A Lifetime-Based Peer Selection Mechanism for Peer-to-Peer Video-on-Demand Systems

Igor M. Moraes and Otto Carlos M. B. Duarte
Grupo de Teleinformática e Automação - PEE/COPPE - DEL/POLI
Universidade Federal do Rio de Janeiro - Rio de Janeiro, Brazil
Email:{igor,otto}@gta.ufrj.br

***Abstract*—In this paper, we propose a peer selection mechanism for peer-to-peer video-on-demand (P2P-VoD) systems. The goal of our mechanism is to increase the availability of chunks between a peer and its partners. For this purpose, the process of selecting partners is based on the lifetime of peers, which is time since the beginning of the video playback. Thus, a peer selects as partners other peers with close lifetimes to increase the probability of finding chunks of interest in these selected partners. Results show that the proposed mechanism is efficient for different interactivity patterns. With the proposed mechanism, more than 97% of the video chunks required by a peer are available on its selected partners. This result is achieved even considering that only 10% of the video chunks can be cached by partners. In opposition, the conventional random selection mechanism requires much more disk space, which corresponds to a cache size of at least 70% of chunks, to provide the same level of availability**

I. Introduction

Peer-to-peer live-video streaming systems are a great success having millions of users today [1], [2]. Recently, peer-to-peer video-on-demand (P2P-VoD) streaming systems also have attracted the attention of researchers and developers [3], [4], [5], [6]. Most of P2P-VoD systems are based on the mesh-pull architecture[1], where a video stream is divided into chunks and each peer explicit requests the desired chunks from its partners [4]. Although VoD systems employ an architecture similar to the one typically used by live-video systems, the development of VoD systems is a challenge mainly because peers can interact with the system by stopping and jumping the video playback [3].

Basically, the operation of P2P streaming systems relies upon the establishment of partnerships among peers [7], hereafter also referred to as users or nodes. The partners are the neighbors of a peer in the overlay network and a peer only exchanges chunks with its partners [8]. In live-video systems, peers receive the video and start to playback at the instant of the request. Consequently, the request time does not often correspond to the beginning of the video. In addition, users cannot control playback actions in live-video systems. Hence, peers are interested in a given set of chunks during the same interval of time and, consequently, the playback progress is synchronized among peers. Experimental results suggest that the time lags among the playbacks of peers are unlikely higher than one minute [8]. Thus, peers easily exchange chunks with its partners because they are interested in the same part of video. On the other hand, in VoD systems, each peer may be interested in a different part of the video because peers can start to playback at anytime and can also interact with the video during its playback. Therefore, the number of peers simultaneously interested in the same part of the video tends to decrease because of the playback desynchronization among peers. As a consequence, the number of partners that have common chunks decreases.

An alternative employed by the recent P2P-VoD systems is caching [1]. In these systems, each peer contributes with a pre-determined and fixed amount of disk space as a cache of recently received video chunks. Thus, peers that are not necessarily on the same part of the video may have chunks to exchange because of the data stored in the cache. Using cache to deal with the playback desynchronization among peers, however, has some drawbacks. As the cache size on each peer is limited, cache replacement algorithms are required. In addition, the high-definition TV (HDTV) becomes popular and users want high-quality videos. The higher the video quality, the greater the amount of disk space required to store the recently played chunks. Therefore, alternatives to reduce the cache size are required.

Peer selection mechanisms can increase the availability of chunks between a peer and its partners and also reduce the cache size in P2P-VoD streaming systems. Currently, most of the systems selects and updates the partners of a peer at random [8]. The random selection is suitable for live-video systems because, as mentioned before, the lags among the playback of peers are not high. On the other hand, in VoD systems, the random selection is less efficient because peers join the system at different instants of time and control the video playback. Hence, the probability of a peer joining the system to select partners with mutual interest depends on the arrival time of the other peers at the system and the number of interactions already made by these peers.

In this paper, we propose a specific peer-selection mechanism for P2P-VoD systems, called LIPS (LIfetime-based Peer Selection). The goal of LIPS is to increase the probability of establishing partnerships among peers with more chunks of interest to exchange. The selection procedure is based on the lifetime of peers, which is the time since the beginning of the video playback. A peer selects as its partners the peers with close lifetimes compared with its own. Hence, the probability

[1] In the rest of this paper, when we refer to P2P streaming systems we are actually referring to mesh-pull-based systems.

of selecting partners interested in the same parts of video is higher than in the random selection mechanism. With LIPS, up to 99% of the video chunks required by a peer are available on its selected partners.

This paper is organized as follows. Section II discusses the works related to peer selection mechanisms for P2P streaming systems. Section III presents the system model adopted in this paper. Section IV describes the proposed mechanism. Section V presents the simulation environment used in the analysis. Section VI compares the performance of LIPS and random selection for videos with different interactivity patterns. Finally, section VII concludes this paper.

II. Related Work

There are several proposals of peer selection mechanisms for P2P streaming systems [9], [10] but the majority is not specific for VoD systems. To our knowledge, this work represents the first attempt to propose a specific peer selection mechanism for P2P-VoD systems considering user interactivity. The existing proposals for live-video systems try to improve the quality of video distribution in the overlay network instead of to increase the chunk availability among peers. Most of them are based on performance metrics related to the transmission capacity, playback continuity, and maintenance of video quality. Liang and Nahrestedt [10] propose to separate the peers of a given system in groups according to the performance characteristics of each one. Therefore, a peer only selects as its partners other peers of its group. For Liu *et al.* [9], the quality of video distribution is related to the physical organization of the overlay nodes. For these authors, nodes that are physically close have a higher probability to efficiently forward the video. Then, the goal is to construct the overlay network to optimize the physical distances among neighbor nodes. For P2P-VoD systems, we consider that the selection of partners with chunks of interest to exchange is more important than the optimization of the overlay links. We argue that it is better to have chunks to exchange using a non-optimal link instead of having an optimal link with a few chunks to exchange.

III. System Model

We assume a P2P-VoD system that works as follows. All peers of the system are cooperative and are interested in a given video, which is divided into chunks. Initially, all the chunks are only stored by the source. A peer joins the system by contacting the source. The source sends to the joining peer a subset of active nodes, called candidates. These candidates are chosen according to the peer selection mechanism employed by the system. The peer then contacts every node of the subset received. If the peer receives a positive acknowledgment from at least one of them, it successfully joins the system. The nodes that have sent back positive acknowledgment are considered partners of the new peer. The partners are the neighbors of a peer in the overlay network and a peer only exchanges buffer maps and chunks with its partners. The buffer map represents the availability of chunks in the cache of a peer. Peers store all, or part of, the received chunks in its caches and do not pre-fetch chunks that it has not requested for playback. In addition, each peer defines a window of interest, as proposed by Shah and Pâris [11]. This sliding window contains the next W chunks to be requested by a peer. With this mechanism, peers do not waste time downloading chunks that do not belong to its current window of interest, which contains the most urgently needed chunks. The chunks within the window of interest are scheduled according to the rarest-first algorithm [8].

IV. The LIfetime-based Peer Selection (LIPS)

The proposed peer selection mechanism, called LIPS (LIfetime-based Peer Selection), is specific for P2P-VoD systems. The goal of LIPS is to establish partnerships among peers with chunks of interest to exchange. The chunks of interest of a peer are the missing chunks in the range of its window of interest. As mentioned before, it is more difficult to synchronize the playback of peers in VoD systems than in live-video systems because peers can interact with the video by stopping, jumping back, and jumping forward the playback. Therefore, the number of peers that are simultaneously interested in a specific chunk of video varies and, consequently, the number of partners from which a peer can request this specific chunk tends to decrease.

The process of selecting peers with LIPS is based on the lifetime of peers. The lifetime indicates how long a peer is in the system watching a given video. Using the lifetime, we increase the probability of establishing partnerships among peers interested in the same video chunks in a given period of time, as occurs in live-video systems. Our basic assumption is that peers that joined the system at close time instants are probably interested in the same video chunks. Thus, these peers are selected as partners. In addition, if the partners are playing the same part of the video they are also interested in the same chunks at the same time. As a consequence, the previously received chunks can be discarded because both nodes have already played/forwarded this part of the video. Therefore, LIPS reduces the cache size in the nodes.

The LIPS mechanism works as follows. The lifetime of a peer is given by a counter incremented by one unit at each unit of time. Thus, LIPS does not require clock synchronization. It only requires peers to define the same unit of time to increment the counters. Each peer updates and announces its lifetime to other peers. Peers announce its lifetimes through membership messages already used by P2P streaming systems [8], thus not requiring new control messages. A new peer joins the system as described in Section III. With LIPS, the node that sends the candidates assumes that the lifetime of a new peer is zero and sends to this peer a list with the L-last peers that recently joined the system. The update of partners is periodically triggered at each t units of time. This procedure is needed because the maximum number of partners could not already been achieved or because some partners fail or leave the system. Furthermore, a peer and its partners can change its playback points and, consequently, they may have no more chunks of interest to exchange.

Algorithm 1 Partners update algorithm.

```
Require: t = update_time, A(t) > 0 and n ∈ A(t)
  newPartners ← calcNewPartners(A(t), P_n(t))
  α ← 1
  while newPartners > 0 or α < A do
    for all node i ∈ A(t) and i ∉ P_n(t) and i ≠ n do
      if (|l_n(t) − l_i(t)| < αR and size of P_n(t) <
      MAX_NUM_PARTNERS) then
        add(i, P_n(t))
        newPartners ← newPartners − 1
      end if
    end for
    α ← α + 1
    if α = A then
      add(source, P_n(t))
    end if
  end while
```

The Algorithm 1 updates the set of partners of a node n. The algorithm inputs, at update time t, are the set of active peers in the system, $\mathcal{A}(t)$, and the set of partners of node n, $\mathcal{P}_n(t)$. The algorithm has two main parts. The first one calculates the number of partners to be selected at t represented by the variable $newPartners$. The steps of this part were suppressed because the lack of space and are represented by the function calcNewPartners($\mathcal{A}(t), \mathcal{P}_n(t)$). The second part of the algorithm selects the partners of n, based on the lifetime of active peers in the system. The selection procedure employs the expanding ring algorithm. We define two parameters, the expanding factor α and the ring limit R, to determine if the lifetime of a node i, $l_i(t)$, is close enough to the lifetime of n, $l_n(t)$, according to the equation $|l_n(t) - l_i(t)| < \alpha R$. At each step of the loop, if the number of partners to be selected is not zero or α is less than the previously defined threshold A, the parameter α is incremented to expand the ring. This procedure is repeated until one of the stop conditions is satisfied. If the algorithm stops because α is equal to A, this means that the maximum allowed lag between the lifetimes of n and its partner candidates is achieved. In addition, the size of the set of partners is not the maximum possible size. Thus, we add the source to the set of partners of n and then node n can request chunks directly to the source.

V. Simulation Environment

In order to evaluate the performance of LIPS and compare it to the random selection, we developed a specific simulator, written in C++. The developed simulator generates the synthetic load to represent the interactivity behavior of peers during the video playback and also implements the peer selection mechanisms and the chunk scheduling and exchange.

The interactivity pattern of peers impacts the performance of P2P-VoD systems and, mainly, the peer selection mechanisms. According to the interactivity pattern, peers change the playback point and, consequently, modify the part of video they are interested in. Thus, the number of chunks of interest available on partners depends on the interactivity of users. In our analysis, we consider different interactivity patterns for VoD systems defined by Costa *et al.* [12]. The authors define behavior patterns for entertainment and educational videos. The parameters of these videos are extracted, respectively, from real workloads of TV UOL[2] and eTeach[3] servers. The frequency of each type of interaction and the probability distributions used to characterize the peers' behavior are presented in Tables I and II.

Table I
Frequency of interactions.

Parameters	**Entertainment**	**Educational**
Interactions per user	1.29	4.74
Pause	83%	57%
Jump backward	13%	25%
Jump forward	4%	17%

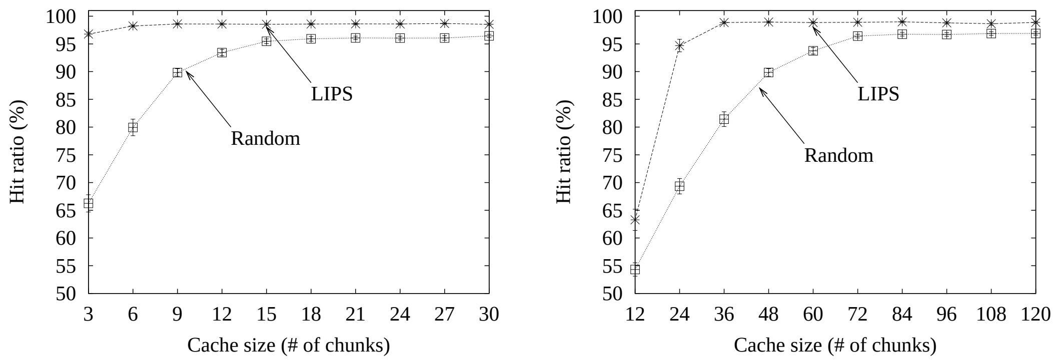
Table II
Distributions used to characterize the behavior of peers.

Parameters	**Entertainment**	**Educational**
Video length	300 s	1200 s
Peers arrival	Exponential (mean = 10 s)	Lognormal ($\mu = 3.95, \sigma = 0.95$, mean = 81.55 s)
Pause time	Weibull ($\alpha = 11.11, \beta = 0.57$, mean = 25 s)	Weibull ($\alpha = 13, \beta = 0.42$ mean = 55 s)
Jump backward distance	Exponential (mean = 20 s)	Exponential (mean = 40 s)
Jump forward distance	Exponential (mean = 7 s)	Exponential (mean = 40 s)

VI. Results

We compare the proposed mechanism, LIPS, with the random selection through simulation. We evaluate the performance of both mechanisms for the two interactivity patterns presented in Section V. The simulation parameters are the following. We assume that the playback rate is 350 kb/s for both videos and the viewing duration of a chunk is 10 s. Thus, the chunk size is 437.5 kB. The entertainment and the educational videos are composed of 30 and 120 chunks, respectively. During the simulation, 50 peers arrive at the system to watch the videos. We assume that peers start the playback from the beginning[4] of the video and none of them fails or leaves the system during the playback. We define four peer upload capacities according to the experimental results presented by Huang *et al.* [3]. Peers can contribute to the system with 200 (36%[5]), 360 (28%), 600 (25%), and 1000 kb/s (11%). For both mechanisms, LIPS and random selection, the maximum number of partners is equal to 4 [8]. The size of the window of interest is equal to 10% of the video length in terms of chunks. Thus, we have 3 and 12 chunks, respectively, for the entertainment and the educational videos. The window size is calculated according to the equation proposed by Shah and Pâris [11]. The update period is equal to 10 s. Furthermore, if there is a worst partner, it will be replaced at each update. The worst partner is one of the partners without chunks of interest

[2]The TV UOL is the VoD service of the largest ISP in Latin America.
[3]Placed at University of Wisconsin-Madison.
[4]More than 98% of users start the video playback from the beginning [12].
[5]The percentage of peers in the system with this capacity.



(a) Entertainment (video length is 30 chunks).

(b) Educational (video length is 120 chunks).

Figure 1. The hit ratio as a function of the cache size.

at the update time. The parameters of LIPS are the following. The ring limit R is defined based on the mean interarrival time of peers defined by each video. If R is smaller than this interval, the probability of selecting partners in the first step of the expanding ring procedure is low. Thus, R=10 s to the entertainment video and R=80 s to the educational video. The expanding factor threshold is $A = 5$ for both videos. The same interactivity pattern generated at each simulation run is applied to both mechanisms. For every point of the curves, we calculated the confidence interval for a 95% confidence level.

The cache size is fixed and equal for all peers. A cache of L chunks means that a peer can store until L chunks of the video in its hard disk. The most recently received chunks are cached, i.e. the ones belonging to the current window of interest and the previous $L - W$ chunks from the beginning of the window, where W is the window size.

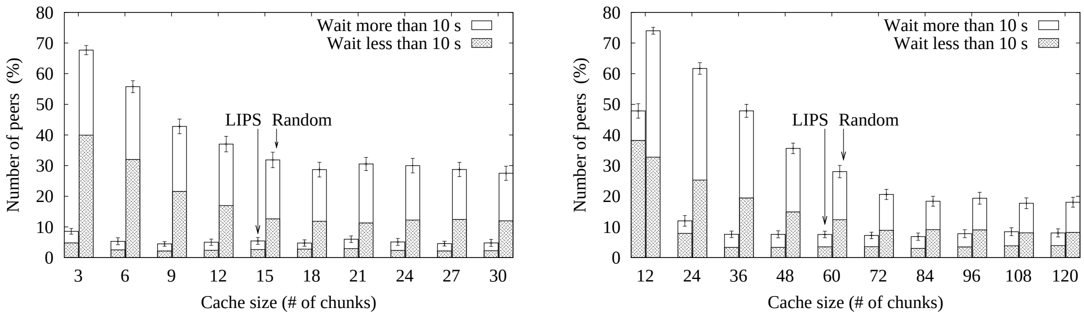
The performance metrics are evaluated as a function of the cache size that ranges from 10% to 100% of the video length in terms of number of chunks. The more the amount of chunks cached, the higher the probability of peers having chunks of interest to exchange. This expected behavior is confirmed by Figure 1 that presents the hit ratio as a function of the cache size. Each point of the curves represents the mean value of this metric for all peers of the system. We define the hit ratio as the percentage of chunks of interest a peer finds in the cache of its partners, excluding the source. The chunks of interest of a peer are the missing chunks in the range of its window of interest. The hit ratio indicates the efficiency of partners selected by each mechanism. The higher the hit ratio, the easier to obtain chunks of interest without the help of the source.

Figure 1 shows that LIPS provides a higher hit ratio than the random selection for both types of videos regardless of the cache size. For the entertainment video, Figure 1(a), LIPS achieves a hit ratio higher than 98% considering a cache size of only 6 chunks, which corresponds to 20% of the chunks of this video. On the other hand, this hit ratio is not achieved by random selection. This mechanism requires a cache size of 30 chunks, or 100% of chunks, to provide its maximum hit ratio, 96.8%. The same hit ratio value is provided by LIPS when the cache size is only 3 chunks. Thus, for the entertainment video, our proposal provides a higher hit ratio than the maximum one provided by the random selection and, at the same time, saves up to 80% of disk space. For the educational video, Figure 1(b), the maximum hit ratio provided by LIPS is about 99%. This ratio is achieved for a cache size of 36 chunks, which corresponds to 30% of the chunks of this video. Once again, this hit ratio is not achieved by random selection even if all the chunks could be stored. The maximum hit ratio provided by random selection is 97% considering a cache size of more than 84 chunks, or 70% of the chunks. In this case, LIPS achieves a higher hit ratio than the maximum one of the random selection and saves up to 40% of disk space.

We also conclude that the video popularity impacts the performance of LIPS. The more popular the video, the higher the number of simultaneous peers in the system and, consequently, more partner candidates a peer has. Thus, it is easier to find partners with chunks of interest. During simulations, the number of simultaneous peers is up to 36 and 22 for entertainment and educational videos, respectively. Thus, the entertainment video is more popular than the educational one. In addition, the interarrival time of peers at the system for more popular videos is lower than for the less popular ones. Therefore, for more popular videos, fewer chunks must be cached because the difference between the playback points of peers tends to be smaller.

The better selection of partners provided by LIPS is explained as follows. The probability of a peer joining the system to select partners with chunks of interest depends on two factors: the arrival time of the other peers at the system and the number of interactions already made by these peers. The random selection does not take into account these factors. On the other hand, LIPS selects partners based on the time the peers are in the system. Therefore, the probability to establish partnerships among peers with chunks of interest increases, because with LIPS this probability depends only on the interactions of peers. According to the results, the selection of partners simply based on the lifetime improves the system performance compared with the random selection for the analyzed interactivity patterns.

The playback continuity is one of the most critical metrics to determine the users' satisfaction. The higher the hit ratio, the higher the number of chunks of interest available on the



(a) Entertainment (video length is 30 chunks).

(b) Educational (video length is 120 chunks).

Figure 2. The number of waiting peers as a function of the cache size.

selected partners. Thus, maintaining the playback continuity becomes easier. This behavior is ratified by Figure 2 that shows the number of waiting peers and the waiting time of peers as a function of the cache size for both mechanisms. Waiting peers are the peers that have no chunk related to the video content of the current playback point. Consequently, the playback of these peers is paused until they receive the missing chunk or a timeout expires. Waiting time is the total time the video playback is paused because of the absence of chunks needed to the video playback. The striped and the blank parts of each vertical bar indicate, respectively, the number of peers that wait less and more than 10 s to receive the missing chunks. This is the duration of a chunk. The sum of both parts represents the total number of waiting peers. LIPS is always represented by the first bar of the group of two bars plotted for each cache size. For both videos, LIPS provides higher playback continuity than the random selection. With LIPS, only 5% of peers watching the entertainment video wait for a missing chunk considering a cache size greater than 6 chunks whereas, with the random selection, up to 56% of the nodes wait for a missing chunk. Even for a cache size of 30 chunks and a hit ratio of 96.8%, 27.5% of peers watching the entertainment video experience an interruption on the playback. For the educational video and a cache size greater than 24 chunks, the percentages of waiting peers are 12% and 62%, at most, for LIPS and random selection, respectively. For the maximum cache size and the random selection, 20% of peers still experience an interruption on the playback. Furthermore, most of peers employing random selection waits more than 10 s to receive a missing chunk. Thus, with random selection, more nodes wait for more time when compared with LIPS.

VII. Conclusion

In this paper, we proposed the LIfetime-based Peer Selection (LIPS) mechanism. LIPS selects partners based on the lifetime of peers. Thus, the probability to establish partnerships among peers with chunks of interest increases, because with LIPS this probability depends only on the interactions of peers. The results show that the partners selected by LIPS are more efficient than the ones selected by the random selection for different interactivity patterns. With LIPS, up to 99% of the video chunks required by a peer are available on his selected partners. In contrast, the random selection only achieves 97% of hit ratio if a cache size of at least 70% of all chunks is employed. To provide a hit ratio higher than the maximum one provided by the random selection, LIPS needs that peers store only 20% of the video chunks. Thus, the proposed mechanism saves up to 80% of disk space. LIPS also provides higher playback continuity than the random selection. With LIPS, only 7.6% of peers, at most, experience an interruption on the playback for a cache size greater than 30% of the video chunks. For the same cache size, with the random selection, up to 50% of the nodes wait for a missing chunk.

Acknowledgment

This work was supported by CNPq, CAPES, FINEP, FUNTTEL, and FAPERJ.

References

[1] J. Liu, S. G. Rao, B. Li, and H. Zhang, “Opportunities and challenges of peer-to-peer Internet video broadcast,” *Proc. IEEE*, vol. 96, no. 1, pp. 11–24, Jan. 2008.

[2] I. M. Moraes, M. E. M. Campista, J. L. Duarte, D. G. Passos, L. H. M. K. Costa, M. G. Rubinstein, C. V. N. de Albuquerque, and O. C. M. B. Duarte, “On the impact of user mobility on peer-to-peer video streaming,” *IEEE Wireless Commun. Mag.*, vol. 15, no. 6, pp. 54–62, 2008.

[3] Y. Huang, T. Z. J. Fu, D.-M. Chiu, J. C. S. Lui, and C. Huang, “Challenges, design and analysis of a large-scale P2P-VoD system,” in *ACM SIGCOMM*, Aug. 2008, pp. 375–388.

[4] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. Rodriguez, “Exploring VoD in P2P swarming systems,” in *IEEE INFOCOM*, May 2007, pp. 2571–2575.

[5] C. Huang, J. Li, and K. Ross, “Can Internet video-on-demand be profitable?” in *ACM SIGCOMM*, Aug. 2007.

[6] N. Vratonjic, P. Gupta, N. Knezevic, D. Kostic, and A. Rowstron, “Enabling DVD-like features in P2P video-on-demand systems,” in *Peer-to-Peer Streaming and IP-TV Workshop - P2P-TV*, Aug. 2007.

[7] D. A. G. Manzato and N. L. S. da Fonseca, “Incentive mechanism for the CoopNet network,” *Peer-to-Peer Networking and Applications*, vol. 1, no. 1, pp. 29–44, Mar. 2008.

[8] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, “CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming,” in *IEEE INFOCOM*, Mar. 2005, pp. 2102–2111.

[9] Z. Liu, Y. Shen, S. Panwar, K. Ross, and Y. Wang, “On scalability of proximity-aware peer-to-peer streaming,” in *IEEE INFOCOM*, May 2007, pp. 2561–2565.

[10] J. Liang and K. Nahrstedt, “RandPeer: Membership management for QoS sensitive peer-to-peer applications,” in *IEEE INFOCOM*, Apr. 2006, pp. 1–10.

[11] P. Shah and J. F. Pâris, “Peer-to-peer multimedia streaming using BitTorrent,” in *IEEE IPCCC*, 2007, pp. 340–347.

[12] C. Costa, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. Ribeiro Neto, “Analyzing client interactivity in streaming media,” in *Intl. World Wide Web Conference - WWW*, Aug. 2004, pp. 534–543.