Do BitTorrent-like VoD Systems Scale under Flash-Crowds?

Lucia D'Acunto, Tamas Vinko and Johan Pouwelse
Delft University of Technology, The Netherlands
l.dacunto@tudelft.nl

Abstract—The efficiency of BitTorrent for file sharing has inspired a number of BitTorrent-based P2P protocols for Video-on-Demand (VoD). It has been shown that these systems are scalable in steady-state: the service quality provided to the users does not depend on the number of users in the system. However, it is not well understood how these systems scale under flash-crowds. In this work, we model a general BitTorrent-like VoD system and we find that under a flash-crowd the quality-of-service (QoS) degrades with the number of users. Also, our analysis shows that, at the very beginning of a flash-crowd, the maximum number of simultaneous users that can obtain a given service level is intrinsically related to two fundamental system parameters, namely the initial service capacity and the efficiency of piece exchange of the underlying P2P protocol. Finally, we illustrate the impact of peers turning into seeders (i.e peers that have finished downloading and remain in the system to upload) on the system scale.

I. INTRODUCTION

Recently, a lot of research has been conducted on how to effectively use the P2P architecture to provide large scale video streaming services. The high efficiency of BitTorrent, which allows for the utilization of a large fraction of peer upload bandwidth [3], has inspired a number of P2P algorithms for video distribution, both live and on demand [1], [9], [12].

While many studies have been conducted on the scalability of BitTorrent-like live video streaming systems, both in the steady-state and during flash-crowds [4], [8], the analysis of the BitTorrent approach for VoD distribution has mostly focused on the steady-state [6], [10].

In this paper we aim at filling this gap, and provide an analysis of the scalability of the BitTorrent-based approach during flash-crowds. In fact, since flash-crowds are not infrequent in VoD systems [12], service providers need to understand the performance of P2P technology before deploying their system, in order to be able to provide users with an acceptable level of service. The contributions of this paper are the following:

1) we propose a simple model to study the effects of a flash-crowd on the service quality experienced by the users (Section III);
2) we show that the scale of a BitTorrent-like VoD system during flash-crowd is constrained by the efficiency of piece exchange of the underlying P2P protocol as well as the initial service capacity (Section IV);
3) we illustrate the impact of peers turning into seeders on the system scale (Section IV).

Hence, our analysis will allow service providers to set the appropriate parameters of a BitTorrent-like VoD system, such as the initial service capacity necessary to support a certain number of users at a given playback rate.

II. RELATED WORK

BitTorrent is a popular P2P protocol for file sharing. One important characteristic that makes BitTorrent very scalable is the *rarest-first* piece selection policy. This policy increases piece diversity in the system and therefore allows an efficient utilization of peer upload bandwidth.

Because of its high efficiency, a lot of effort has focused on adapting BitTorrent to VoD [1], [2], [11]. Also, extensive work has been done on modelling and analyzing BitTorrent-like VoD systems. Parvez et al. [6] study the performance of such systems and conclude that they are scalable during the steady-state. Lu et al. [10] propose a fluid model and study the evolution of downloaders and seeders in time. However, they do not analyze the average startup delay experienced by peers (i.e. how long a peer has to wait before playback starts), which is an important metric to evaluate the performance of a P2PVoD system.

III. FLASH CROWD MODEL FOR A P2PVoD SYSTEM

This section presents a simple fluid model that characterizes the behavior of a BitTorrent-like VoD system during a flash-crowd. The model is inspired by the classical fluid model for BitTorrent [7] and the P2PVoD models in [6], [10].

A. Model description

We consider a system where a group of peers joins to watch a specific video file with playback rate R and duration L. We assume that there is an original source (consisting of one or more servers or initial seeders) which contributes an aggregate upload capacity C_s. Peers have an average upload capacity μ and join the system at rate $\lambda(t)$ and do not leave before they have become seeders. Seeders, on the other hand, depart from the system at a rate γ (hence, we can consider $\frac{1}{\gamma}$ to be the average seeding time).

B. Flash-crowd scenario

When a flash-crowd occurs, the system enters a critical phase where the available bandwidth is scarce. After some time, the appearance of seeders will alleviate the negative performance impact, even if peers continue joining at a high rate. Therefore, in our analysis we distinguish two phases in a flash-crowd, depending on whether seeders have started appearing or not.

- *Starting phase ($0 < t < t_s$)*

  Excluding the initial source, there is no seeder in the system until the first downloaders complete the download of the entire video file (which happens at time t_s), thus becoming the first seeders. Hence, at the very beginning of a flash-crowd, the evolution of downloaders $x(t)$ and seeders $y(t)$ in the system is characterized by the following equations

$$\begin{cases} \dfrac{dx(t)}{dt} = \lambda(t), \\ \dfrac{dy(t)}{dt} = 0, \\ x(0) = \lambda(0), y(0) = 0, \end{cases} \quad (1)$$

  and the average download speed can be computed as

$$u(t) = \frac{C_s + \mu \eta x(t)}{x(t)} = \frac{C_s}{x(t)} + \mu \eta. \quad (2)$$

  The parameter η indicates the *effectiveness of piece exchange*. When $\eta = 0$, the downloaders do not get any data from each other, meaning that their upload bandwidth is completely unutilized.

  The time t_s at which the first downloaders complete their downloads can be calculated as follows

$$t_s = \frac{LR}{u(t_s)} = \frac{LR}{\frac{C_s}{x(t_s)} + \mu \eta} = \frac{LR}{\frac{C_s}{\int_0^{t_s} \lambda(t)dt} + \mu \eta}. \quad (3)$$

  Eq. (3) can be solved with numerical analysis (e.g. using the fixed point iteration method).

- *Seeders appearance phase ($t > t_s$)*

  As soon as seeders start appearing, the total upload capacity $U(t) = C_s + \mu(\eta x(t) + y(t))$ of the system at time t, divided by the file size LR of the video, can be considered the rate at which downloaders become seeders. In this case, the equations that characterize the system are

$$\begin{cases} \dfrac{dx(t)}{dt} = \lambda(t) - \dfrac{U(t)}{LR}, \\ \dfrac{dy(t)}{dt} = \dfrac{U(t)}{LR} - \gamma y(t), \\ x(t_s) = \int_0^{t_s} \lambda(t)dt, y(t_s) = 0, \end{cases} \quad (4)$$

  Hence, the average download speed at time t is:

$$u(t) = \frac{C_s + \mu(\eta x(t) + y(t))}{x(t)}. \quad (5)$$

C. System design parameters

From Eqs. (2) and (5) we can notice that the average download speed of peers is affected by some parameters which only depend on the peers and cannot be controlled, such as μ, $\lambda(t)$ and γ, and some others which are specific to the system design and as such can be opportunely tuned, namely the efficiency of piece exchange η and the capacity C_s of the initial source. More specifically, in order to make the best use of a downloader bandwidth, the underlying P2P protocol must adopt an efficient piece selection strategy. In BitTorrent
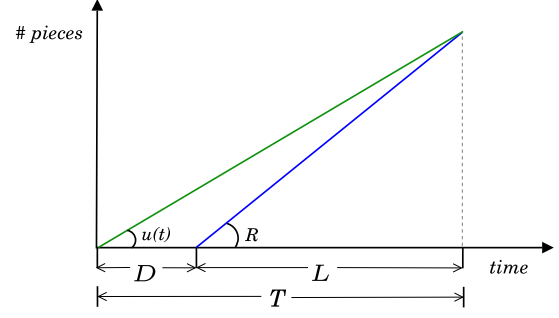

Fig. 1. Average startup delay when $u(t) < R$. The download speed and the video playback rate are represented, respectively, in terms of pieces downloaded and pieces played in time.

this is achieved by means of a rarest-first strategy, which allows downloaders to utilize, on average, 90% of their upload bandwidth [3]. However, to meet the realtime constraints of the video being played, BitTorrent-like VoD systems use a variant of this strategy, which is normally less efficient [2] (in any case, it is clear that the piece exchange efficiency cannot exceed 1). On the other hand, the service provider has much more control on the value of the initial service capacity. Based on the average peer bandwidth, peer behavior and efficiency of the P2P protocol in use, the service provider can set the appropriate value for C_s to support a certain number of peers at a given playback rate.

D. Peer behavior

Peer behavior can have a profound impact on the performance of a P2P system and, therefore, on its scalability. We will analyze this aspect by considering different peer arrival and departure patterns in our P2PVoD model.

More specifically, we assume that peers join the system at an exponentially decaying rate, as proposed by Guo et al. [5]: $\lambda(t) = \lambda_0 e^{-\frac{t}{\tau}}$, where λ_0 is the initial arrival rate at time 0 and τ is the decaying factor. The total number of peers joining the system is $\tau \lambda_0$. By varying the parameter τ, flash-crowds with different intensities can be simulated.

Analogously, the impact of different seeding behaviors can be studied by varying the value of seeder departure rate, γ, between ∞ (no seeding at all) and 0 (seeding for ever).

E. Startup delay

The average startup delay, $D(t)$, is the time a user needs to wait before playback starts, and thus it depends on the average download speed. In principle, the system could scale indefinitely, if we let $D(t)$ be large enough. In practice, if we want to guarantee an acceptable QoS, the startup delay has to meet the following requirements:

1) it should be large enough to allow the user to watch the video without subsequent stall times;
2) it should be small enough to enable the playback to start nearly in real-time.

Therefore, in our analysis, $D(t)$ is the metric that determines whether the system can scale or not. Let $T(t) = LR/u(t)$ be
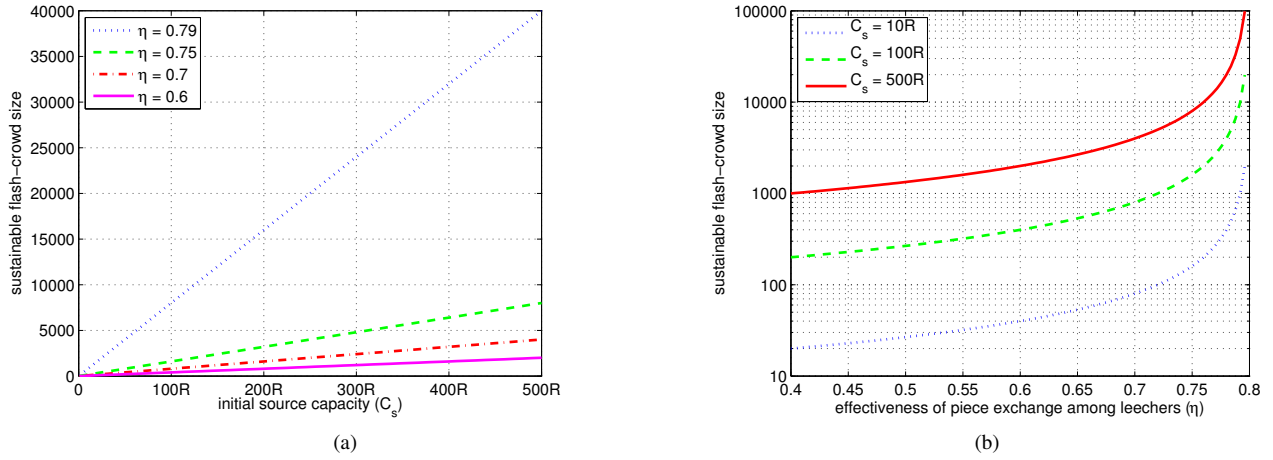
Fig. 2. Sustainable flash-crowd size as a function of (a) the capacity of the source and (b) the effectiveness of piece exchange η. $\mu = 1Mbps$, $R = 0.8\mu$, $d = 0$. In this case, Eq. (9) becomes $N = \frac{C_s}{(0.8-\eta)\mu}$, and we can observe that $N \to \infty$ as η approaches 0.8.

the expected total download time. Then the minimum startup delay that meets the QoS requirements is

$$D(t) = \max\left(0, \left(\frac{R}{u(t)} - 1\right) L\right). \qquad (6)$$

That is, when $u(t) \geq R$ the startup delay is 0, otherwise it can be calculated as the waiting time necessary to have the expected download time be no longer than the duration of the video (Figure 1).

IV. NUMERICAL RESULTS

Based on our model, in this Section we will analyze the scalability of a BitTorrent-like VoD system during a flash-crowd.

A. The influence of system design parameters

We assume that, for the QoS requirements to be satisfied, the startup delay should be no larger than a certain value $D_{\max}$. If we express $D_{\max}$ as a fraction d of the duration L of the video, then we have the following QoS constraint

$$D(t) \leq D_{\max} = dL. \qquad (7)$$

The average startup delay experienced by peers that join during the starting phase of a flash-crowd is

$$D(t) = \max\left(0, \left(\frac{Rx(t)}{\mu\eta x(t) + C_s} - 1\right) L\right) \qquad (t \in (0, t_s)). \qquad (8)$$

It is easy to see that the right most term in Eq. (8) is a strictly increasing function, since its derivative is always positive for $t \in (0, t_s)$. This means that the average startup delay is a monotonically increasing function in the starting phase, with upper bound in t_s. In the second phase of a flash-crowd, with peers becoming seeders, the average startup delay is likely to decrease (especially if seeders stay in the system long enough). Hence, we can consider the startup delay $D(t_s)$ experienced by peers joining at time t_s to be the worst case. Then, in

order to ensure that Eq. (7) is satisfied, it would be sufficient to enforce the following condition

$$D(t_s) = \max\left(0, \left(\frac{Rx(t_s)}{\mu\eta x(t_s) + C_s} - 1\right) L\right) \leq dL,$$

which can alternatively be expressed as

$$x(t_s) \leq N = \begin{cases} \dfrac{(d+1)C_s}{R - (d+1)\mu\eta} & \text{if } R > (d+1)\mu\eta, \\ \infty & \text{if } R \leq (d+1)\mu\eta. \end{cases} \qquad (9)$$

Eq. (9) illustrates the intrinsic relation between the fundamental system design parameters C_s and η and the maximum number of concurrent users N that can be supported at a playback rate R during a flash-crowd.

As we can observe from Figure 2(a) and 2(b), when $R > (d+1)\mu\eta$, N grows linearly as C_s increases and hyperbolically as $R - (d + 1)\mu\eta$ approaches 0. Hence, a natural solution to cheaply increase the system scale during a flash-crowd would be to have the playback rate R be nearly $(d + 1)\mu\eta$. However, considering that η cannot be in any case larger than 1, this approach would either lead to excessively sacrificing the quality of the video (R small) or the QoS requirement (d high). It is then evident that, for the system to be able to sustain a reasonable scale during flash-crowds, a significant amount of service capacity has to be initially provided by the source.

The importance of a high initial service capacity is also recognized by many BitTorrent communities, which seek for volunteers to seed newly released content (e.g. `http://eztv.it`). The content is normally uploaded to these peers before its official release date, which is per se an incentive to volunteer.

B. The Influence of peer behavior

In this section we will analyze the influence of peer behavior on the startup delay. Figure 3 illustrates the impact of flash-crowd intensity on the value of $D(t_s)$. The flash-crowd intensity is defined as the fraction of peers joining the system
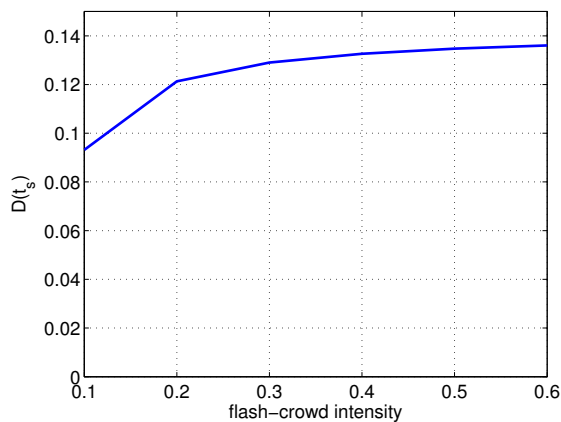
Fig. 3. Evolution of $D(t_s)$ for different flash-crowd intensities. $D(t_s)$ is expressed in terms of the duration L of the video. $L = 3600s$, $\mu = 1Mbps$, $\eta = 0.7$, $R = 0.8\mu$, $C_s = 50R$, total number of peers = 15000.
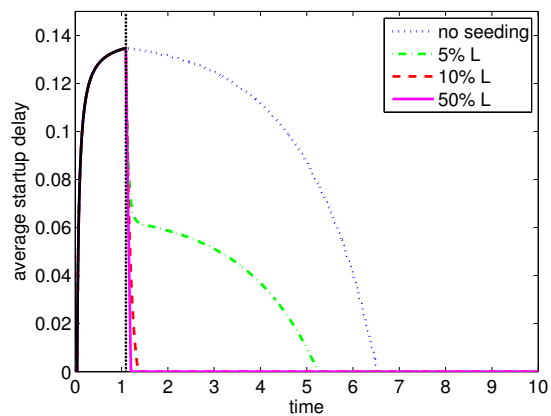

Fig. 4. Evolution of startup delay for different seeding behaviors. Both the time and the average startup delay are expressed in terms of the duration L of the video. The vertical dotted line represents the time t_s when the first seeders appear in the system. $L = 3600s$, $\mu = 1Mbps$, $\eta = 0.7$, $R = 0.8\mu$, $C_s = 50R$, total number of peers = 15000, flash-crowd intensity = 0.5.

within the first L seconds from the first peer arrival, out of all the peers that will join the system. For example, an intensity of 0.1 means that 10% of the arrivals occurs within the first L seconds from the first peer arrival.

Figure 4 shows the average startup delay of peers for different values of seeding times under a flash-crowd of intensity 0.5. As we can observe, if no peer stays in the system to seed, the average startup delay remains high for a long time. Then it starts to slowly decrease, only due to the fact that there are less and less new peers joining. However, when seeders contribute their bandwidth, even for as short a time as 5% of the length L of the video, the startup delay rapidly decreases after t_s. The situation gets better as peers start seeding for longer times, although with diminishing returns. In particular, we notice that a seeding time of 50% of the duration of the video L, is already sufficient to have the startup delay of newcomers drop to 0 almost immediately after t_s.

The above findings suggest that, once the system scale has reached a reasonable level, peer resources would be sufficient for the system to be self sustainable and scale even further. If we define $u_p(t)$ as the amount of $u(t)$ which only depends on peer contribution, then it is evident that when $u_p(t) \geq R$ the P2P system has become self-sustainable and the capacity provided by the original source is not necessary anymore,

$$u_p(t) = \frac{\mu\eta x(t) + \mu y(t)}{x(t)} \geq R \Longrightarrow \frac{y(t)}{x(t)} \geq \frac{R - \mu\eta}{\mu}.$$

That is, a reasonable seeders-to-downloaders ratio has to be reached, if we want to provide VoD with a certain level of service using solely the P2P infrastructure.

V. Conclusions

In this paper, we have analyzed the scalability of BitTorrent-like VoD systems under flash-crowds and found out that it is intrinsically related to the efficiency of the underlying P2P protocol and the initial service capacity. In particular, although it is important to make the P2P protocol as efficient as possible, it will still be necessary to provide a high initial

service capacity, in order to support many peers joining almost simultaneously. However, our preliminary results also show that, once the P2P system has reached a sufficient seeders-to-downloaders ratio, this initial service capacity is not necessary anymore, as the system has become self sustainable.

References

[1] A. Vlavianos, M. Iliofotou and M. Faloutsos. BiToS: Enhancing BitTorrent for Supporting Streaming Applications. In *Proceeding of IEEE Global Internet Symposium*, 2006.
[2] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. Rodriguez. Is high-quality vod feasible using P2P swarming? In *Proceedings of the 16th international conference on World Wide Web*, 2007.
[3] C. Herley A.R. Bharambe and V.N. Padmanabhan. Analyzing and Improving a BitTorrent Networks Performance Mechanisms. In *Proceedings of IEEE INFOCOM*, April 2006.
[4] F Liu, B. Li, L. Zhong, B Li, D. Niu. How P2P Streaming Systems Scale Over Time Under a Flash Crowd? In *IPTPS*, 2009.
[5] L. Guo and S. Chen and Z. Xiao and E. Tan and X. Ding and X. Zhang. A performance study of BitTorrent-like peer-to-peer systems. *IEEE Journal on Selected Areas in Communications*, January 2007.
[6] N. Parvez, and C. Williamson and A. Mahanti and R. Carlsson. Analysis of BitTorrent-like Protocols for On-Demand Stored Media Streaming. In *Proceedings of ACM SIGMETRICS*, 2008.
[7] D. Qiu and R. Srikant. Modeling and performance analysis of bit torrent-like peer-to-peer networks. In *ACM SIGCOMM*, August 2004.
[8] S. Tewari and L. Kleinrock. Analytical model for BitTorrent-based live video streaming. In *Proceedings of IEEE NIME Workshop*, 2007.
[9] X. Zhang, J. Liu, B. Li, and T.S. P. Yum. DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming. In *Proceedings of IEEE INFOCOM*, 2005.
[10] Y. Lu and J.J.D. Mol and F. Kuipers and P. van Mieghem. Analytical Model for Mesh-based P2PVoD. In *Proceedings of IEEE International Symposium on Multimedia (ISM 2008)*, December 2008.
[11] Y. Yang, A.L.H. Chow, L. Golubchik and D. Bragg. Improving QoS in BitTorrent-like VoD Systems. In *Proceedings of IEEE INFOCOM*, 2010.
[12] Y.Huang, T.Z.J. Fu, D.H. Chiu, J.C.S. Lui and C. Huang. Challenges, design and analysis of a large-scale p2p-vod system. In *Proceedings of ACM SIGCOMM*, 2008.