How Do P2P Streaming Systems Scale Over Time Under a Flash Crowd?

Fangming Liu[†], Bo Li[*†], Lili Zhong[†], Baochun Li[‡], Di Niu[‡]
[†]*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 in our empirical experiences with real-world systems that, it is not uncommon to have hundreds of thousands of viewers trying to join a program in the first few minutes of a live broadcast. This phenomenon in live streaming systems, referred as the flash crowd, poses unique challenges in the system design. In this paper, we develop a mathematical model to capture the inherent relationship between time and scale during a flash crowd. We derive an upper bound on the system scale, and then demonstrate that the timing factor plays a critical role for such a system to scale. In addition, our analysis also brings a more in-depth understanding with respect to the use of Gossip protocols, *i.e.*, the effects of partial knowledge.**

I. 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.

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. 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.

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 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. While recognizing the significance of these prior works, our study is different from and complementary to them. 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.

II. System Model and Fundamental Principles

A. 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 I. 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

[*]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.

access technologies and measurement studies of existing P2P systems [10]. 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. II-B) later.

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 focus 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.

▷ *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.

▷ *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 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.

TABLE I
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 ($= U_s/R$).

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. II-B, 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, and N.

B. 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:

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

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

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

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

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

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

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

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 $q(t,k) \sim$ 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

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

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. III.

Furthermore, as demonstrated by both the real-world experience [3] and the numerical results (Sec. III) 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

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

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. III.

III. 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.

A. 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 total 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.

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.

B. 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, 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 capacity (*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

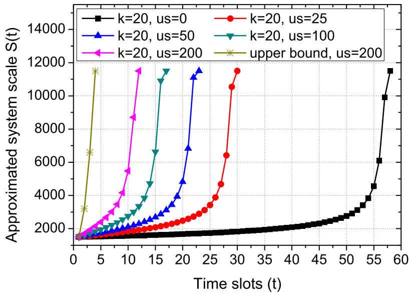

Fig. 1. Approximated system scale along time slots, with different amount of server capacity provisioning. We set the initial system scale M to 1500 and flash crowd scale N to 10000. The number of partners for new peers k is set to a typical value of 20. The relative server capacity provisioning u_s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

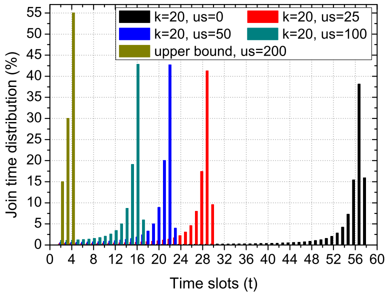

Fig. 2. Peer join time distribution versus time slots, with different amount of server capacity provisioning. We set the initial system scale M to 1500 and flash crowd scale N to 10000. The number of partners for new peers k is set to a typical value of 20. The relative server capacity provisioning u_s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

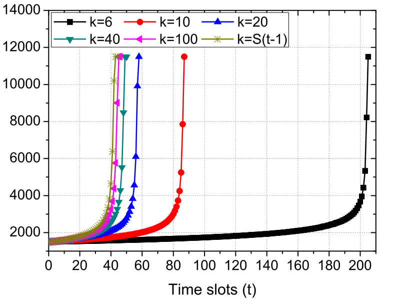

Fig. 3. Approximated system scale over time slots, with different settings of the number of partners for new peers k. We set the initial system scale M to 1500 and flash crowd scale N to 10000. The value of k varies from 6 to $S(t-1)$. Others are set as $u_s = 0, h = x = 5$.

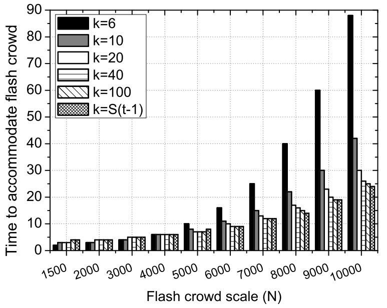

Fig. 4. Time to accommodate different scales of a flash crowd, under different settings of the number of partners for new peers k. We set the initial system scale M to 1500. The value of k varies from 6 to $S(t-1)$. Others are set as $u_s = 0, h = 6, x = 5$.

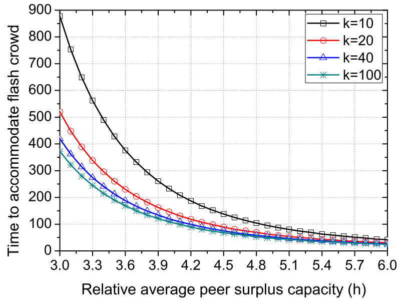

Fig. 5. Time to accommodate a flash crowd of $N = 10000$ peers when relative average peer surplus capacity h varies, under different settings of the number of partners for new peers k. We set the initial system scale M to 1500. The value of k varies from 10 to 100. Others are set as $u_s = 0, x = 5$.

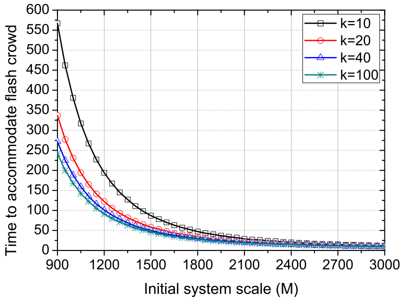

Fig. 6. Time to accommodate a flash crowd of $N = 10000$ peers when the initial system scale M varies, under different settings of the number of partners for new peers k. The value of k varies from 10 to 100. Others are set as $u_s = 0, h = x = 5$.

gaps (in terms of time saved) between different settings of k are more profound.

IV. 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. In addition, 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.

We believe that this work represents only the first step towards analyzing flash crowd behavior of P2P streaming systems. For example, it is desirable to consider more general and bursty patterns of peer arrival and departure, which is more representative of real-world systems. 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.

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 2008*, 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 2008*, 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] 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 2006)*, Oct. 2005.