UNDERSTANDING THE FLASH CROWD IN P2P LIVE VIDEO STREAMING SYSTEMS

Fangming Liu[†], *Bo Li*[*†], *Lili Zhong*[†], *Baochun Li*[‡]

[†]*Hong Kong University of Science & Technology*, [‡]*University of Toronto*

ABSTRACT

Peer-to-Peer (P2P) live video streaming systems have recently received significant attention, with commercial deployment gaining increased popularity in the Internet. It is evident from our experiences with real-world systems that, it is not uncommon to have hundreds of thousands of users trying to join a program in the first few minutes of a live broadcast. This phenomenon, unique in live streaming systems, referred to as the flash crowd, poses significant challenges in the system design. In this paper, we develop a mathematical model to capture the inherent relationship between time and scale in P2P streaming systems under the flash crowd. Specifically, we show that there is an upper bound on the system scale with respect to a time constraint. In addition, our analysis has brought forth an in-depth understanding on the effect from the Gossip protocol and churn effects.

***Index Terms*—** Video streaming, peer-to-peer, flash crowds

1. INTRODUCTION

Recently, the Internet has witnessed a significant increase in the popularity of peer-to-peer (P2P) live media streaming applications, that deliver real-time and sustained media content to potentially millions of users. As participating peers not only download media streams, but also contribute their upload bandwidth capacities to serve one another, such systems are potentially more scalable, and are thus cost-effective to be deployed, compared to traditional infrastructure-based solutions, such as IP multicast or Content Delivery Networks.

While recent measurement studies [1, 2] on real-world P2P streaming systems have demonstrated that the streaming performance can be typically maintained at a high level once the systems have reached a reasonable scale, this is challenged by a severe phenomenon called the *flash crowd*, in which there could be a large number of peers arriving at the system within a short period of time, just after a new live event has been released. It is evident in our empirical experiences from the latest version of Coolstreaming+ [3] that, it is considerably more challenging for a P2P streaming system to accommodate an abrupt surge of newly arrived peers, with reasonable streaming qualities and initial startup delays. As a result, we also observe a considerable portion of peers undergoing a flash crowd could opt to leave the system due to impatience, which leads to more dynamic system scaling behavior and more constrained system scale limits.

In this paper, we seek to analyze and understand the inherent relationship between time and scale in P2P streaming systems under a flash crowd scenario (henceforth referred to as *scale-time*), through a tractable analytical model that we propose. Specifically, our major contributions are: (1) We first derive the fundamental constraint of the scale-time relationship with the upper bound of system scale over time, which explains in depth *why* the intuitive *"demand vs. supply"* condition is insufficient to capture the system scale. (2) We further proceed to an enhanced constraint that quantitatively characterizes *how* the system scale is further constrained by the timing constraint, if the partial knowledge of peers and their competition for the limited upload bandwidth resources in the system are taken into account. (3) Motivated by our empirical experiences, we further extend our model and analysis to more general and realistic peer arrival patterns, and investigate the impact from peer departures due to impatience on the system scale. In addition, our analytical framework also offers us the flexibility to investigate the effects of various critical factors, including the initial system scale, the scale of the flash crowd, the peer upload capacity, and the number of partners each peer has.

The remainder of this paper is organized as follows. Sec. 2 discusses related work. Sec. 3 presents our theoretical model for P2P streaming systems under a flash crowd, along with the fundamental scale-time relationship and the roles of various critical factors. In Sec. 4, we carry out a series of numerical analysis to demonstrate the scale-time relationship in P2P streaming systems under a flash crowd, as well as the effects of various critical factors. Finally, Sec. 5 concludes the paper with remarks on future work.

2. RELATED WORK

With respect to analytical studies on P2P streaming systems, Kumar *et al.* [4] have derived the maximum streaming rate for churnless systems and developed a stochastic fluid model

*The research was support in part by grants from RGC under the contracts 615608, and 616207, by a grant from NSFC/RGC under the contract N_HKUST603/07, by a grant from HKUST under the contract RPC06/07.EG27.

with peer churn to examine its performance. There have also emerged a number of analyses on the performance bounds of tree-based or mesh-based systems in terms of streaming rate, delay, and server load (*e.g.*, [5–7]), particularly through the perspective of chunk dissemination to participating peers. Along this direction, a more recent study [8] has analyzed the performance gap between the fundamental limits and actual performance of mesh-pull systems. Zhou *et al.* [9] have compared, through a stochastic model, different chunk scheduling strategies based on the performance metrics of continuity and startup latency. Our study is different from and complementary to these prior works in that we analyze the asymptotic scaling behavior of P2P streaming systems during a flash crowd, while paying particular attentions on essential aspects such as the partial knowledge of peers and their competition for the limited resources, as well as the peer departures due to impatience.

Flash crowd issues were also examined in other P2P applications. Yang and Veciana [10] used a branching process model to examine the service capacity of BitTorrent-like file sharing systems during flash crowds. Another modeling work in [11] provided a theoretical evaluation of the scalability of a distributed randomized P2P search protocol that provides transmission of objects from servers currently suffering with flash crowds. Leibnitz *et al.* [12, 13] compared the file download performances of Client/Server and P2P systems while considering flash crowd arrivals and users' impatience. Different from these works, we study the flash crowd challenges in P2P streaming systems with more stringent requirements on the bandwidth resources to meet the streaming rate. To our knowledge, this paper, for the first time, attempts to provide an analytical characterization and understanding of the scale-time relationship in P2P streaming systems, with a particular focus on the flash crowd and various critical factors.

3. SYSTEM MODEL AND FUNDAMENTAL PRINCIPLES

3.1. System Model

In this section, we present our basic model for P2P live video streaming under a flash crowd, including the underlying assumptions and notations summarized in Table 1. We consider a video with rate $R = xr$ to be streamed to all participating peers, where r is the bit rate corresponding to a unit of bandwidth, and R corresponds to the bandwidth requirement of x units. This can alternatively be related to the concept of *substreams* in the real-world large-scale P2P streaming system Coolstreaming+ [2], in which a media stream is divided into multiple substreams and peers could subscribe to different substreams from different partners.

For a peer i, let u_i denote the upload capacity of the peer. The peer download capacity is assumed not to be the bottleneck, which is in accordance with most of the recent Internet access technologies and measurement studies of existing P2P systems [14]. Given a streaming rate R, we define the *relative surplus upload capacity* h_i of a peer i as the ratio of $(u_i - R)$ to r. Let u be the average peer upload capacity and h be the relative average peer surplus capacity, which will be elaborated in Theorem 1 (Sec. 3.2) later.

Table 1. Key Parameters in the System Model.

Notation	Definition
M	Initial system scale.
N	Flash crowd scale.
R	Video streaming rate ($= xr$).
u_i	Upload capacity of peer i.
h_i	Relative surplus upload capacity of peer i ($= (u_i - R)/r$).
u	Average peer upload capacity.
h	Relative average peer surplus capacity ($= (u - R)/r$).
k	Number of partners of a new peer ($\geq x$).
$S(t)$	System scale (number of existing peers) in the t-th time slot.
U_s	Server capacity provisioning.
u_s	Relative server capacity provisioning ($= U_s/R$).

To capture essential aspects of practical systems, yet be still simple enough to yield relevant insights, our model mainly considers the following aspects:

▷ *First, initial system capacity*. We assume initially there are M existing peers that already joined the system. That is, they have obtained sufficient upload bandwidth resources to satisfy the streaming rate, and are able to contribute their upload capacities to the system. We assume that there exists one or multiple servers in the system with aggregate upload capacity U_s. Given a streaming rate R, the relative server capacity u_s is defined as the ratio of U_s/R.

▷ *Second, flash crowd*. We start by focusing on an extreme flash crowd scenario where $N(\gg M)$ peers arrive at approximately the same time [8], just after a new live event has been released. Each new peer that has yet to join the system needs to gather at least x units of upload bandwidth resource from those existing peers to meet the streaming rate requirement. Our model strives to capture the difficulty for peers to gather sufficient upload bandwidth resources at startup, which we believe is a critical issue under a flash crowd. Furthermore, we also extend our model to more general and realistic peer arrival patterns during a flash crowd, in order to make our analysis more representative of real-world systems.

▷ *Third, system scale and initial startup delays*. Without loss of generality, we assume that time t is slotted. If a *new peer* — one that has not yet joined the system — has obtained sufficient upload bandwidth resource (*i.e.*, x units) at the t-th time slot, it is regarded as "joined the system" and counted towards the system scale $S(t)$ of existing peers. Otherwise, the peer will continue to seek upload bandwidth resource along the subsequent time slots until it joins the system. In our model, once a peer is able to join the system, it will not leave the system during the flash crowd. From the perspective of user experience, the time t represents the initial startup delays for peers. In addition, motivated by our recent empirical study [3] that peers undergoing a flash crowd could opt to leave the system due to excessive startup delays, we further refine our model by taking into account peers' impatience and its impact on the system scale over time.

▷ *Fourth,* we first analytically consider the case of global knowledge and centralized control of the system, which yields an upper bound of the system scale over time. Further, we proceed to demonstrate the effects of partial knowledge, by a simple random partner selection strategy. Specifically, each new peer will randomly select k partners from the current set of existing peers to ask for their surplus upload capacities in each time slot. Since an existing peer can be selected by a number of new peers, it would randomly choose a certain number of them to supply its upload bandwidth resource, depending on its surplus capacity. Such a random partner selection strategy with parameter k essentially represents the decentralized gossiping among peers to gather upload bandwidth resource. This is a reasonable assumption, as such a strategy is typically adopted in many practical P2P systems (*e.g.*, BitTorrent and Coolstreaming) for bootstrapping peers, mainly due to its simplicity.

Different from the perspective of chunk dissemination that takes the peer streaming buffer state or/and chunk scheduling as main consideration (*e.g.*, [5, 6, 8, 9]), we attempt to provide a complementary perspective in this paper: we analyze the asymptotic scaling behavior of the system, rather than the individual peer behavior.

Based on this system model, we are able to derive a tractable theoretical framework in Sec. 3.2, which reveals the fundamental relationship between time and scale in P2P streaming systems under a flash crowd, as well as insights on the impacts from various critical factors, including k, h, M, N, peer arrival patterns, and peer departures due to impatience.

3.2. Scale-Time Relationship with Critical Factors

First of all, we derive the fundamental constraint of the scale-time relationship in P2P streaming systems, even with global knowledge and centralized control of the systems: *While "the average peer uploading capacity should be no less than the average peer downloading rates" is a necessary condition for P2P streaming systems to scale, it is insufficient to capture the system scale, as the upload bandwidth resource from newly arrived peers cannot be utilized immediately*. This leads to the following upper bound of system scale over time.

***Theorem* 1** *For a P2P streaming system with a given streaming rate R and average peer upload capacity u, the system scale after the t-th time slot, $S(t)$, has the following upper bound:*

$$S(t) \leq \min\{(1 + \frac{u-R}{R})^t (M + C) - C, N + M\}, \quad (1)$$

where $C = U_s/(u-R)$, M is the initial system scale at time $t = 0$, U_s is the server capacity provisioning, and N is a flash crowd of newly arrived peers.

Proof: Clearly, the system scale cannot exceed the total number of peers, including both existing and new peers; thus, $S(t) \leq N + M$.

Furthermore, the system scale after each time slot $S(t)$ is bounded by the aggregate upload bandwidth resource that is *currently available* in the system, which depends on the number of existing peers in previous time slots (*i.e.*, $S(t-1)$) and their surplus upload capacities h_i, as well as the server capacity provisioning U_s. If these resources can be fully utilized, which essentially implies that global knowledge and centralized control of the system can be achieved, then

$$\begin{aligned} S(t) &\leq S(t-1) + \frac{\sum\limits_{i\in S(t-1)} h_i}{x} + \frac{U_s}{R} \\ &= S(t-1) + S(t-1)\frac{h}{x} + \frac{U_s}{R} \\ &\leq (1+\frac{h}{x})^t S(0) + \frac{U_s}{u-R}\left((1+\frac{h}{x})^t - 1\right) \\ &= (1+\frac{u-R}{R})^t (M + \frac{U_s}{u-R}) - \frac{U_s}{u-R}. \end{aligned}$$

Combining the above two bounds gives Eq. (1). Equivalently, it also implies the minimum time to accommodate a flash crowd of N peers. ■

Note that this fundamental upper bound neither depends on specific flash crowd arrival patterns, nor the bandwidth unit. However, it intuitively would still be too optimistic as it assumes all current surplus bandwidth resources from existing peers can be fully utilized. *Since the system scale is further constrained by the partial knowledge of peers and their competition for limited resources, how can we quantify such effects?* To this end, we proceed to analyze the scale-time relationship with a random partner selection strategy as follows.

Since it has already been proved in [4, 8] that the average peer upload capacity u satisfies $u > R$ in large-scale streaming systems, we shall focus on the general homogeneous case where $u_i = u > R$ (*i.e.*, $h_i = h > 0$) for all peers. This is reasonable as we are more interested in the asymptotic collective behavior of the system rather than the individual peer behavior. As we focus on such a homogeneous case, we first ignore the server capacity, and will introduce it as a parameter later.

***Lemma* 1** *For a P2P streaming system with each peer having partial knowledge of the system and a random partner selection strategy (*i.e.*, each new peer independently and randomly selects k partners from the set of existing peers), the number of new partners of an existing peer during the t-th time slot, $q(t,k)$, is a random variable that follows a binomial distribution with parameters $(N+M-S(t-1), k/S(t-1))$, and an expected value of*

$$E[q(t,k)] = \frac{k(N+M-S(t-1))}{S(t-1)}, \quad (2)$$

where $S(t-1)$ is the current number of existing peers in the system.

Proof: At the beginning of the t-th time slot, the number of existing and new peers in the system is $S(t-1)$ and $N+M-S(t-1)$, respectively. Since each new peer independently and randomly selects k partners from those existing peers, the probability for an existing peer to be selected as a partner by a new peer is $C^{k-1}_{S(t-1)-1}/C^{k}_{S(t-1)} = k/S(t-1)$. Hence, the probability for an existing peer to be selected as a partner by i new peers is a binomial distribution with parameters $(N+M-S(t-1), k/S(t-1))$. Hence, the expected value of $q(t,k)$ can be expressed as Eq. (2). ■

Based on Lemma 1, we can derive an approximation of the expected system scale as follows.

***Theorem* 2** *For a P2P streaming system with each peer having partial knowledge of the system and a random partner selection strategy, assume that each existing peer could randomly provide each of its new partner with 1 unit of upload bandwidth resource with a probability of $h/q(t,k)$. If we use the expected value $E[q(t,k)]$ given by Eq. (2) as an approximation of $q(t,k)$, then the expected system scale after the t-th time slot, $E[S(t)]$, can be approximated by*

$$E[S(t)] \approx S(t-1) + (N+M-S(t-1)) \times \sum_{i=x}^{k} C^i_k p(t,k,h)^i \left(1-p(t,k,h)\right)^{k-i}, \quad (3)$$

where $p(t,k,h) \approx h\alpha(t)/k$ is the probability for a new peer to obtain 1 unit of upload bandwidth resource from an existing peer; and $\alpha(t) = S(t-1)/(N+M-S(t-1))$ is the ratio of the number of existing peers to the number of new peers in the system at the beginning of the t-th time slot.

Proof: Based on Lemma 1, we have $q(t,k) \sim \text{Binomial}(N+M-S(t-1), k/S(t-1))$. Since one of the important features of a binomial distribution is that its probability mass function $\Pr[q(t,k)=j]$ gains the highest value at $j = E[q(t,k)]$, we choose $E[q(t,k)]$ given by Eq. (2) to approximate $q(t,k)$ for all existing peers. Then, $p(t,k,h)$ can be derived as

$$p(t,k,h) \approx \frac{h}{E[q(t,k)]} = \left(\frac{h}{k}\right)\left(\frac{S(t-1)}{N+M-S(t-1)}\right) = \frac{h}{k}\alpha(t).$$

Then, the amount of upload bandwidth resource i that can be obtained by a new peer can be simplified to a binomial distribution with parameters $(k, p(t,k,h))$. The corresponding probability mass function is $C^i_k p(t,k,h)^i \left(1-p(t,k,h)\right)^{k-i}$.

Furthermore, recall that a new peer needs to gather at least x units of upload bandwidth resource (corresponding to the streaming rate R) to join the system; hence, the expected system scale after the t-th time slot, $E[S(t)]$, can be approximated by Eq. (3). ■

Theorem 2 with Eq. (3) qualitatively indicates that, $p(t,k,h)$ plays an important role for the system scale, which depends on $\alpha(t)$, h, and k. The effects of these factors will be thoroughly demonstrated in Sec. 4.

Furthermore, as demonstrated by both the real-world experience [3] and the numerical results (Sec. 4) derived from our model, P2P streaming systems by nature do not react well to a flash crowd. Specifically, the system scale grows relatively slower during the initial time slots. This motivates a natural question: *How a certain amount of server capacity provisioning can help improve the system scale?* Based on Theorem 2, we can approximately derive the improved system scale with a given amount of server capacity provisioning as follows.

***Corollary* 1** *For a P2P streaming system with a streaming rate of R and an aggregate server upload capacity U_s, assume that server(s) support a number of $u_s = U_s/R$ randomly selected new peers at the beginning of each time slot. The remaining $N+M-S(t-1)-u_s$ new peers still rely on the $S(t-1)$ existing peers through a random partner selection strategy. Then, the expected system scale $E[S(t)]$ given by Theorem 2 can be potentially improved as*

$$E[S(t)] \approx S(t-1) + u_s + (N+M-S(t-1)-u_s) \times \sum_{i=x}^{k} C^i_k p'(t,k,h,u_s)^i \left(1-p'(t,k,h,u_s)\right)^{k-i} \quad (4)$$

where $p'(t,k,h,u_s) = h\alpha'(t,u_s)/k$, $\alpha'(t,u_s) = S(t-1)/(N+M-S(t-1)-u_s)$, and $u_s = U_s/R$ is the relative server capacity.

The proof of Corollary 1 is similar to the proof of Theorem 2. The effects of the parameter u_s will be quantitatively demonstrated in Sec. 4.

In addition to the extreme flash crowd scenario, our model can easily be extended to more general and realistic peer arrival patterns as follows.

***Corollary* 2** *For a P2P streaming system with each peer having partial knowledge of the system and a random partner selection strategy as assumed in Theorem 2, then given any specific peer arrival pattern $\lambda(t)$, the expected system scale*

$E[S(t)]$ given by Theorem 2 can be extended as

$$E[S(t)] \approx S(t-1) + (\int_0^t \lambda(\tau)d\tau + M - S(t-1)) \times \sum_{i=x}^{k} C_k^i p(t,k,h)^i (1-p(t,k,h))^{k-i}, \quad (5)$$

where $p(t,k,h) \approx h\alpha(t)/k$ is the probability for a new peer to obtain 1 unit of upload bandwidth resource from an existing peer; and $\alpha(t) = S(t-1)/(\int_0^t \lambda(\tau)d\tau + M - S(t-1))$ is the ratio of the number of existing peers to the number of new peers in the system at the beginning of the t-th time slot.

The proof of Corollary 2 is also similar to the proof of Theorem 2. In Sec. 4, we will examine the scale-time relationship in P2P streaming systems under both the extreme flash crowd scenario and another two typical peer arrival patterns.

Based on Corollary 2, we are further interested in an important performance concern: the startup delays experienced by peers that arrive at different time slots during a flash crowd. To this end, we derive the *distribution of peer startup delays* as follows. First, let $f(t,k,h) = \sum_{i=x}^{k} C_k^i p(t,k,h)^i (1-p(t,k,h))^{k-i}$. For the set of peers that arrive in the t_i-th time slot, the expected portion of peers that have not joined the system at the end of the t_j-th time slot is

$$W(t_i,t_j) = \prod_{t=t_i}^{t_j} (1 - f(t,k,h)), \qquad \text{for} \quad (t_j \geq t_i).$$

Then, the expected portion of peers that arrive in the t_i-th time slot and then join the system in the t_j-th time slot, can be expressed as

$$\begin{aligned} J(t_i,t_j) &= W(t_i,t_{j-1}) - W(t_i,t_j) \\ &= \left(\prod_{t=t_i}^{t_{j-1}} (1 - f(t,k,h)) \right) \times f(t_j,k,h), \quad (6) \end{aligned}$$

where $t_j \geq t_i$. We will use Eq. (6) to examine the peer startup delays in Sec. 4.

Furthermore, observed from our recent measurement study on real-word P2P streaming systems [3], a considerable portion of users undergoing a flash crowd could opt to leave the system due to excessive startup delays. To capture this user behavior and its impact on the system scale, we further introduce a *peer impatience time threshold θ* into our model, after which a new peer aborts its joining attempt and leaves the system. In reality, individual peers could have different impatience times θ, which tends to be a random variable highly depending on peers' individual behaviors and the current states they are being in (*e.g.*, how long they have waited so far). However, in order to make our analysis tractable, we use an average threshold $E[\theta]$ as an approximation of impatience time for all the new peers. Then, the scale-time relationship given in Corollary 2 can be further extended as follows.

***Corollary* 3** *For a P2P streaming system with each peer having partial knowledge of the system and a random partner selection strategy as assumed in Theorem 2, if we use an expected threshold $E[\theta]$ as an approximation of impatience time for all the new peers, after waiting for which new peers would leave the system; then, given any specific peer arrival pattern $\lambda(t)$, the expected system scale $E[S(t)]$ given by Corollary 2 can be extended as*

$$E[S(t)] \approx S(t-1) + \left(\int_0^t \lambda(\tau)d\tau + M - S(t-1) - \int_0^{t-1} D(\tau)d\tau \right) \sum_{i=x}^{k} C_k^i p'(t,k,h,\theta)^i (1-p'(t,k,h,\theta))^{k-i}, \quad (7)$$

where $D(\tau) \approx \lambda(\tau - E[\theta])l(\tau)$ for $\tau \geq E[\theta]$ (otherwise, $D(\tau) = 0$) is the number of new peers that leave the system over time slots due to impatience, $l(\tau) = 1 - ((S(\tau) - S(\tau - E[\theta]))/(\int_0^\tau \lambda(t)dt + M - S(\tau - E[\theta]) - \int_0^{\tau - E[\theta]} D(t)dt))$ is a probability for new peers that arrived at the $(\tau - E[\theta] + 1)$-th time slot yet still have not obtained sufficient upload bandwidth resources for startup at the end of τ-th time slot. $p'(t,k,h,\theta) \approx h\alpha'(t,\theta)/k$, and $\alpha'(t,\theta) = S(t-1)/(\int_0^t \lambda(\tau)d\tau + M - S(t-1) - \int_0^{t-1} D(\tau)d\tau)$.

Proof: Suppose new peers arrive at the system over time slots according to a certain pattern $\lambda(t)$. If we use an expected threshold $E[\theta]$ as an approximation of impatience time for all the new peers, then at the end of each time slot $\tau(\geq E[\theta])$, there will be a number of $D(\tau)$ new peers that arrived at the beginning of the $(\tau - E[\theta] + 1)$-th time slot leave the system due to impatience (*i.e.*, their waiting times exceed $E[\theta]$).

Within the period of $[\tau - E[\theta], \tau]$, the number of new peers that compete for the limited upload bandwidth resources is

$$\int_0^\tau \lambda(t)dt + M - S(\tau - E[\theta]) - \int_0^{\tau - E[\theta]} D(t)dt,$$

among which $S(\tau) - S(\tau - E[\theta])$ new peers have obtained sufficient upload bandwidth resources and successfully joined the system during this period. Hence, the probability for the new peers that arrived at the beginning of the $(\tau - E[\theta] + 1)$-th time slot to join the system during this period is $(S(\tau) - S(\tau - E[\theta]))/(\int_0^\tau \lambda(t)dt + M - S(\tau - E[\theta]) - \int_0^{\tau - E[\theta]} D(t)dt)$. In other words,

$$l(\tau) = 1 - \frac{S(\tau) - S(\tau - E[\theta])}{\int_0^\tau \lambda(t)dt + M - S(\tau - E[\theta]) - \int_0^{\tau - E[\theta]} D(t)dt}.$$

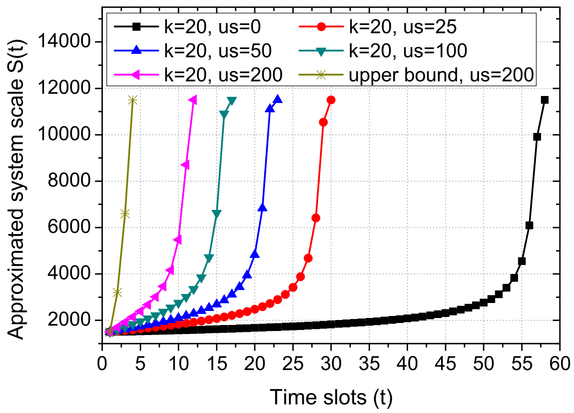

Fig. 1. Approximated system scale over time slots, with different amount of server capacity provisioning. We set the initial system scale M to 1500 and flash crowd scale N to 10000. The number of partners for new peers k is set to a typical value of 20. The relative server capacity provisioning u_s varies from 0 to 200. Others are set as $h = x = 5$.

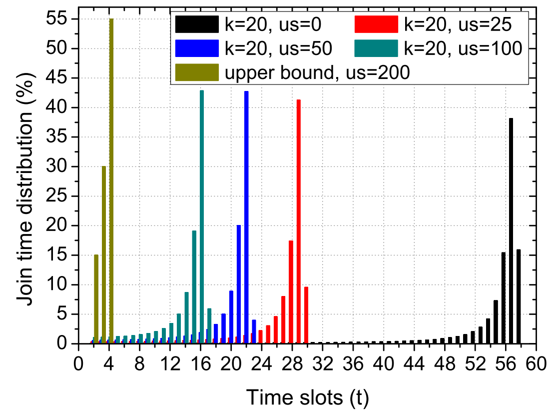

Fig. 2. Peer join time distribution versus time slots, with different amount of server capacity provisioning. We set the initial system scale M to 1500 and flash crowd scale N to 10000. The number of partners for new peers k is set to a typical value of 20. The relative server capacity provisioning u_s varies from 0 to 200. Others are set as $h = x = 5$.

Since the total number of new peers that arrived at the beginning of the $(\tau - E[\theta] + 1)$-th time slot is $\lambda(\tau - E[\theta])$, $D(\tau)$ can be approximated by

$$D(\tau) \approx \lambda(\tau - E[\theta])l(\tau), \qquad \text{for} \quad \tau \geq E[\theta].$$

For $\tau \in [0, E[\theta]]$, we have $D(\tau) = 0$, as the waiting times of all the new peers within this initial period have not exceeded the impatience threshold.

Since we still assume a random partner selection strategy as in Theorem 2, the remaining part of proof is similar to the proof of Theorem 2 except that both $\alpha'(t, \theta)$ and $p'(t, k, h, \theta)$ should take into account a set of departure peers up to the current time $\int_0^{t-1} D(\tau)d\tau$. Hence, $E[S(t)]$ given by Eq. (5) can be extended as Eq. (7). ■

Corollary 3 with Eq. (7) qualitatively indicates that the departures of impatient peers during a flash crowd could indirectly alleviate the heavy competition among new peers for the limited pool of upload bandwidth resources, thus the system could scale up more quickly in a transient period. However, the cost is the loss of a considerable portion of peers, which eventually cuts down the system scale. We shall quantitatively demonstrate these effects in Sec. 4.

4. NUMERICAL RESULTS AND INSIGHTS

In this section, we take advantage of the theoretical results derived from our model to demonstrate the fundamental scale-time relationship in P2P streaming systems under a flash crowd, as well as the effects of various critical factors.

4.1. Scale-Time Relationship and Join Time Distribution

Fig. 1 compares the approximated system scale over time slots obtained by Theorem 1, 2 and Corollary 1, under the same flash crowd scenario setting. We observe the following:

First, the system scale grows relatively slower during initial time slots, as a surge of newly arrived peers compete for the limited surplus capacities from a relatively smaller number of existing peers. This results in considerable difficulty for new peers to obtain sufficient upload bandwidth resources.

Second, as more peers gradually joining the system with positive gain of surplus capacities, the ratio of the number of existing peers to the number of new peers $\alpha(t)$ continuously increases and the entire system capacity improves; thus the system scale ramps up more and more quickly.

Third, as expected, the system scale can be improved with an additional amount of server capacity provisioned, especially for the initial time slots. However, we note that the improvement slows down with more and more server capacity provisioned, as demonstrated by the decreasing gaps between the curves.

To reflect the user experience under a flash crowd, Fig. 2 plots the peer join time distribution (*i.e.*, the percentage of peers that joined the system in each time slot). It shows that potentially many peers could suffer from long startup delays under a flash crowd; while only a small portion of peers can join the system within the initial time slots. As an additional amount of server capacity is provisioned, the join time distribution noticeably shifts towards the earlier time slots, with a relatively larger portion of peers joining the system with shorter startup delays.

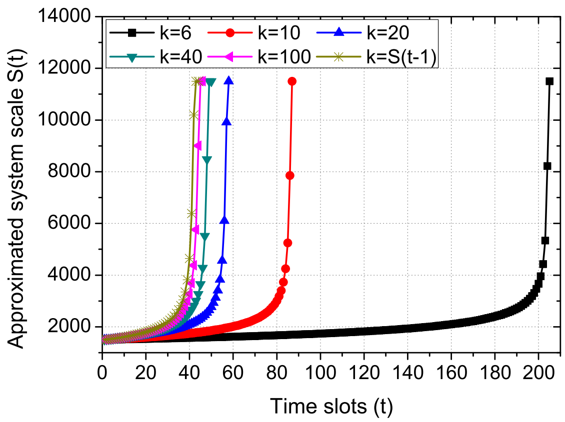

Fig. 3. Approximated system scale over time slots, with different settings of the number of partners for new peers k. We set the initial system scale M to 1500 and flash crowd scale N to 10000. The value of k varies from 6 to $S(t-1)$. Others are set as $u_s = 0, h = x = 5$.

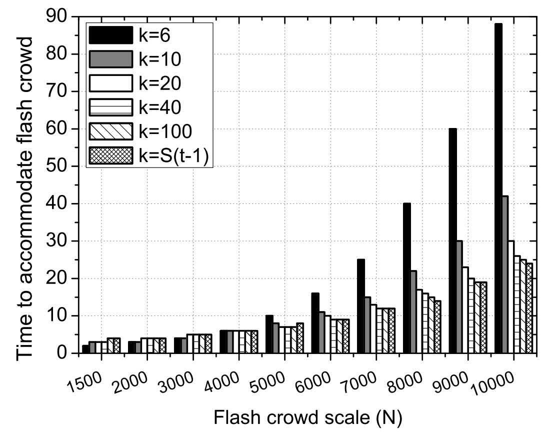

Fig. 4. Time to accommodate different scales of a flash crowd, under different settings of the number of partners for new peers k. We set the initial system scale M to 1500. The value of k varies from 6 to $S(t-1)$. Others are set as $u_s = 0, h = 6, x = 5$.

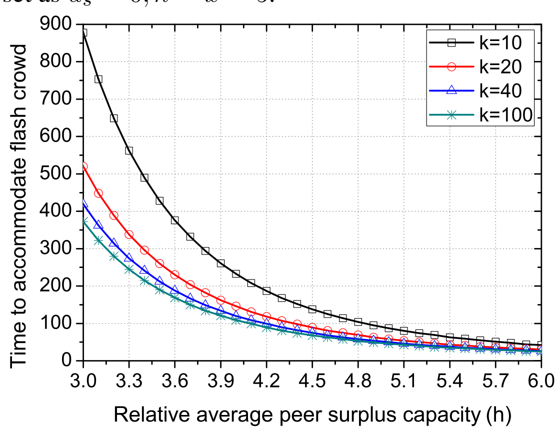

Fig. 5. Time to accommodate a flash crowd of $N = 10000$ peers when relative average peer surplus capacity h varies, under different settings of the number of partners for new peers k. We set the initial system scale M to 1500. The value of k varies from 10 to 100. Others are set as $u_s = 0, x = 5$.

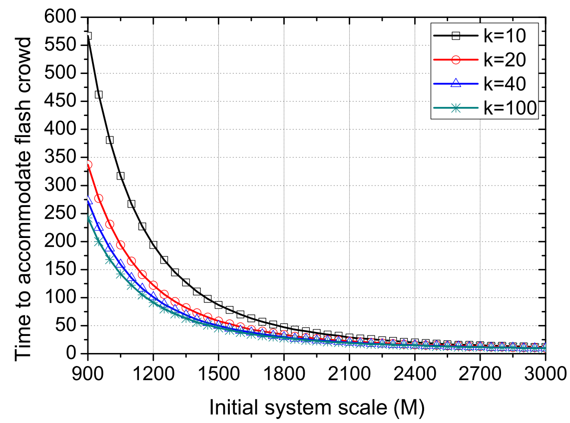

Fig. 6. Time to accommodate a flash crowd of $N = 10000$ peers when the initial system scale M varies, under different settings of the number of partners for new peers k. The value of k varies from 10 to 100. Others are set as $u_s = 0, h = x = 5$.

The above findings suggest that an adequate amount of additional server capacity provisioning could help alleviate the flash crowd effect in P2P streaming systems, and improve the user experience with shorter initial startup delays. Specifically, it can help improve the system scale during the initial period of a flash crowd. Once the system scale reaches a reasonable level (*e.g.*, this can be simply reflected by $\alpha(t)$, which can be roughly captured by the tracking server used for peer registration and discoveries), peer resources would then be sufficient for the system to scale up further, and thus the server capacity can be reduced accordingly.

4.2. Sensitivity Analysis on Critical Factors

We next demonstrate the effects of several critical factors indicated by Theorem 2, by carrying out a series of sensitivity analysis. Specifically, we apply the classical approach of varying one or two parameters while keeping others constant.

First, Fig. 3 compares the approximated system scale over time slots, by varying the number of partners for new peers k. We observe that the system scale improves significantly as k increases in the range of typical settings that real-world systems use [2]. Equivalently, the time to accommodate a given scale of a flash crowd decreases significantly. However, when k continues to increase to larger values up to the size of current set of existing peers $S(t-1)$, the improvements, though still exist, become relatively minor.

We further examine the effects of k by comparing the time to accommodate different scales of a flash crowd when k varies, as shown in Fig. 4. We observe that: (1) When the flash crowd is less severe relative to the initial system capac-

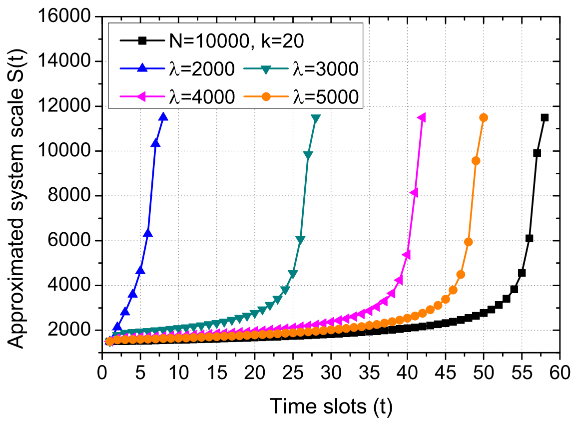

Fig. 7. Approximated system scale over time slots, under a constant peer arrival pattern $\lambda(t) = \lambda$. We set the initial system scale M to 1500 and flash crowd scale N to 10000, with the peer arrival rate λ varying from 2000 to 5000. The number of partners for new peers k is set to a typical value of 20. Others are set as $u_s = 0, h = x = 5$.

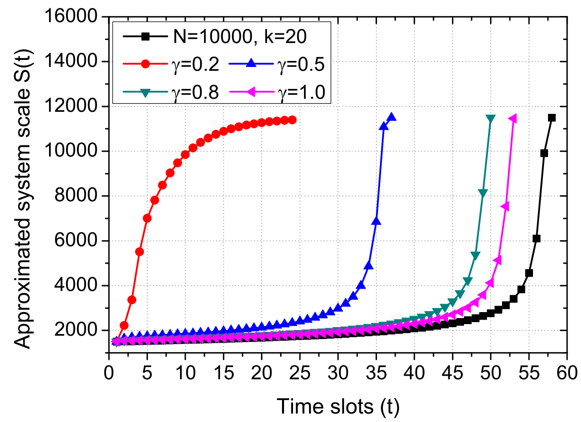

Fig. 8. Approximated system scale over time slots, under an exponentially decreasing peer arrival pattern $\lambda(t) = \beta e^{-\gamma t}$. We set the initial system scale M to 1500 and flash crowd scale $N = \lim_{t\to\infty} \int_0^t \lambda(\tau)d\tau = 10000$, with the parameter γ varying from 0.2 to 1.0. The number of partners for new peers k is set to a typical value of 20. Others are set as $u_s = 0, h = x = 5$.

ity (*i.e.*, the demand to supply ratio of $(Nx)/(Mh)$ is relatively less stringent), results are relatively insensitive to different values of k. Specifically, the increase of k actually does not help (*e.g.*, when the flash crowd scale $N = 4000$, the time to accommodate it under different values of k stays nearly the same); or could even bring negative effects when the flash crowd scale decreases. This is in conflict with the intuitive belief that an increase of the number of partners for peers can always help reduce the startup delays and improve the system scale. (2) As the scale of the flash crowd increases, our results become more sensitive to different values of k, and there are remarkable improvements by increasing k. However, excessive increase of k brings relatively minor improvements, which consists with previous observation from Fig. 3.

Finally, we examine the impact from the relative average peer surplus capacity h, the initial system scale M, and their correlation with k. Fig. 5 and Fig. 6 plot the time to accommodate a given scale of a flash crowd when h or M varies, respectively, under different settings of k. We observe that: (1) As expected, the increase of h or M can effectively reduce the time to accommodate flash crowd, as it essentially enhances the entire system capacity. In general, the more upload bandwidth resources exist in the system (though it takes time to utilize them), the less time it takes to accommodate a flash crowd. (2) The impact of k observed in Fig. 4 is also verified. When the upload bandwidth resource is relatively constrained (*i.e.*, when h or M decreases), the performance gaps (in terms of time saved) between different settings of k are more profound.

4.3. Effects of Peer Arrival Patterns

In addition to the extreme flash crowd scenario, we further utilize Corollary 2 to examine the scale-time relationship in P2P streaming systems under two typical peer arrival patterns as follows.

First, Fig. 7 plots the approximated system scale over time slots under a constant peer arrival rate $\lambda(t) = \lambda$. To make the result comparable to that of the extreme flash crowd scenario, the total number of new peers that arrive at the system (*i.e.*, the flash crowd scale) is limited to the same as in Fig. 1, *i.e.*, $\lambda\tau = N$, where N peers arrive at the system in a period of τ with a constant rate λ. We can see that our previous discussion on the scale-time relationship still holds for such a constant peer arrival pattern. More importantly, given a specific scale of flash crowd, the system scale under a constant peer arrival pattern grows more quickly with shorter initial startup delays compared to the extreme flash crowd case (marked as $N = 10000$). This indicates that a more realistic peer arrival pattern is less challenging than the pessimistic flash crowd scenario. As the peer arrival rate λ decreases, the initial startup delays decrease remarkably. Specifically, when the peer arrival rate is close to the initial system scale (*e.g.*, $\lambda = 2000$ which is close to $M = 1500$), the system is able to accommodate all the new peers shortly after they arrived.

Based on the above observation, a natural intuition is that *all the new peers arriving at the system within a shorter period (as λ increases) is the sole reason that makes the flash crowd more challenging*. To examine this intuition, we further compare the constant peer arrival pattern with an even

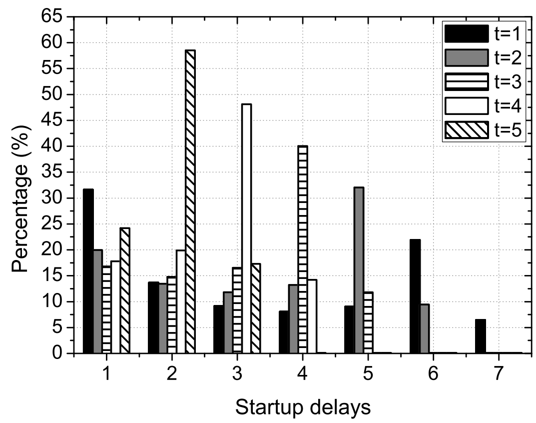

Fig. 9. The distribution of startup delays for peers that arrive at different time slots, under a constant peer arrival pattern $\lambda(t) = \lambda$. We set the initial system scale M to 1500 and flash crowd scale N to 10000, with the peer arrival rate $\lambda = 2000$. The number of partners for new peers k is set to a typical value of 20. Others are set as $u_s = 0, h = x = 5$.

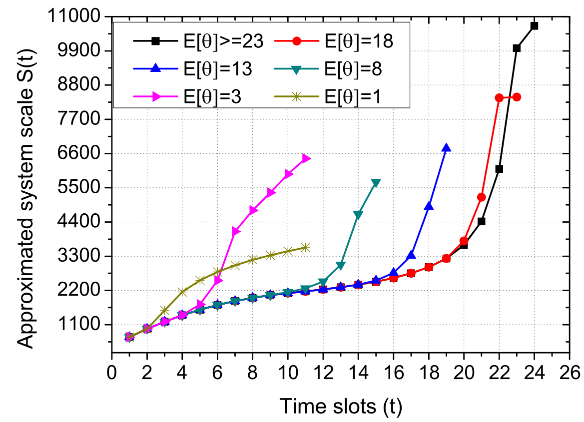

Fig. 10. Approximated system scale over time slots, under a constant peer arrival pattern $\lambda(t) = \lambda$ and different values of the expected peer impatience time threshold $E[\theta]$. We set the initial system scale M to 700 and flash crowd scale N to 10000, with $\lambda = 1000$ and $E[\theta]$ varying from 23 to 1 time slots. The number of partners for new peers k is set to a typical value of 20. Others are set as $u_s = 0, h = x = 5$.

more realistic pattern — exponentially decreasing peer arrival pattern $\lambda(t) = \beta e^{-\gamma t}$ as assumed in [13]. To make a fair comparison, we match them under the setting that almost all the new peers arrive at the system within the same period. Specifically, the flash crowd scale is limited to $\lim_{t\to\infty} \int_0^t \lambda(\tau)d\tau = \beta/\gamma = N$; moreover, we also set $\int_0^t \beta e^{-\gamma\tau}d\tau = \lambda t = \delta N$, where δ is close to 1.0. Under such a setting, we use the correlation between λ and γ to plot the approximated system scale over time slots under $\gamma = 0.5, 0.8, 1.0$ in Fig. 8, which correspond to the curves with $\lambda = 2000, 3000, 4000$ in Fig. 7. Beyond the above intuition, it shows that although almost all the new peers arrive in a same period for both of the patterns, the resulting performances of them differ significantly. This implies that with a same number of new peers arriving at the system in a same period, different peer arrival patterns could result in remarkably different system scaling performance.

Additionally, we also observe that the shape of the curve with $\gamma = 0.2$ is different from other curves. The rationale is that when the flash crowd scenario is less severe relative to the initial system capacity (*e.g.*, $\beta = N\gamma = 2000$ which is close to $M = 1500$), the growing rate of the system scale over time could catch up with the peer arrival rate, after which the flash crowd period actually ends and the system scale is constrained by the relatively lower peer arrival rate, rather than the system capacity.

Next, we examine the startup delays experienced by peers that arrive at different time slots during a flash crowd. Using Eq. (6), Fig. 9 plots the distributions of startup delays for peers that arrive at different time slots, under a constant peer arrival pattern with $\lambda = 2000$. It clearly demonstrates that different arrival times could result in remarkably different distributions of startup delays. Generally, later arrivals tend to experience shorter startup delays compared to early arrivals. For example, while almost all the peers that arrive at $t = 5$ quickly join the system within 3 time slots, those that arrive at $t = 1$ are likely to wait much longer. This is consistent with our previous analysis in Sec. 4.1 that, during the initial period of the flash crowd, it is difficult for early arrived peers to obtain sufficient upload bandwidth resources. As more peers gradually join the system, the entire system capacity improves, from which later arrivals as well as those waiting peers benefit.

4.4. Effects of Peer Impatience and Departures

It is evident in our recent measurement study [3] that, a considerable portion of peers undergoing a flash crowd could opt to leave the system due to impatience. Here we use Corollary 3 to examine the impact from this user behavior on the system scale.

Fig. 10 compares the approximated system scale over time slots with different values of the expected peer impatience time threshold $E[\theta]$, under a constant high peer arrival rate relative to the initial system scale. We observe the following interesting results. On one hand, as the expected peer impatience time threshold $E[\theta]$ decreases, the system could scale up more quickly during the flash crowd with relatively more peers, especially later arrivals, experiencing shorter startup delays. Noticeably, the approximated system scale with $E[\theta] = 3$ quickly reaches a high level at the 10-th time slot, which is nearly 3 times of the other curves with larger values of $E[\theta]$. This implies that the departures of impatient peers during a flash crowd could alleviate the heavy competition for the limited pool of upload bandwidth resources, as reflected by $\alpha'(t, \theta)$ and $p'(t, k, h, \theta)$ in Corollary 3.

On the other hand, it clearly demonstrates that with more stringent peer impatience time thresholds, the system would scale up to even more constrained limits, as a considerable portion of peers have left the system due to impatience. For example, after all the new peers during the flash crowd either left or joined the system, the final system scale with smaller values of $E[\theta]$ is only around half (*e.g.*, $E[\theta] = 3, 8, 13$) or even a quarter (*e.g.*, $E[\theta] = 1$) of that with loose impatience thresholds (*e.g.*, $E[\theta] \geq 23$).

5. CONCLUSION AND FUTURE WORK

In this paper, we have studied the inherent relationship between time and scale in P2P streaming systems during a flash crowd, through a mathematical framework we developed. We have derived an upper bound on the system scale and demonstrated that the timing factor plays a critical role for such a system to scale. Furthermore, our analysis also brings a more in-depth understanding with respect to the partial knowledge of peers and their competition for the limited pool of upload bandwidth resources, as well as important insights on a few other critical factors. In addition, our analysis also demonstrates that the system scaling behavior and peers' startup delays are strongly affected by different peer arrival patterns and peer departures due to impatience.

There are several avenues for further studies. For example, since P2P streaming systems under a flash crowd show transient behaviors [3], it is desirable to construct a transient stochastic model which characterizes variabilities of relevant factors and provides more accurate analysis and implications for system design. In addition, it is also desirable to consider other bursty patterns of peer arrival and departure, which is more representative of real-world systems. Furthermore, from the perspective of additional server capacity provisioning, it is also important to dynamically adjust additional capacities from servers to adapt to the size of the flash crowd. We defer these investigations to our future work.

6. REFERENCES

[1] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Trans. Multimedia*, Dec. 2007.

[2] B. Li, S. Xie, Y. Qu, Y. Keung, C. Lin, J. Liu, and X. Zhang, "Inside the New Coolstreaming: Principles, Measurements and Performance Implications," in *Proc. of IEEE INFOCOM*, Apr. 2008.

[3] B. Li, Y. Keung, S. Xie, F. Liu, Y. Sun, and H. Yin, "An Empirical Study of Flash Crowd Dynamics in a P2P-based Live Video Streaming System," in *Proc. of IEEE Globecom*, Nov. 2008.

[4] R. Kumar, Y. Liu, and K. W. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," in *Proc. of IEEE INFOCOM*, Apr. 2007.

[5] Y. Liu, "On the Minimum Delay Peer-to-Peer Video Streaming: How Realtime Can It Be?" in *Proc. of ACM Multimedia*, Sep. 2007.

[6] T. Bonald, L. Massoulie, F. Mathieu, D. Perino, and A. Twigg, "Epidemic Live Streaming: Optimal Performance Trade-Offs," in *Proc. of ACM SIGMETRICS*, Jun. 2008.

[7] S. Liu, R. Z. Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance Bounds for Peer-Assisted Live Streaming," in *Proc. of ACM SIGMETRICS*, Jun. 2008.

[8] C. Feng, B. Li, and B. Li, "Understanding the Performance Gap between Pull-based Mesh Streaming Protocols and Fundamental Limits," in *Proc. of IEEE INFOCOM*, Apr. 2009.

[9] Y. Zhou, D. Chiu, and J. Lui, "A Simple Model for Analyzing P2P Streaming Protocols," in *Proc. of IEEE International Conference on Network Protocols (ICNP)*, Oct. 2007.

[10] X. Yang and G. de Veciana, "Service Capacity of Peer to Peer Networks," in *Proc. of IEEE INFOCOM*, Mar. 2004.

[11] D. Rubenstein and S. Sahu, "Can unstructured p2p protocols survive flash crowds?" *IEEE Transactions on Networking*, vol. 13, no. 3, pp. 501–512, Jun. 2005.

[12] K. Leibnitz, T. Hoßfeld, N. Wakamiya, and M. Murata, "Peer-to-Peer vs. Client/Server: Reliability and Efficiency of a Content Distribution Service," in *20th International Teletraffic Congress*, Jun. 2007.

[13] T. Hoßfeld and K. Leibnitz, "Modeling and Evaluation of an Online TV Recording Service," in *9th Annual Workshop on Mathematical Performance Modeling and Analysis*, Jun. 2007.

[14] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, Analysis, and Modeling of BitTorrent-like Systems," in *Proc. of ACM Internet Measurement Conference (IMC)*, Oct. 2005.