher nodes. This means that the number of seeders in the system are assumed to be constant and only leechers arrive to and leave the system. This can be done by using the following simple admission policy: 1) the system starts admitting all new arrivals as seeders until n_s clients have arrived. 2) All new arrivals after that point are admitted as leechers. 3) When a seeder leaves the system, one of the leechers is randomly selected by the proxy server and is promoted to become a seeder. Following this policy will always keep the number of seeders to n_s, and then the random variable $N(t)$ will represent the number of leechers in the system at time t. Now, we can compute the probability of supporting a bitrate r as following

$$\begin{aligned} P(\text{support bitrate } r) &= P(r \leq \frac{n_s u_s + N u_l}{N}) \\ &= P(N \leq \frac{n_s u_s}{r-u_l}) = F(\frac{n_s u_s}{r-u_l}) \end{aligned}$$

where $F(w) = \sum_{x=0}^{w} \frac{e^{-\rho}\rho^x}{x!}$. We know that for large ρ we can approximate the Poisson distribution with a Gaussian distribution with mean $\mu = \rho$ and standard deviation $\sigma = \sqrt{\rho}$. Hence, we can compute the probability of supporting a stream of rate r as

$$\begin{aligned} P(\text{support bitrate } r) &= P(\frac{N-\rho}{\sqrt{\rho}} \leq \frac{\frac{n_s u_s}{r-u_l} - \rho}{\sqrt{\rho}}) \\ &= \Phi(\frac{\frac{n_s u_s}{r-u_l} - \rho}{\sqrt{\rho}}) \end{aligned}$$

where $\Phi(z)$ is the cumulative distribution function of the standard Normal random variable. Now let $\phi_{1-\alpha}$ be a positive real number such that $\Phi(\phi_{1-\alpha}) = 1-\alpha$, then a sufficient condition to guarantee supporting bitrate r with confidence $(1-\alpha) \times 100\%$ is $\big(\frac{n_s u_s}{r-u_l} - \rho\big)/\sqrt{\rho} \geq \phi_{1-\alpha}$ which gives the following lower bound on the number of seeders

$$n_s \geq \frac{(\phi_{1-\alpha}\sqrt{\rho} + \rho)(r-u_l)}{u_s} \tag{3}$$

C. Constrained churnless system

Now we consider a realistic P2P client configuration where each client has a limited number of inbound and outbound connections. We define these limits as following.

- S_{in} is the maximum number of incoming connections a seeder can accept. Each one of these connections should have a leecher on the other end of the connection. Note that data will only be flowing from the seeder to leechers in these connections.
- Y_{in} is the maximum number of incoming connections a leecher can accept. Again, each one of these connections should have another leecher on the other end of the connection. Data should flow in both directions between leechers.
- Y_{out} is the number connections a leecher can initiate, where connections can be initiated to either seeders or other leechers. We assume that there is no limit on the number of connections a leecher can initiate which means the bottleneck is in the number of connections that actually get established. This number is mainly controlled by the two parameters S_{in}, Y_{in}.

As in [21], we define η as the *efficiency* of the P2P protocol which can be computed as the probability of any leecher finding new content at other leechers when they establish a connection. It was shown in [22] that BitTorrent efficiency can exceed 0.9 if the file has more than ten pieces. It is clear that η is a function of many parameters of the P2P protocol specially the algorithm used for data exchange among peers (e.g. rarest piece first in BitTorrent protocol). The P2P protocol efficiency means that a leecher has an effective upload rate of ηu_l. Denote d as the average download rate for any leecher in the swarm, then d can be computed as

$$\begin{aligned} d &= \sum_x E[d|\text{leecher is connected to } x \text{ seeders}] \times Pr\{x\} \\ &= \sum_x \left(\frac{x u_s}{S_{in}} + \frac{(Y_{out}-x)\eta u_l}{Y_{in}}\right) \times Pr\{x\} \\ &= \frac{Y_{out}\eta u_l}{Y_{in}} + \left(\frac{u_s}{S_{in}} - \frac{\eta u_l}{Y_{in}}\right)\sum_x x Pr\{x\} \end{aligned} \tag{4}$$

Note that $\sum_x x Pr\{x\}$ is the average number of seeders connected to each leecher which can be approximated by the value $n_s S_{in}/n_l$. Note also that when $n_l \geq n_s$, we can calculate an approximate value for the number of connections each leecher can establish by $Y_{out} = (n_s S_{in} + n_l Y_{in})/n_l$. Substituting these two expressions in equation 4, then d can be reduced to

$$d = \frac{n_s u_s + \eta n_l u_l}{n_l} \tag{5}$$

Since d can be considered the average bitrate that can be supported by the system, then the number of seeders sufficient to support that bitrate is $n_s = n_l(r - \eta u_l)/u_s$. The expression in equation 5 is interesting in two aspects. First, the average leecher download rate is not directly related to the constraints of the system, namely the maximum number of uploading connections for both seeders and leechers. Second, comparing the above expression to the maximum bitrate that can be achieved in the unconstrained churnless system in equation 2 we can see that the only difference is η, the P2P protocol efficiency. This is intuitive because the difference between the unconstrained and the constrained systems is in the ability to use the upload capacity of leechers efficiently with a limited number of connections which is represented by the efficiency of the P2P protocol.

D. Constrained system with churn

Since we only have an estimate of the average bitrate that can be supported by a constrained churnless system, not an upper bound as we had with the unconstrained system, we will develop our analysis to obtain a confidence interval for the number of seeders that should be sufficient for supporting a bitrate r. Since we know the average bitrate that can be supported in a churnless constrained system from equation 5, we can now compute the probability of supporting a range of bitrates around r in a system with churn as follows

$$\begin{aligned} P(r) &= P\Big(\frac{n_s u_s + \eta N u_l}{N} - \epsilon \leq r \leq \frac{n_s u_s + \eta N u_l}{N} + \epsilon\Big) \\ &= P\Big(\frac{n_s u_s}{r - \eta u_l + \epsilon} \leq N \leq \frac{n_s u_s}{r - \eta u_l - \epsilon}\Big) \qquad (6) \end{aligned}$$

As we did earlier, N can be approximated by a Gaussian random variable with mean ρ and variance ρ. In order to compute a $(1-\alpha) \times 100\%$ confidence interval for N we set the following conditions

$$\frac{\frac{n_s u_s}{r - \eta u_l + \epsilon} - \rho}{\sqrt{\rho}} \leq -\phi_{1-\alpha/2} \quad , \quad \frac{\frac{n_s u_s}{r - \eta u_l - \epsilon} - \rho}{\sqrt{\rho}} \geq \phi_{1-\alpha/2}$$

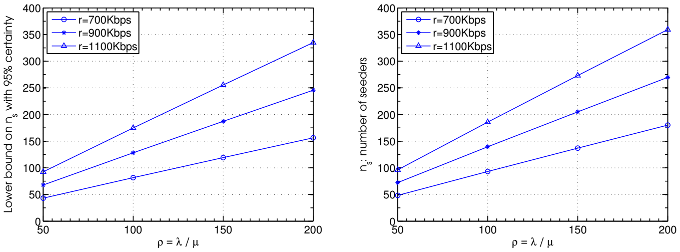
We define $\hat{\phi} = \phi_{1-\alpha/2}$, then we get the following confidence interval for the number of seeders

$$\frac{(\rho + \hat{\phi}\sqrt{\rho})(r - \eta u_l - \epsilon)}{u_s} \leq n_s \leq \frac{(\rho - \hat{\phi}\sqrt{\rho})(r - \eta u_l + \epsilon)}{u_s}$$

It is important to mention here that this inequality generates valid intervals only for ϵ values that satisfy the condition $\epsilon \geq \frac{\phi_{1-\alpha/2}(r - \eta u_l)}{\sqrt{\rho}}$. We can see that ϵ is inversely proportional to ρ which means that the higher client arrival rates and the longer clients stay in the system, the lower ϵ becomes. Lower values of ϵ mean a smaller interval in inequality 6 which yields a higher guarantee that the number of seeders given by the above interval will be sufficient for supporting the bitrate r. In figure 2 we plot the lower bound of the number of seeders against ρ for both constrained and unconstrained systems with node churn. These plots assume a 95% confidence interval on the number of seeders sufficient to support bitrate r. We can observe that the difference in the number of seeder sufficient to support different bitrates is not large.

IV. ADAPTIVE HYBRID LIVE VIDEO STREAMING

In the previous section we assumed that the CDN proxy server has a single bitrate for the live stream, in this section we consider the case when the proxy has multiple bitrates of the same live video and clients try to get the best stream according to the quality of their Internet connection. Recall that adaptive streaming in this paper comes from the fact that the CDN proxy server has a limited capacity and it may not be able to satisfy all client requests, note here that by clients we mean only *seeders* because they are the only clients that actually receive data from the proxy. In that case, the proxy will try to accommodate as many clients as possible by considering the possibility of delivering streams of lower bitrates to some clients in order to save proxy capacity.



(a) Unconstrained system (b) Constrained system

Fig. 2: n_s vs ρ for different video bitrates for systems with churn, $\alpha = 0.05$

In this section we answer the following questions about the operation of the adaptive streaming strategy:"which clients should be downgraded to streams of lower bitrates?", "what should these new lower bitrates be?", "how to get an optimal allocation of bitrates to clients while minimizing client downgrading?", and "does the adaptive solution always exist?" In order to answer these questions we formulate the adaptive streaming strategy as a linear optimization problem. We assume that when a client connects to the proxy it has a good estimate of its available bandwidth and then it requests the stream with the best bitrate accordingly. The proxy will then try to give each client the bitrate it requested or if necessary will give the client a lower bitrate, we call the difference between these two bitrates *client dissatisfaction*. The objective of our formulation is to minimize total *client dissatisfaction* over all clients.

A. Unconstrained case

Denote $r_1, \ldots, r_R$ as the different bitrates provided by the CDN proxy and assume that $r_1 > r_2 > \ldots > r_R$. Also denote n_{l_i} as the number of leechers that request a video stream of bitrate r_i and n_{s_i} as the number of seeders that receive a stream of bitrate r_i where $i = 1, \ldots, R$. Define x_{ij} as the fraction of clients that request bitrate r_i but receive bitrate r_j where $j = i, \ldots, R$ and $\sum_{j=i}^{R} x_{ij} = 1$. When $x_{ii} = 1$, this means that bitrate r_i will be delivered to all clients that requested that rate and none of them will be downgraded to a lower bitrate.

Churnless system. We know from equation 2 for the unconstrained churnless system that the relation between the number of seeders and number of leechers for each bitrate r_i can be written as $n_{s_i} u_s \geq n_{l_i}(r_i - u_l)$. Now the adaptive streaming problem can be formulated as the following linear optimization problem

$$\min \sum_{i=1}^{R} \sum_{j=i}^{R} x_{ij} n_{l_i} (r_i - r_j) \tag{7}$$

subject to: $\sum_{j=i}^{R} x_{ij} = 1$, $0 \leq x_{ij} \leq 1$ for $i = 1, \ldots, R$

$$n_{s_i} u_s \geq \left(n_{l_i} x_{ii} + \sum_{k=1}^{i-1} n_{l_k} x_{ki} - n_{s_i} \right) (r_i - u_l) \tag{8}$$

$$\sum_{i=1}^{R} n_{s_i} r_i \leq C_{proxy} \tag{9}$$

As we mentioned previously, we would like to minimize the total client dissatisfaction which is represented by the *min* objective function above. It is interesting to observe that minimizing client dissatisfaction is equivalent to maximizing *inter-client fairness* defined in [23]. Inter-client fairness is a measure of "utility" acquired by clients in the system, where greater utility means more user satisfaction. Fairness for a single client is defined as the ratio of the delivered bitrate to the actual requested bitrate. Inter-client fairness is defined as the weighted average of client fairness over all clients. To see how we make this observation, we rewrite the objective function as following

$$\sum_{i=1}^{R} \sum_{j=i}^{R} x_{ij} n_{l_i} r_i (1 - \frac{r_j}{r_i}) = \sum_{i=1}^{R} n_{l_i} r_i - \sum_{i=1}^{R} r_i \left(\sum_{j=i}^{R} n_{l_i} x_{ij} \frac{r_j}{r_i} \right)$$

The term $\sum_{i=1}^{R} n_{l_i} r_i$ has no variables and hence could be removed from the objective function. In the latter term, the weighted sum of ratios r_j/r_i can be normalized to get *inter-client fairness* as in [23]. Finally, minimizing this sum with a negative sign is equivalent to maximizing inter-client fairness.

Inequality 8 represents the condition for the number of seeders necessary to support bitrate r_i. Recall that after excluding the seeders themselves there are two sets of leechers that are going to get that bitrate. The first set contains the leechers who actually requested bitrate r_i and were not downgraded to a lower bitrate and these are represented by the term $n_{l_i} x_{ii}$. The second set consists of leechers that requested a higher bitrate and were downgraded to bitrate r_i and this set is represented by the term $\sum_{k=1}^{i-1} n_{l_k} x_{ki}$. Inequality 9 represents the proxy server capacity constraint. Note here that seeders are the only clients that receive data from the proxy and this is why leechers are not included in this condition. This optimization problem is guaranteed to have a solution only if the system can support the lowest bitrate r_R for all clients, which can be interpreted as the following condition

$$\frac{C_{proxy}}{r_R} \geq \frac{r_R - u_l}{u_s} \sum_{i=1}^{R} n_{l_i}$$

Solving this problem will result in values for x_{ij} and n_{s_i} for all i, j. Clearly, n_{s_i} will be the number of seeders that should receive video of bitrate r_i from the proxy. If $n_{s_i} = 0$ for any i it means that bitrate r_i will not be supported by the server. Moreover, $n_{s_i} = 0$ means either no clients requested bitrate r_i or some clients requested r_i but the server decided not to deliver it and downgraded these clients to lower bitrates due to overload and lack of server capacity. Alternatively, $n_{s_i} > 0$ does not necessarily mean that some clients requested bitrate r_i, it could mean that no clients requested rate r_i but the server chose to downgrade some of the clients who requested a higher bitrate to bitrate r_i. The values we get for x_{ij} can be used to randomly choose a fraction of leechers who requested bitrate r_i and deliver bitrate r_j to them.

System with churn. Assume that any arriving client will request a video stream of bitrate r_i with probability θ_i, and define $\lambda_i = \theta_i \lambda$ where λ is the general client arrival rate. We also assume that any client will stay in the system for a random period of time with average $1/\mu$. Then the number of clients of bitrate r_i at any time in the system becomes a Poisson random variable with an average $\rho_i = \lambda_i/\mu$. In this case we observe that n_{l_i} in equations 7, 8 is a Poisson random variable with mean ρ_i. Using equation 3 we can solve the same optimzation problem after replacing inequality 8 with the following one

$$n_{s_i} u_s \geq (\phi_{1-\alpha} \sqrt{\hat{\rho}_i} + \hat{\rho}_i)(r_i - u_l)$$

where $\hat{\rho}_i = \rho_i x_{ii} + \sum_{k=1}^{i-1} \rho_k x_{ki}$

Solving such a nonlinear optimization problem can be complicated and since the number of seeders n_{s_i} we get from solving this problem is approximate we choose to use a linear approximation of the above inequality. We set $\sqrt{\hat{\rho}} = a + b\hat{\rho}$ and we use curve fitting tools to find values of constants a, b. Our simulation results show that this is a very good approximation and we do not lose the accuracy of our model. In this case the value of x_{ij} is considered as the probability that when a new client requests bitrate r_i it is granted bitrate r_j.

B. Constrained case

We know from equation 5 that the relation between the number of seeders and number of leechers for bitrate r_i in a constrained churnless system is $n_{s_i} u_s \geq n_{l_i}(r_i - \eta u_l)$. Hence, in order to find the optimal solution for the adaptation strategy for the constrained churnless system we can solve the optimization problem in equation 7 after replacing inequality 8 with the following one

$$n_{s_i} u_s \geq \left(n_{l_i} x_{ii} + \sum_{k=1}^{i-1} n_{l_k} x_{ki} - n_{s_i} \right) (r_i - \eta u_l) \tag{10}$$

Similar to what we did in the previous section, in order to find the optimal adaptive strategy for the constrained system with churn, we can solve the same optimization problem after replacing inequality 8 with the following one

$$n_{s_i} u_s \geq (\phi_{1-\alpha/2} \sqrt{\hat{\rho}_i} + \hat{\rho}_i)(r_i - \eta u_l - \epsilon)$$

where $\hat{\rho}$ is defined as in the previous section.

C. CDN adaptive live streaming

One of our aims in this paper is to answer the following question "is a hybrid adaptive system better than a classic CDN adaptive system?", and if so, how much better will that be?

By a classic CDN system, we mean the system where clients are served by the closest edge server of the CDN. Moreover, by a CDN adaptive system we mean that edge servers have multiple streams of different bitrates for the same video and clients can request different bitrates according to the quality of their Internet connection. Adaptive streaming here is defined in the same way as it was defined in section IV, where edge servers can decide to downgrade some of the clients to lower bitrates in case the server capacity is not sufficient to give each client its desired bitrate.

We will first develop an analysis for the system with a single bitrate by following similar steps to what we did in the hybrid system. Consider a CDN system with a fixed number of users n. Assume the CDN edge server has capacity C_e and provides a video stream of bitrate r. We can clearly see that the maximum bitrate that can be delivered by the server of all users is C_e/n. Now consider a system with churn, we follow the same assumptions of section III of Poisson arrivals of rate λ and general distribution of duration in the system with an average $1/\gamma$. Then, the number of clients in the system at any time $N(t)$ is a Poisson random variable with mean $\rho = \lambda/\gamma$. Now we can compute the probability that the system will support clients with bitrate r as follows

$$\begin{aligned} P(\text{support } r) &= P(r \leq \frac{C_e}{N}) = P(N \leq \frac{C_e}{r}) \\ &= P(\frac{N-\rho}{\sqrt{\rho}} \leq \frac{C_e/r-\rho}{\sqrt{\rho}}) = \Phi(\frac{C_e/r-\rho}{\sqrt{\rho}}) \end{aligned}$$

And then we can obtain the following condition on the edge server capacity similar to what we did before $C_e \geq r(\rho + \phi_{1-\alpha}\sqrt{\rho})$ which guarantees with confidence $(1-\alpha) \times 100\%$ that edge server capacity will be sufficient for providing bitrate r to arriving clients with rate ρ.

There are many performance metrics that could be used to compare the performance of CDN and hybrid adaptive streaming systems. For example, we could use the total number of clients that could be accommodated with a certain level of service, or we could use the quality of service received by clients in these two systems. We select client *dissatisfaction* as our performance metric because we believe it is a reasonable quantified measure of the quality of the service received by clients. Similar to what we did in section IV-A, we model the adaptive streaming problem of a CDN system as a linear optimization problem with the objective of minimizing total client dissatisfaction.

Assume CDN edge servers have bitrates $r_1, \ldots, r_R$ where $r_1 > r_2 > \ldots > r_R$. Denote n_i as the number of clients that request bitrate r_i at the edge server. Define x_{ij} as the fraction of clients that request bitrate r_i but receive bitrate r_j. The CDN adaptive streaming problem can now be formulated as following

$$\min \sum_{i=1}^{R} \sum_{j=i}^{R} x_{ij} n_i (r_i - r_j) \tag{11}$$

subject to: $\sum_{j=i}^{R} x_{ij} = 1$, $0 \leq x_{ij} \leq 1$ for $i = 1, \ldots, R$

$$\sum_{i=1}^{R} r_i \left(n_i x_{ii} + \sum_{k=1}^{i-1} n_k x_{ki} \right) \leq C_e \tag{12}$$

Equation 11 represents our objective function of minimizing total client dissatisfaction. Edge server capacity constraint is represented by inequality 12. Recall that in a CDN system all clients receive data only from the edge server as compared to inequality 9 in the hybrid system where only seeders receive data from the proxy server. This condition means that total data rate delivered to all clients should not exceed the edge server link capacity. In order to understand inequality 12 remember that there are two sets of clients that receive bitrate r_i; the first one is represented by the term $n_i x_{ii}$ and these are the clients that requested bitrate r_i and were not downgraded to a lower bitrate. The second set is represented by the term $\sum_{k=1}^{i-1} n_k x_{ki}$ and these are the cliets that requested a bitrate higher than r_i but were downgraded by the server and eventually received bitrate r_i. It is important to mention that this optimization problem is guaranteed to have a solution only if the condition $C_e \geq r_R \sum_{i=1}^{R} n_i$ holds, which means that the CDN edge server can support the stream with the minimum bitrate r_R to all of its clients. Solving this problem, we can get the real positive values x_{ij} which can be used by the system to implement an admission control policy where the edge server should downgrade each client requesting bitrate r_i to a lower bitrate r_j with probabilty x_{ij}.

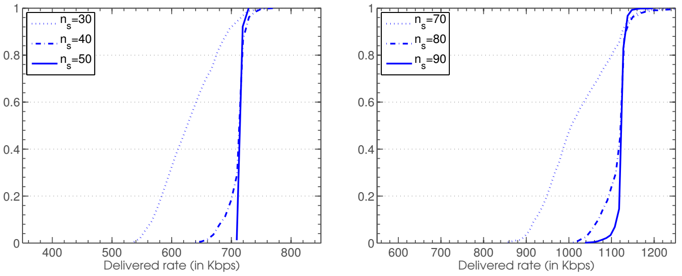
For the system with churn we can solve the same optimization problem after replacing inequality 12 with the inequality $\sum_{i=1}^{R} r_i(\hat{\rho}_i + \phi_{1-\alpha}\sqrt{\hat{\rho}_i}) \leq C_e$. We also use the linear approximation $\sqrt{\hat{\rho}} = a + b\hat{\rho}$ we used before.

V. ANALYSIS VALIDATION

A. Hybrid CDN/P2P streaming

In this section we validate our analytical results through simulation. We validate single bitrate analysis only because adaptive streaming results are based on the same analysis. We wrote a discrete event simulator for the system with a BitTorrent like client. We choose BitTorrent because it is one of most popular P2P clients and because we believe it is possible to integrate our system in real BitTorrent clients. We simulate the basic BitTorrent protocol and ignore some of the complicated details (such as tit-for-tat, peer choking, etc.).

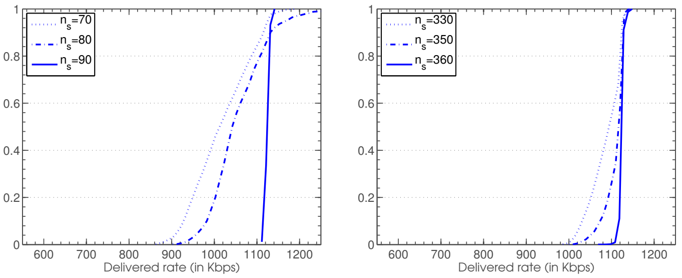
In our simulator we assume that the proxy creates a torrent file for each video file (chunk). When a client connects to the proxy, if the proxy decides to treat the client as a seeder then the client downloads both the torrent and video files for each video chunk. Once a seeder has these two files for a chunk, it starts seeding the torrent file in the BitTorrent like client. On the other hand, if the proxy decides that the current seeders are enough, the requesting client is treated as a leecher and it downloads only torrent files for video chunks as soon as they are created. Once a leecher downloads a torrent file, it starts downloading the corresponding video file from the seeding



(a) $r = 700Kbps, \rho = 50$

(b) $r = 1100Kbps, \rho = 50$

Fig. 3: CDF of average delivered rate for unconstrained system with churn



(a) $\rho = 50$

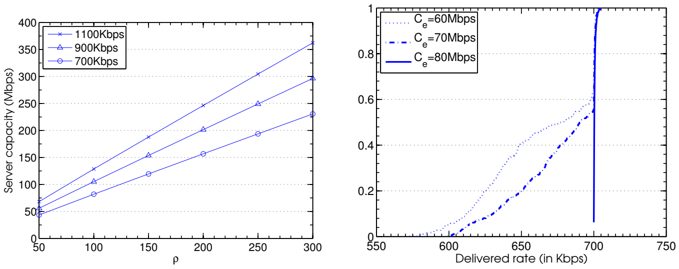
(b) $\rho = 200$

Fig. 4: CDF of average delivered rate for constrained system with churn, $r = 1100Kbps$

peers. We assume that the BitTorrent tracker functionality is implemented by the proxy server.

Throughout the rest of this section we assume the following simulation setting. Video streams are split at the proxy server into chunks of ten seconds long. The upload rate of any client is selected randomly as either 350Kbps or 500Kbps. Client arrival is assumed to follow a Poisson process with arrival rates ranging from 100 to 400 clients/hour, and recall that this is a good approximation of client arrivals in real live streaming systems [19]. According to multiple studies of live streaming systems [19], [24], client viewing duration was found to follow a heavy-tailed distribution. We follow the model in [24] and assume that client viewing duration is represented by a mixed-exponential distribution. The mixed-exponential probability density function (PDF) is $f(x) = \sum_{i=1}^{n} a_i \lambda_i e^{-\lambda_i x}$. This can easily be described as a set of n exponential distributions, with λ_i as the rate of exponential distribution i and a_i as the probability of selecting the i^{th} distribution. We also use the values of parameters a_i, λ_i that were obtained in [24]. For the constrained case we set $S_{in} = 20, Y_{in} = 10$.

We execute multiple simulation runs with multiple video bitrates ranging from $300Kbps$ to $2.4Mbps$. For each simulation run, we fix the video bitrate, number of seeders, and the number of leechers (churnless) or client arrival rate (churn). Each simulation run is worth ten hours or video streaming and at the end of each run we compute the average delivered bitrate for each leecher then we compute the cumulative distribution function (CDF) of the delivered bitrate for all leechers. In figure 3(a) we plot the CDF of delivered data rate using different number of seeders when the streamed video bitrate



(a) Server capacity (C_e) vs ρ

(b) $\rho = 100, r = 700Kbps$

Fig. 5: CDN system with churn

is 700Kbps and the arrival rate is 50 clients/hour. In figure 3(b) we repeat the same thing when the streamed bitrate is 1100Kbps. Similarly, in figures 4(a) and 4(b) we plot the CDF of delivered data rate when the streamed bitrate is 1100Kbps and client arrival rates are 50, 200 clients/hour respectively. In these plots, we can clearly see that solid lines represent the number of seeders that are sufficient to support respective bitrates to leechers, while other lines which represent lower number of seeders are not sufficient to support respective bitrates. This result matches the analysis we developed earlier in figure 2. For example, in figure 4(a) we can see that $n_s = 90$ is the minimum number of seeders that are sufficient to support the bitrate 1100Kbps, and this matches the lower bound of n_s we can get from figure 2(b) for $r = 1100$Kbps and $\rho = 50$.

B. *CDN streaming*

We wrote a client/server simulator to validate our analytical results of the single bitrate CDN-based system in section IV-C. Similar to the hybrid case, we use a Poisson process to simulate client arrivals and a mixed-exponential distribution to simulate video viewing duration. Our objective in this section is to validate the following condition we developed in section IV-C for single rate systems, $C_e \geq r(\rho + \phi_{1-\alpha}\sqrt{\rho})$. We use this condition to plot the lower bound on server capacity required for supplying bitrate r for different bitrates in figure 5(a). We execute multiple simulation runs with different video bitrates and different client behaviors. In figure 5(b) we plot the CDF of data rates delivered to clients when $\rho = 100$ and the server is streaming a bitrate of 700Kbps for different values of the server capacity, C_e. We can see that $C_e = 80$Mbps is the minimum server capacity sufficient for delivering a video bitrate of 700Kbps which matches the lower bound on C_e in figure 5(a).

VI. ILLUSTRATIVE CASE STUDY

In the previous section we developed analysis for general CDN and hybrid live streaming systems. It should be emphasized that our results can be applied to a wide variety of system configurations and parameters. In this section we consider a case study of two systems with typical configurations and we use our analysis to evaluate the performance of adaptive live video streaming in these systems. Our goal in this case study is to measure the improvement in performance (if any) from using hybrid CDN/P2P systems.

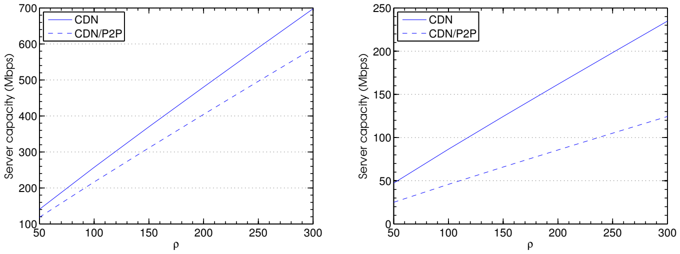
We use two evaluation metrics for this purpose. The first one is the *inter-client fairness* described in section IV-A. Recall that *inter-client fairness* has a value of 100% when all clients receive the bitrates they originally requested and as clients start to get lower bitrates (due to adaptation) this value becomes lower. The second evaluation metric is *quantifying the savings in CDN server capacity* when we use the hybrid scheme as compared to the CDN scheme.

We assume a CDN system that offers a live video stream encoded in eight different qualities with the minimum bitrate as 350Kbps and the maximum bitrate as 2.4Mbps. We also consider three different profiles of streaming requests, namely *low*, *uniform*, and *high*. Each profile represents a different distribution of client requests over the different available bitrates. In the *low* profile, client requests are mainly focused on the four lowest bitrates from 350Kbps to 1.1Mbps. Similarly, in the *high* profile, clients request streams of the four highest bitrates from 1.3Mbsp to 2.4Mbps. On the other hand, all bitrates have the same probability of being requested in the *uniform* profile. In the hybrid system, the average upload rate of a leecher, u_l, is assumed to be 300Kbps and the average upload rate of a seeder, u_s, is assumed to be 500Kbps.

We performed evaluation for systems with and without churn but we show results only to the more important case of systems with churn. We assume the CDN has a fixed server/proxy capacity of 500Mbps then we change client arrival rates and also change request distributions to be one of the profiles mentioned above.

In figure 6 we plot inter-client fairness against ρ, the average number of customers in the system, for both hybrid and CDN systems. In addition, we plot the same metric for a single-rate hybrid streaming system. Although in this case clients request different bitrates, we can apply single-rate hybrid streaming in the following manner: if the system is able to support the lowest bitrate that was requested by some clients then this bitrate is provided for all clients. Otherwise, the system will try all lower bitrates until it finds a bitrate that can be supported to all clients. Note that a client is always capable of receiving a bitrate lower than the bitrate it requested, but the reverse is not true.

One expected observation for both hybrid adaptive and CDN systems and for all profiles is that inter-client fairness starts as 100% for lower number of customers in the system then it drops when there are more customers in the system. This is because the server(proxy) capacity is sufficient to satisfy client requests when the number of clients in the system is low, but when this number grows high, the server becomes over-loaded and unable to satisfy all clients with the bitrates they asked for. At this point, the server(proxy) applies the adaptation strategy for the CDN(hybrid) system and decides to downgrade some clients to lower bitrates. The only exception to this observation is the *low* profile in figure 6(c) where inter-client fairness stays at 100% even when the number of clients in the system increases. This is because in the *low* profile case, the server(proxy) was able to give each client the bitrate it asked for without having to downgrade any client.



(a) High bitrate bias

(b) Low bitrate bias

Fig. 7: Required server capacity for CDN/P2P and CDN systems with churn

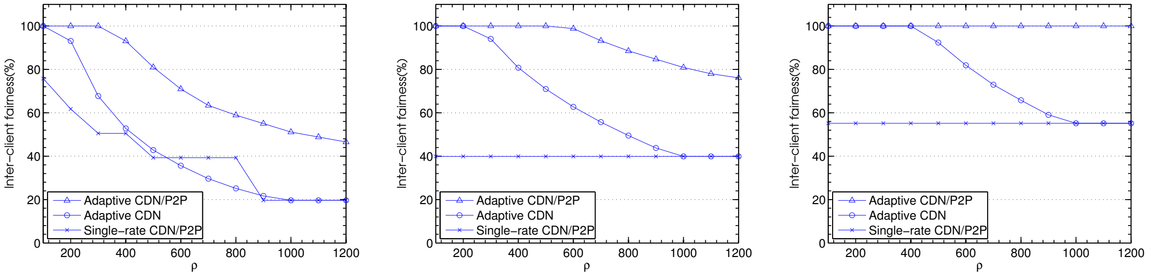
On the other hand, the single-rate hybrid system starts with inter-client fairness less than 100% for all profiles. This is because the best bitrate this system can provide is the lowest bitrate requested by some clients, for example in the *high* profile case, the best bitrate that can be provided for all clients is 1.3Mbps. Additionally, since the lowest bitrate (in this system), 350Kbps, is requested by some clients in the *low* and *uniform* profiles, the system has to provide that bitrate to all clients no matter how many clients are in the system, and this is why inter-client fairness remains constant in these two profiles. An important observation is that, as the number of customers in the system increases, the CDN adaptive system performance approaches the performance of the single-rate hybrid system, and in some cases (*high* profile), single-rate systems can even do better. From figure 6 we can see that hybrid systems can improve inter-client fairness from 20% to 40% over CDN systems depending on the distribution of the number of requests to different bitrates and the average number of customers in the system.

In order to measure savings in CDN server capacity from using the hybrid system compared to CDN systems we ask the following question; if we were to achieve inter-client fairness of 100%, how much server capacity do we need for both the hybrid and CDN cases? To do that, we fix all system parameters including client arrival rate and request distribution profile then we compute the server(proxy) capacity that will satisfy all client requests with their desired bitrates for both the CDN and the hybrid cases. Using the analysis developed in sections III-D, IV-C we can calculate savings in CDN server capacity as

$$C_e - C_{proxy} = \sum_{i=1}^{R} (\phi_{1-\alpha}\sqrt{\rho_i} + \rho_i)(\eta u_l + \epsilon)$$

It is interesting to see that capacity saving from using hybrid systems is directly proportional to both the average number of clients in the system ρ_i and the average client upload rate u_l. This means that as the number of clients in the system increases, hybrid systems become more effective which can be observed from figures 6, 7.

In figure 7 we plot the computed server(proxy) capacity against the average number of customers in the system. We can see that hybrid systems can save about 21%, 32%, and 100% of CDN server capacity in the high, uniform, and low profiles



(a) High bitrate bias (b) Uniform distribution over bitrates (c) Low bitrate bias

Fig. 6: Inter-client fairness for systems with churn

respectively. The reason capacity savings is less in the *high* profile than other profiles is that peer contribution in hybrid systems become more limited when clients request higher bitrates. This is because we assume clients have asymmetric upload and download rates which means client upload rate is much less than client download rate. For example, a standard cable connection in the United States has a download rate of 3Mbps while the upload rate is usually in the range $380/760$Kbps. On the other hand, hybrid systems can be much more effective in other countries where users have more symmetric download/upload rates (e.g. China).

VII. CONCLUSION

In this paper, we analyze adaptive streaming in a hybrid CDN/P2P live streaming system. Our analysis is driven by the need to develop solutions to two important design questions in hybrid systems. The first question is how to find a way to switch the operation of the system between the CDN and P2P modes. The second question is how to find the best bitrate adaptation strategy. This strategy basically specifies how bitrates are assigned to different clients when the CDN server can not satisfy each client with the bitrate it requested due to capacity constraints. We believe that these two design decisions mostly control the effectiveness of any hybrid adaptive streaming system.

We develop a stochastic fluid model for a hybrid streaming system with a single video bitrate. We obtain theoretical results that help the CDN server decide when to switch from the CDN to P2P mode. After that, we extend that model to the multiple bitrate case and we develop a linear optimization formulation to get the best bitrate adaptation strategy. Using simulations, we validate our analysis. We use our analysis to evaluate a case study of typical CDN and hybrid systems. Results show that hybrid systems can improve average user satisfaction about 20% to 40% as compared to CDN systems depending on the distribution of client requests to different bitrates. We also find that hybrid systems can achieve significant savings in CDN server capacities as compared to CDN systems; these savings could be from 21% to 100% again depending on the distribution of client requests to different bitrates.

REFERENCES

[1] Cisco, “Cisco visual networking index: Forecast and methodology, 2009–2014,” 2010.

[2] “Inauguration day, by the numbers.” [Online]. Available: ”http://news.cnet.com/8301-13577_3-10145923-36.html”

[3] “http://www.youtube.com/live.”

[4] S. Nelakuditi, R. Harinath, E. Kusmierek, and Z.-L. Zhang, “Providing smoother quality layered video stream,” in *NOSSDAV*, 2000.

[5] T. Kim and M. Ammar, “Optimal quality adaptation for mpeg-4 fine-grained scalable video,” in *Infocom*, 2003.

[6] X. Zhang, J. Liu, B. Li, , and T.-S. P. Yum, “Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming,” in *Infocom*, 2005.

[7] X. Heia, C. Liang, J. Liang, Y. Liu, and K. Ross, “Insights into pplive: A measurement study of a largescale p2p iptv system,” in *IPTV Workshop, International World Wide Web Conference*, 2006.

[8] C. Huang, J. Li, , and K. Ross, “Can internet video-on-demand be profitable?” in *Sigcomm*, 2007.

[9] R. Rejaie and A. Ortega, “Pals: Peer-to-peer adaptive layered streaming,” in *NOSSDAV*, 2003.

[10] A. T. Nguyen, B. Li, and F. Eliassen, “Chameleon: Adaptive peer-to-peer streaming with network coding,” in *Infocom*, 2010.

[11] Hao Yin and Xuening Liu and Tongyu Zhan and Vyas Sekar and Feng Qiu and Chuan Lin and Hui Zhang and and Bo Li, “Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experience with LiveSky,” in *Multimedia*, 2009.

[12] D. Xu, S. Kulkarni, C. Rosenberg, and H.-K. Chai, “Analysis of a cdn–p2p hybrid architecture for cost-effective streaming media distribution,” *Multimedia Systems*, vol. 11, pp. 383–399, 2006.

[13] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, “Performance bounds for peer-assisted live streaming,” in *SIGMETRICS*, 2008.

[14] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, “Globally distributed content delivery,” *IEEE Internet Computing*, vol. 6, pp. 50–58, 2002.

[15] B. Krishnamurthy, C. Wills, and Y. Zhang, “On the use and performance of content distribution networks,” in *ACM SIGCOMM INTERNET MEASUREMENT WORKSHOP*, 2001.

[16] A.-J. Su, D. Choffnes, A. Kuzmanovic, and F. Bustamante, “Drafting behind akamai,” in *SIGCOMM*, 2006.

[17] Akamai, “http://www.akamai.com/dl/featured-sheets/akamai-media-streaming.pdf.”

[18] R. Kumar, Y. Liu, and K. Ross, “Stochastic fluid theory for p2p streaming systems,” in *Infocom*, 2007.

[19] K. Sripanidkulchai, B. Maggs, and H. Zhang, “An analysis of live streaming workloads on the internet,” in *Internet Measurement Conference (IMC)*, 2004.

[20] L. Kleinrock, *Queueing Systems, Vol. 1:Theory*. John Wiley ans Sons, 1975.

[21] D. Qiu and R. Srikant, “Modeling and performance analysis of bittorrent-like peer-to-peer networks,” in *SIGCOMM*, 2004.

[22] S. Tewari and L. Kleinrock, “Analytical model for bittorrent-based live video streaming,” in *IEEE Consumer Communications and Networking Conference*, 2007.

[23] T. Jiang, M. Ammar, and E. Zegura, “Inter-receiver fairness: A novel performance measure for multicast abr sessions,” in *SIGMETRICS*, 1998.

[24] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao, “Modeling user activities in a large iptv system,” in *Internet Measurement Conference (IMC)*, 2009.