Scalability and Efficiency of Push-Driven P2PTV Systems

Cyril Cassagnes, Damien Magoni, Hyunseok Chang, Wenjie Wang, and Sugih Jamin

***Abstract*—Television transmitted over IP (IPTV) presents numerous opportunities for users as well as service providers, and has attracted significant interest from industry as well as research communities in recent years. Among the emerging IPTV delivery architectures, the peer-to-peer based delivery mechanism is considered attractive due to the relative ease of service deployment and potential bandwidth savings. However, the question of how well P2PTV networks would support a growing number of users has not been fully investigated so far. In this paper, we try to address this question by studying scalability and efficiency factors in a typical P2P based live streaming network. Through the use of the data provided by a production P2PTV network, we carry out simulations whose results show that there are still hurdles to overcome before P2P based live streaming could become widely deployed.**

***Index Terms*—P2PTV, overlay network, peer churn, peer selection.**

I. Introduction

With the increasing broadband speed and continued improvement in video compression technologies, Internet-based television (IPTV) services have been experiencing sustained growth lately. When it comes to realizing IPTV services in today's Internet, peer-to-peer (P2P) based delivery mechanism is considered an attractive option because of the ease of deployment and potential bandwidth savings.

In a typical P2P based IPTV network, clients retrieve video streams by connecting to the broadcast server or any other existing clients that are already connected to the network. The broadcast server generates packetized video streams by encoding live TV signals captured from satellite. After joining the network, clients can contribute their uplink bandwidth by forwarding the incoming video streams to other clients needing those streams. To allow more efficient utilization of client's uplink bandwidth, the video streams are typically distributed via the P2P network in the unit of *chunks* (e.g., [1]) or *sub-streams* (e.g., [2], [3]). Chunks are time-divided segments of packetized streams, while sub-streams are space-divided subsets of the original streams (e.g., layers in H.264 SVC). The chunks or sub-streams are either *pushed* by forwarding clients, or *pulled* by receiving clients, depending on the P2P sharing protocol used. In the pull-driven delivery, clients search and pull individual stream units in an opportunistic way, while in the push-driven approach, a client establishes a *virtual* connection to a forwarding client, and continues to receive data pushed from the forwarder until either end terminates the connection. Push-driven delivery design was shown to be more efficient than pull-based counterpart in recent work [4].

Compared to traditional P2P data sharing or progressive streaming of video on demand, optimizing end-user experience in the P2P based live streaming environment is a non-trivial task because of its more stringent delay constraint and limited shared buffer space. In addition, upload capacity constraints and inherent churning behavior of participating clients can add to the difficulty in realizing a fully scalable delivery system. Motivated by our earlier studies on the operational scalability of P2P based live streaming [5], [6], we explore the impact of various peer selection algorithms as well as various overlay configuration parameters on the system performance. Configuration parameters includes the number of sub-streams, the buffer capacity, and the number of search attempts. In our study, we focus on the *push-driven, sub-stream based* streaming architecture, and perform detailed simulations instantiated with the data contributed by a production system employing such an architecture. Upload capacity and session length are important parameters for the P2P network as they determine its scalability and churn.

The remainder of this paper is organized as follows. Section II outlines the previous work done on P2P based live streaming systems. Scalability of such P2PTV networks relies on many characteristics whose impacts on the networks have not been analyzed in depth, to the best of our knowledge. To provide a realistic and fined-grained study of P2P live streaming, Section III presents realistic parameters derived from the analysis of 9.8M sessions collected by the professional-grade Zattoo P2PTV network. Section IV presents evaluation results obtained by simulating the architecture of the Zattoo P2PTV which is one of the largest production P2PTV providers in Europe and identifies the parameters with the most effect on the scalability and efficiency of P2PTV networks. Finally, we conclude the paper by summarizing our contributions and presenting future research directions in Section V.

II. Related Work

With the P2PTV's growing popularity, new P2PTV systems have been proposed, which include ZigZag [7], PRO [8], Anysee [9], and PULSE [10], [11]. Besides such proposals, a large number of measurement papers on existing P2P live

Manuscript received August 25, 2011; revised September 20, 2011.

C. Cassagnes and D. Magoni are with the University of Bordeaux, France (e-mail: {cassagne,magoni}@labri.fr).

H. Chang is with Bell Labs, USA (e-mail: hyunseok@ieee.org).

W. Wang is with PPLive, China (e-mail: wenjiew@gmail.com).

S. Jamin is with the University of Michigan, Ann Arbor, USA (e-mail: jamin@eecs.umich.edu).

streaming systems have been published as well, characterizing workloads [12] or specific systems such as Telefonica IPTV service [13] and CoolStreaming [14], [15].

Being one of the most popular operational P2PTV systems, PPLive has been extensively studied by researchers [16], [17], [18]. Silverston et al. [19] characterized various P2P IPTV traffic generated by PPLive, PPStream, Sopcast and TVants, in terms of the transport-level protocols used and session behaviors.

In addition, numerous research proposals for improving existing P2P-based live streaming systems have been presented, ranging from feasibility study [20] and upload capacity analysis [21], to locality awareness algorithm proposal [22] and stochastic fluid theory for P2P streaming systems [23]. Many proposals based on robust incentives [24], altruism [25], contribution awareness [26] and sub-stream trading [27] aim at avoiding free-riders in large scale systems that have appeared. Most recent works explore topological properties of practical P2P streaming [28]. Small et al. [29] address the subject from a theoretical perspective and propose a heuristic called *Affinity* which considers in its neighbor selection process multiple parameters such as upload capacity, playback delay and bandwidth cost, in order to build optimal P2P topology for live multimedia streaming.

III. Peer Characteristics Measured in a Real P2PTV Network

In order to carry out realistic simulations and to achieve a fine-grained analysis of the scalability and efficiency of a P2PTV network, we must take into account many peer characteristics. To this end, we collected data from the Zattoo's P2PTV network which is a push-based P2P streaming network. In Zattoo, peer selection is based on delay measurements between peers, as well as their topology and geographic information (e.g., IP address, AS number, country, etc.). The network covers eight European countries, serving over 3 million registered users. The data used in our analysis originate from a Zattoo's session database collected during a two-week period from March 10th to 24th, 2008. Each session in the database records user's stream watching behavior including start/end timestamps, number of bytes uploaded/downloaded, etc. The number of sessions recorded amounts to 9.8 million sessions covering 198 channels and 8 countries. From the collected data, we identify four main peer characteristics that may impact a P2PTV network:

1) Session length (Section III-A).
2) Inter-arrival time (Section III-B).
3) Upload capacity (i.e., redistribution factor, Section III-C).
4) NAT compatibility (Section III-D).

A. Session length

Figure 1 shows the distribution of session lengths with the x-axis using a log scale. As already observed in several previous works [13] [16] [17], its Cumulative Distribution Function (CDF) fits a log curve. On this figure, we can observe that 80% of the peers have sessions shorter than 10 minutes. This is a typical value also observed in prior works, but it can vary depending on both channel contents (e.g., news-only, movie-only channels, etc.) and users' channel surfing behavior. It is important to note that the shorter the session length is, the higher the rates of churn is. High churn rate could have an adverse effect on the stability of the overlay.

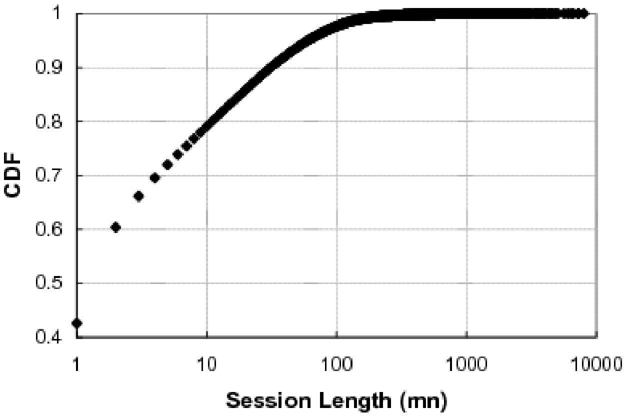


Fig. 1. Distribution of the peers' session length.

B. Inter-arrival time

Figure 2 shows the distribution of the inter-arrival times measured for one popular channel. We have used a log scale for the x axis. We can observe that the CDF distribution roughly fits a log curve for inter-arrival times between 1 and 6 seconds. On this figure, we can locate the knee of the distribution at roughly 10 seconds for the vast majority of the peers (i.e., 96%). This means that the start times of the sessions are often very close to each others suggesting flash crowd patterns. It is not common that two following starting sessions are separated by a long time interval. For this popular channel the maximum inter-arrival time recorded was 300 seconds but this value can vary depending on both channel content and user behavior.

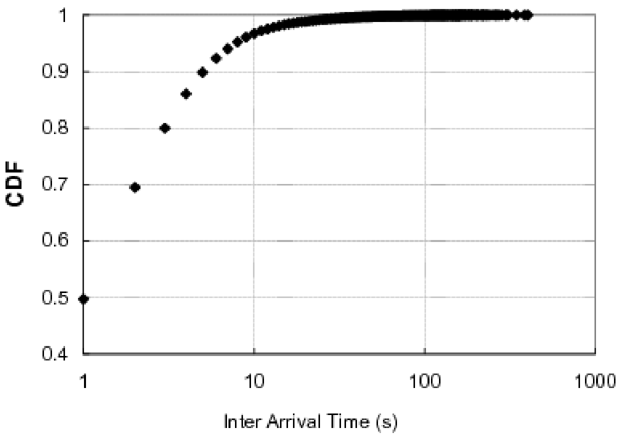


Fig. 2. Distribution of the inter-arrival time.

C. Upload capacity

We define the ratio of one incoming stream that can be redistributed to other peers as a *redistribution factor*, and label it with k. The *redistribution factor* k may take a value from 0 to infinity depending on the peer's uplink capacity. For instance, if $k = 1$, it means that the peer can redistribute a full stream, whereas if $k = 2$, it means that the peer redistributes two copies of the stream. Fractional values are also possible as a full stream can be divided into multiple sub-streams. Subdividing a stream allows a peer to redistribute only a subset of the stream to other peers; for instance, if $k = 0.5$, it means that the peer redistributes only half of the stream due to its uplink bandwidth constraint or buffer availability. The maximum number of peers able to connect to the system will depend on the average value of k: if it is below 1, the system can not scale and the overlay will reach a maximum size.

Figure 3 shows the distribution of the ratio k, where k is the upload rate divided by the download rate. Upload rate is measured between the peer and a specific server located in Europe. As bandwidth measurement depends on many factors and is very difficult to do, the values reported here may not be accurate. The CDF distribution looks like a log plot although it does not fit a log curve. The distribution is therefore highly heavy-tailed.The average value of k computed from all the distribution values is equal to 0.89, thus still below 1 (and its value is not significant since the distribution is not normal). We found that 50% of the peers can redistribute less than 50% of the full stream (i.e., $k < 0.5$), while 82% of the peers can redistribute less than the full stream (i.e., $k < 1$). A P2PTV network relying solely on these redistribution values cannot scale.

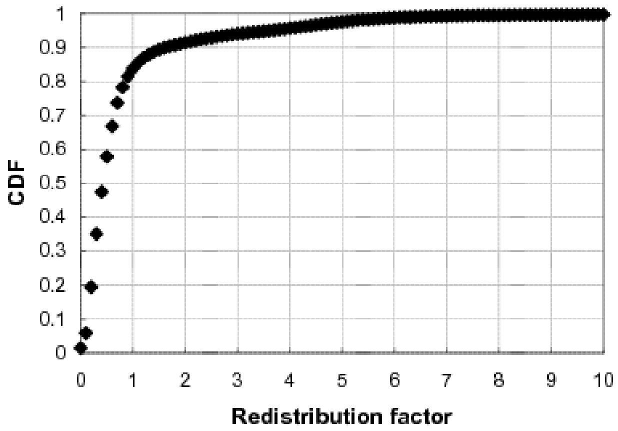
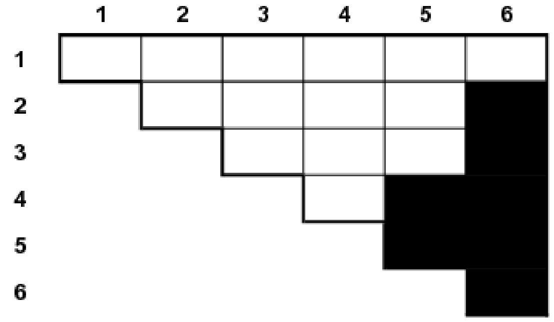


Fig. 3. Achievable redistribution factor.

D. NAT compatibility

In today's Internet, the majority of peers are behind some type of NAT gateways which can in some cases block the communication between two peers [30]. Therefore a typical P2P sharing software implements a NAT traversal method [31] in order to facilitate communication between peers behind NAT gateways. Zattoo's NAT traversal process can detect six different NAT configurations: open host, full cone, IP-restricted, port-restricted, symmetric, and UDP-disabled. Each NAT configuration is assigned a distinct NAT type number, from 1 (open; the least restrictive type) to 6 (UDP-disabled; the most restrictive type).



	1	2	3	4	5	6
1	□	□	□	□	□	□
2		□	□	□	□	■
3			□	□	□	■
4				□	■	■
5					■	■
6						■

Fig. 4. NAT type compatibility matrix.

Figure 4 shows the reachability among the various NAT types. Consider a pair of peers whose NAT types are represented on the row and column of the matrix. Notice that the matrix is symmetric; it does not matter whether the parent peer or the child peer is represented in the row or column. At the intersection lies the reachability. If the cell is white, the peers can connect to each other. If the cell is black, the peers cannot connect to each other.

Figure 5 shows the distribution of the NAT types of the peers. Most peers have a NAT type of 4 or 5. Some peers have a NAT type of 1, 2, or 6. There is a negligible number of peers with NAT type 3.

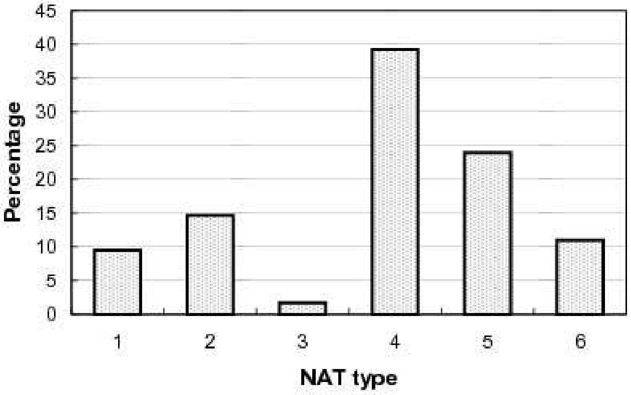


Fig. 5. Distribution of the peers' NAT type.

Given the previous distribution and the matrix, the first chance connectivity probability for a peer to connect at a given layer of depth $n > 1$ is around 56% (total area minus black area divided by total area, $p_1 = (10000 - (1800 + 2000 + 625))/10000 = 0.5575$). On the other hand, if the peers try to optimally connect to each other in order to maximize redistribution, the average connectivity probability for the peers to connect to a layer of depth n is around 90%. In this optimal configuration, all type 6 peers connect to type 1 peers, 60% of type 5 peers connect to type 2 peers, all type 4 peers connect to type 4 peers, and all type 2 peers connect to type 5 peers. The remaining 40% of type 5 peers cannot connect to anyone, thus giving 10% chance of connection failures.

IV. Simulation of a P2P Live Streaming Overlay

In this section, we propose several enhancements in constructing a P2P overlay and we evaluate them by carrying out detailed simulations.

A. Simulation parameters and metrics

Our simulation code faithfully implements Zattoo's peer-division multiplexing protocol for P2P based streaming [3]. The simulation is then performed using the *ne*twork *m*anipulator software called *nem* [32] which is driven by a packet level discrete event engine. An Internet map consisting of 4,200 nodes is used as the underlying topology [33]. Each simulation experiment lasts for 12 hours, and analysis is done only for the last 6 hour period when the system is in a steady state regime. We repeat each experiment 30 times, and we report the average taken from those 30 runs. We draw from the peer characteristics described in Section III to set the following input parameters in our simulation: (i) session length, (ii) NAT type, and (iii) upload capacity (k factor). We randomly instantiate these parameters so that the resulting distributions match those reported in Section III. Table I shows the remaining input parameters.

TABLE I
Simulation parameters.

Parameters	Values
Number of simulation runs per scenario	30
Source capacity	50 peers
Maximum size of candidate peer list	40 peers
Join timeout period	0.25 sec.
Search period	2 sec.
Number of substreams per stream	1, 2, 4, 8, 16
Buffer capacity	3, 6, 12 sec.
Number of peer search attempts	1, 2, 4

Our simulation models Zattoo's peer-division multiplexing based P2P streaming network, except that peer selection is purely random. We consider alternative peer selection algorithms to investigate their impact in Section IV-C. In order to assess the performance of the P2P live streaming system, we study the following output metrics.

1) **View time ratio:** This metric is calculated by peer's view time divided by the peer's life time. From this metric, one can infer how much time is devoted to joining the P2P overlay and searching for available streams before starting to watch a given channel.
2) **Ratio of kicked out peers:** This metric is calculated by the number of peers that could not connect to the P2P overlay during a given period, divided by the total number of new peers joining during the period.
3) **Average number of interruptions per peer:** This metric is calculated by the number of video viewing interruptions for all peers in a given period, divided by the total number of new peers joining during the period.

B. Influence of peer characteristics on simulation results

We analyze the above output metrics by varying the number of newly joining peers per hour, which is defined as the traffic load on the P2P network. We assume that a stream consists of 16 substreams, and that each peer's buffer can store 6 seconds' worth of streaming data. Furthermore, each peer is assumed to attempt two rounds of searches when the peer detects any missing substream in its buffer.

1) Influence of the session length: Figure 6 shows the average view time ratio of peers with different traffic load (i.e., number of newly added peers per hour). To observe the influence of the session length on view time ratio, we set the redistribution factor k to 1. Different NAT configurations are not taken into account in this experiment. Three sets of simulations are performed, one with a constant session length of 12 minutes, another with 24-minutes, and finally with variable session lengths following the distribution shown in Figure 1. Twelve minutes is the average value of the distribution shown in Figure 1. The small difference in the session length curve is due to the necessary startup join time to connect to the network. When sessions are shorter, this startup join time increases, and is no longer negligible, as is the case of the real session length distribution (i.e., around 4%). It turns out that regardless of the session length distribution, the view time ratio is close to 100%. So the number of interruptions is minimal and the short session length allows the overlay not to saturate. We do not show the amount of kicked out peers as it is equal to 0 for all these simulations with $k = 1$.

2) Influence of the upload capacity: Figure 7 shows the average view time ratio of peers with different traffic load. To highlight the influence of the redistribution factor k on view time ratio, we use a constant session length of 12 minutes. As in the previous case, different NAT configurations are not taken into account in this scenario. We run 3 sets of simulations, with fixed redistribution factors $k = 1$ and $k = 2$, as well as variable k following the distribution shown in Figure 3. The figure shows that when realistic redistribution factors k are used, the overlay does not scale. That is, the view time ratio falls rapidly as the traffic load increases. Note that the traffic load is a loose underestimate of the network capacity (as it is computed with the average session length). We can see in Figure 7 that the network starts to get saturated with a traffic load around 500, but not 250, because most of the sessions are much shorter than the average 12 minute length, and thus more sessions are needed to overload the network.

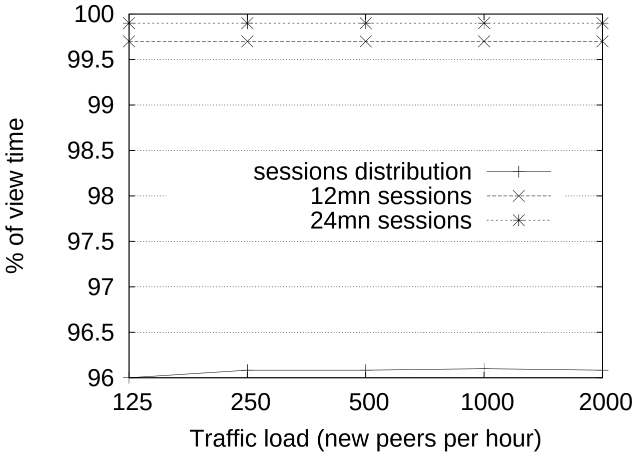


Fig. 6. Average peer view time vs. traffic load.

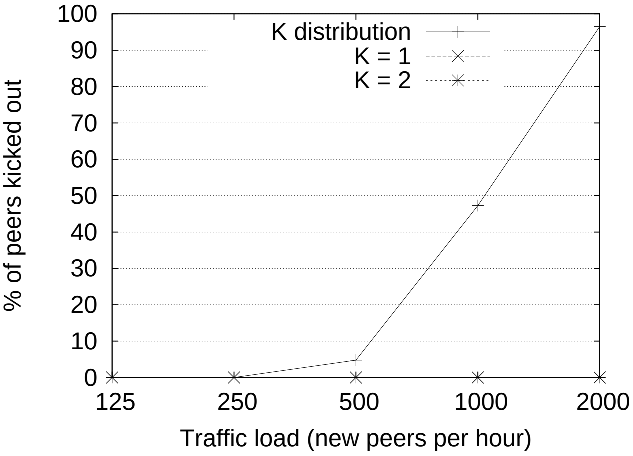


Fig. 8. Number of peers kicked out vs. traffic load.

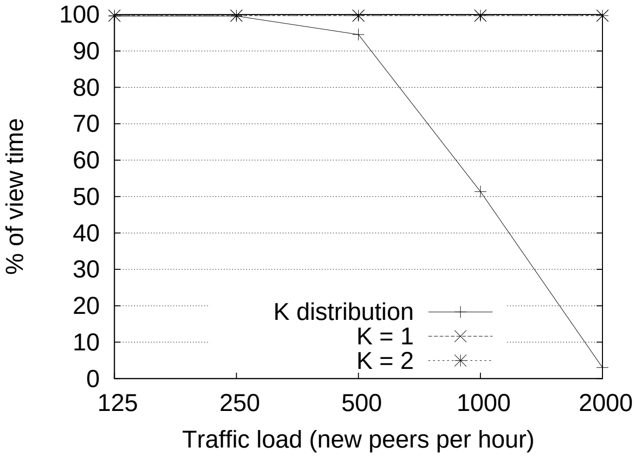


Fig. 7. Average peer view time vs. traffic load.

We observe that at a traffic load of 1,000, only 50% view time ratio is achieved. At a load of 2,000, view time ratio is as low as 5%. At this load, most peers cannot connect to the network as it is completely saturated. The view time ratio is close to 100% for $k = 1$ and $k = 2$ (as already seen above for $k = 1$). This result illustrates the importance of having peers with $k > 1$ for the P2P overlay to scale.

In order to see if the view time ratio decrease is due to a higher startup join delay or because peers cannot connect to the overlay, we look at the number of peers that are kicked out of the network. When peer's view time ratio is very low (e.g. below 1%), it may be because the peer could not get the stream and left prematurely). Figure 8 shows the average number of peers kicked out as a function of the traffic load. To observe the influence of the redistribution factor k on the number of kicked out peers, we use the same parameters as above. The ratio of kicked out peers is equal to 0% for $k = 1$ and $k = 2$. However, when we use the real distribution of k, the number of kicked out peers increases rapidly as the traffic load increases. Figures 7 and 8 together show that the decrease of view time ratio is mainly due to the increase in the number of kicked out peers. These results confirm that the real k distribution is a limiting factor that prevents the overlay from being scalable in itself.

3) Influence of the NAT compatibility: Next, we examine the average view time ratio when we use realistic redistribution factors k and session lengths taken from Figures 3 and 6 respectively. Figure 1 shows the average peer view time ratio as a function of the traffic load. We run two sets of simulations, capturing two different scenarios. In the first scenario, every peer has a NAT type 1 configuration, while in the second scenario, each peer is assigned a NAT type taken from the realistic NAT type distribution shown in Figure 5. When the number of peers is small, everybody can connect directly to the source, and the view time percentage is close to 100%. When the number of peers increases, peers have to connect to each other to create a P2P network. Due to the NAT compatibility issues and the limitations caused by the k factor, the view time ratio gradually decreases to 75% and 60% for the first and second scenarios respectively. The relative gap between the first and second scenarios depends on the traffic load. It starts at close to 0% when the load is 250 or less, and then increases up to roughly 15% when the load is 4,000 peers per hour. In the simulations, peers perform only two rounds of searches for available peers before quitting, which explains why the relative difference between the two scenarios is much more than the theoretical 10% defined at the end of Subsection III-D. The latter figure can be reached only after a sufficient amount of tries. With only one try, there is a relative difference of 43% on average between the two. Also, due to the upload capacity limitation, each peer redistributes only a part of the full stream, resulting in the view time ratio decreasing to 75% at traffic load 4,000 in the first scenario. Results in this plot and the next one are different from the ones presented in Figure 7 and Figure 8 because here the session length of the peers is not fixed but instantiated realistically by following the probability distribution shown in Figure 1. The values of the randomly set lifetimes are on average much shorter than the fixed 12 minutes used in the previous subsection. That explains why the percentage of viewing time is much higher in Figure 9

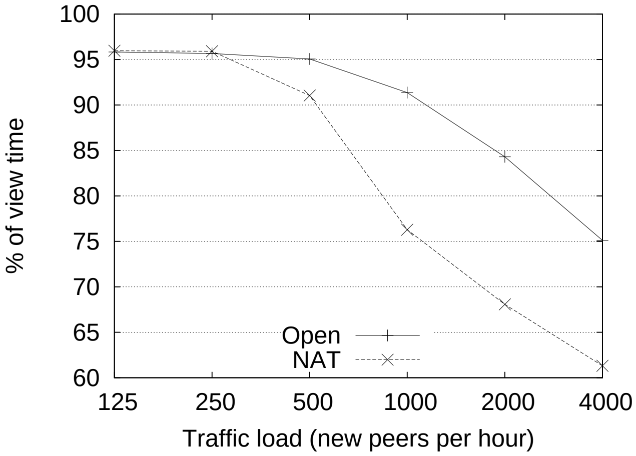


Fig. 9. Average peer view time vs. traffic load.

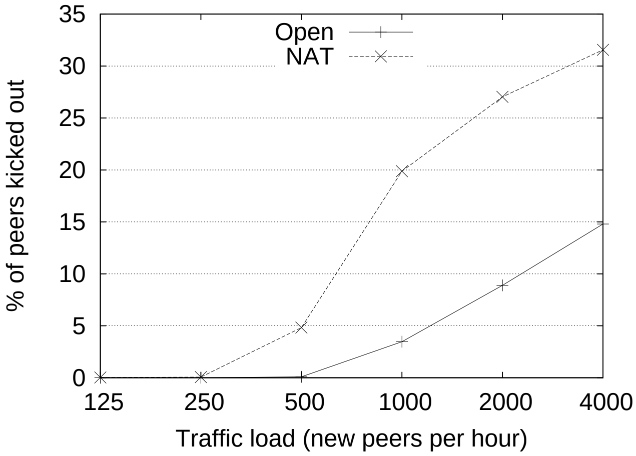


Fig. 10. Number of peers kicked out vs. traffic load.

than in Figure 7 and why the percentage of kicked-out peers is much lower in Figure 8 than in Figure 10. As the peers remain on average less time in the overlay, the traffic load must be much higher to saturate the overlay in order to prevents new peers to join and view the stream.

As before, we consider the average number of kicked out peers that account for the decrease of the view time ratio. Figure 10 plots the average percentage of peers that could not connect to the P2P network as a function of the traffic load. We use the same parameters as in the previous figure. When the load is equal to or less than 250, every peer can connect directly to the source, and there is no kicked out peer. When the number of peers increases, the number of kicked out peers with realistic NAT type distribution gradually increases to around 26% for a load of 2,000 and 31% for a load of 4,000. As the view time ratio is equal to 61% at a traffic load of 4,000, we can conclude that the decrease in view time ratio is mainly accounted for by the kicked out peers. Only a small part is due to the startup join time. When all peers have NAT type 1 configuration, they can connect to each other. Therefore, although the number of kicked out peers does increase due to the k factor, it is always relatively 15% or more below the kicked out peers compared to the realistic NAT type case. Based on all these results, we conclude that in a P2PTV overlay network, the NAT compatibility issue can create a non negligible loss of peers.

C. *Influence of peer selection algorithms on simulation results*

Next, we present the results that highlight the impact of peer selection algorithms on the performance of a typical P2P live streaming system. When a new peer joins an overlay, it performs its own peer selection algorithm to choose the target peer(s) to connect to. If the peer selection is done randomly, the resulting overlay could become quite inefficient in two perspectives. First, the overlay could become too deep and not wide enough, thus incurring large playback lags. Secondly, the overlay could experience a lot of churns, thus incurring frequent playback interruptions for users.

Intuitively, P2PTV overlays could be made more efficient by placing more stable and high bandwidth peers closer to the source. The peer selection algorithm is a good place to influence the evolution of the overlay as we can more or less control where the peers will place themselves in the overlay. If a peer selection algorithm manages to put stable peers close to the source, this should reduce the overall churn in the overlay. Also, if this algorithm manages to put high bandwidth peers close to the source, this should increase the capacity of the overlay while keeping a reasonable depth in the overlay.

A carefully designed peer selection algorithm should improve the efficiency of a streaming overlay by incorporating dynamic parameters such as upload capacities, session lengths, distances among peers and overlay depth positions. Such modifications may impact both P2PTV overlays and peers.

According to the session data collected by Zattoo, the CDF of the redistribution factor k follows an exponential distribution, where 50% of the peers can redistribute less than 50% of the full stream (i.e., $k < 0.5$), and 82% of the peers can redistribute less than the full stream (i.e., $k < 1$). Following this observation, we assign uplink capacity to individual peers so that the resulting uplink distribution becomes identical to the empirical distribution. The NAT type of a peer, which determines the peer's reachability in the overlay, is also taken from the empirical distribution of NAT types. Finally, session's inter-arrival time and session length are all instantiated from the corresponding exponential distributions.

A peer trying to connect to other peers to get all necessary substreams is called an *orphan peer*. It sends search messages to discover other peers, sends join messages to connect to available peers, and finally constructs the full stream from them. A peer who is able and willing to offer a part of or all requested substreams for an *orphan peer* is called an *adoptive peer*. An *adoptive peer* positively answers to the search message of an orphan. Once having multiple positive answers from potential *adoptive peers*, an orphan has to choose to which peer it should send a join message. We evaluate the following four peer selection algorithms all relying on available information:

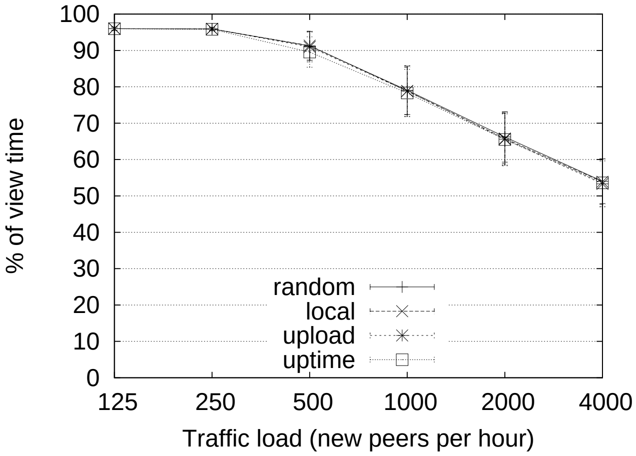


Fig. 11. Average view time vs. traffic load.

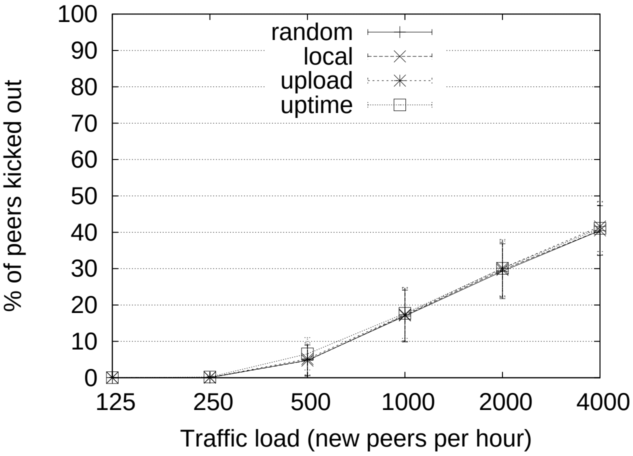


Fig. 12. Average percentage of peers kicked out vs. traffic load.

- **Random**: an orphan peer tries to connect to a randomly selected adoptive peer.
- **Local**: an orphan peer tries to connect to its closest adoptive peer (in terms of the hop distance).
- **Upload**: an orphan peer tries to connect to the adoptive peer proposing the highest upload amount (measured in terms of the number of substreams).
- **Uptime**: an orphan peer tries to connect to the adoptive peer which joined the overlay the earliest as it has the highest probability of remaining in the overlay.

Figure 11 shows the average peer view time ratio as a function of the traffic load. We can see that the effects of the various algorithms on the view time do not make much difference compared to random peer selection. Although there is 7% difference between the worst (uptime) and the best (random) algorithms at a traffic load of 2,000, and 5% difference at a load of 4,000, the differences are not significant. This somewhat unexpected result implies the relative importance of the *user level characteristics* such as upload capacity and session length, over the *system level configurations* such as the peer selection algorithms.

Figure 12 shows the average percentage of peers kicked out as a function of the traffic load. Compared to the previous results, the decrease in view time ratio, as illustrated in Figure 11, is mainly due to peers being kicked out of the network. Only a small percentage is caused by the interruptions resulting from peer disconnections and reconnections.

Figure 13 shows the average number of interruptions per peer as a function of the traffic load. We observe that the average number of interruptions per peer gradually increases when the number of peers increases until reaching traffic load 2,000. When the number of new peers increases, the overlay grows and the average churn rate becomes higher, and thus leading to more frequent connections and reconnections. However, when the overlay is getting saturated by new peers, those new peers cannot manage to join the overlay and are kicked out. The number of connections and reconnections does not grow as much in this case because those kicked out peers do not significantly contribute to this number. However, the total number of peers still increases, and thus the ratio does not increase anymore.

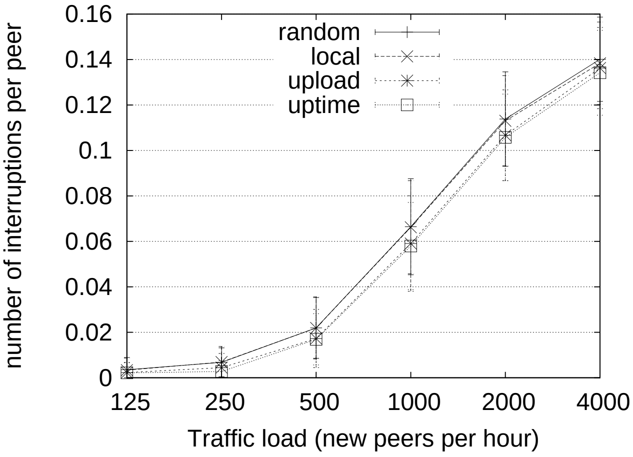


Fig. 13. Average number of interruptions per peer vs. traffic load.

Two remarks can explain the marginal improvements made by the alternative algorithms compared to the simple random selection algorithm. First, the sessions are typically short-lived. Roughly 50% of the sessions are shorter than 1.5 minutes. This creates a lot of churns that render the evolution of the overlay hard to control over time. Second, the redistribution factor distribution is heavily lopsided towards small values; 50% of the peers have an upload capacity lower than 50%. Thus, peers with long sessions may have a low upload capacity and not be so useful. It turns out that these two factors impact the view time ratio much more than the various selection algorithms. When traffic load is high, the algorithm that performs better than random selection is the local algorithm.

D. *Influence of overlay parameters on simulation results*

We now study the influence of three important parameters used in the overlay which are the number of substreams per stream, the buffer capacity and the number of search attempts.

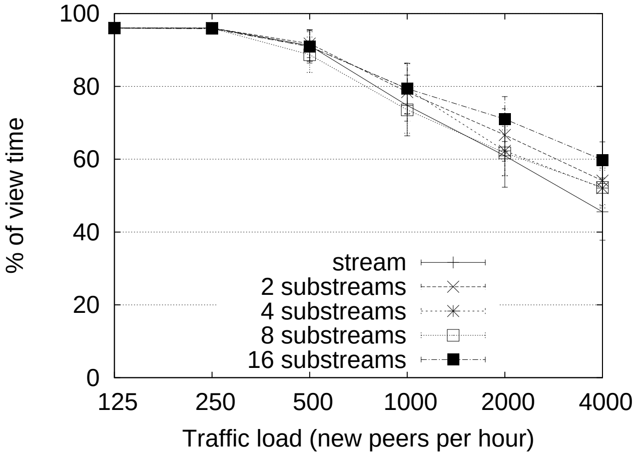


Fig. 14. Impact of the number of substreams on the peers' view time.

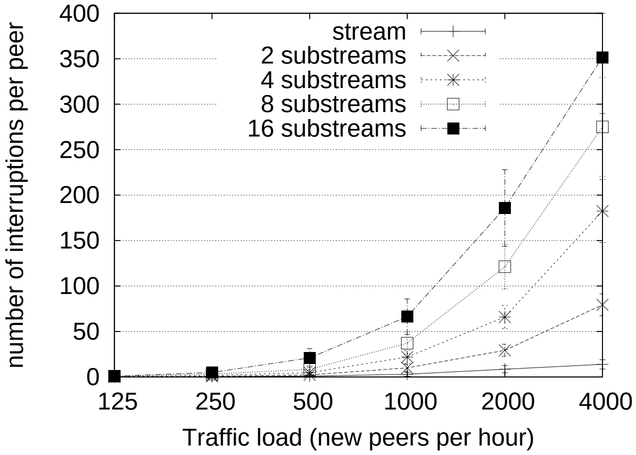


Fig. 15. Impact of the number of substreams on the interruptions.

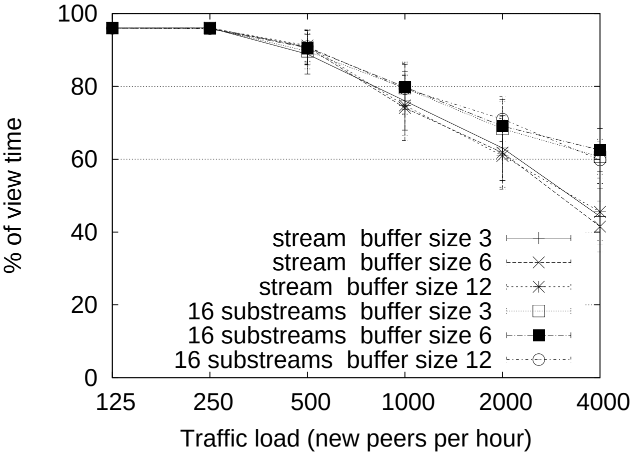


Fig. 16. Impact of the buffer size on the peers' view time.

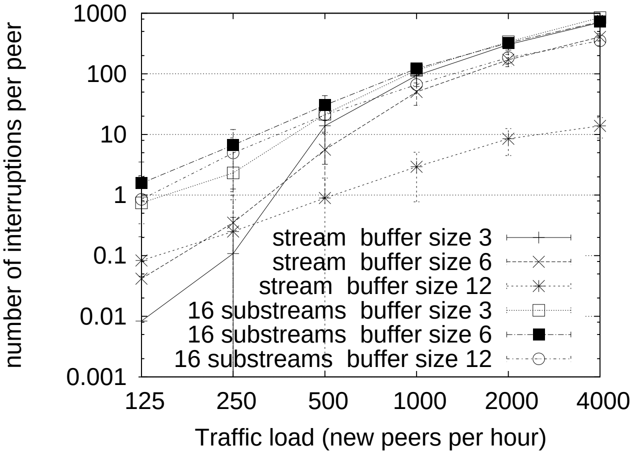


Fig. 17. Impact of the buffer size on the interruptions.

1) Influence of the number of substreams: In P2PTV systems such as Zattoo, a full stream is subdivided into multiple substreams. Here we study the impact of the number of substreams that constitute a full stream. In this experiment, we change the peer buffer size so that it can store 12 seconds of streaming data. we keep the other parameters the same as previously, such as realistic inter-arrival time, random peer selection algorithm, and two search attempts. Figure 14 shows that the view time ratio is increasing when the main stream is divided into multiple substreams. The redistribution is easier as peers can allocate their uplink bandwidth on a finer granularity. As a result, substreams improve system scalability.

Figure 15 shows that the number of interruptions increases with the number of substreams, due to the difficulty in finding multiple different substreams that are roughly in sync. In fact, the substreams increase the control traffic overhead. However, this is not an issue while the view time ratio remains high.

2) Influence of the buffer capacity: We now use the same number of search attempts as above with the random peer selection algorithm. To observe the influence of the buffer capacity, we increase it from 3 to 12 seconds. We show in Figure 16 the impact of the buffer capacity on view time ratio and the number of interruptions per peer. Under the heaviest load of 4,000 new peers per hour, the view time ratio is roughly 60% for 16 substreams and 45% for 1 full stream. This shows that splitting the stream does significantly improve the viewing time. The buffer time however, has no real impact on the viewing time. When observing the number of interruptions shown in Figure 17, we see that it grows with the traffic load and is higher when the stream is divided into 16 substreams. When the stream is not divided, a longer buffer duration reduces the number of interruptions. When the traffic load increases, the number of interruptions for a small or medium buffer tend towards the same values whether the stream is split or not. From the previous simulation results, it appears that the division of the full stream into several substreams increases the number of interruptions (especially under low traffic load) but improves the amount of viewing time. Moreover, a large buffer capacity can reduce the number of interruptions but does not affect the view time ratio. Thus, the best solution is to divide the full stream in many substreams and use a buffer with a large storage capacity (i.e., 12 seconds in our simulations).

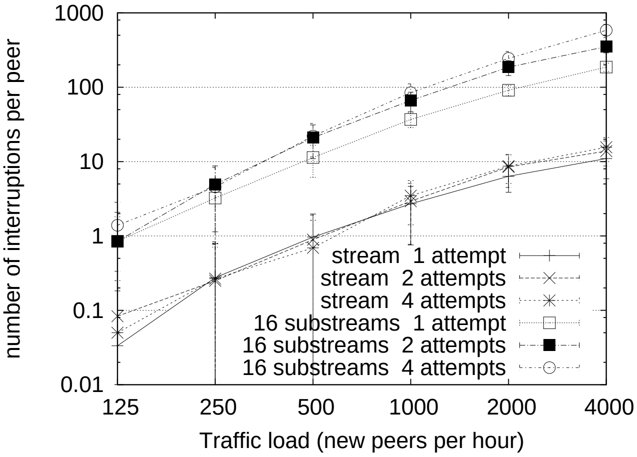


Fig. 18. Impact of the number of search attempts on the interruptions.

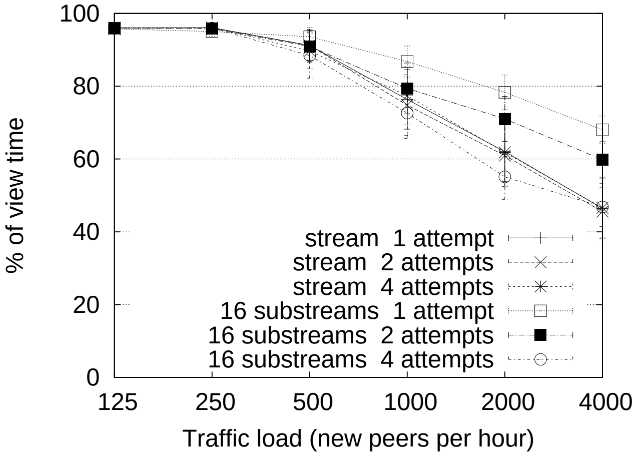


Fig. 19. Impact of the number of search attempts on the peers' view time.

3) Influence of the number of search attempts: We now study the impact of the number of search attempts. We set a buffer capacity of 12 seconds and use the random peer selection algorithm.Regardless of the number of search attempts, Figure 18 shows that the number of interruptions does not change and remains low. A high number of attempts with several substreams improve the overlay scalability. Figure 19 shows the benefit of using multiple substreams. With substreams the view time ratio reaches 70%. When a peer attempts two rounds of searches, the view time ratio curve reaches 60%, and finally with four rounds of search attempts the view time ratio falls below 50%. Figure 20 shows the network saturation point. When this point is reached, new peers start to get kicked out. This number is minimal when the stream is divided into multiple substreams, and the number of search attempts is equal to 1. If peers increase the search attempts at the saturation point, the saturation phenomenon is only amplified. Dividing the stream is necessary to ease the redistribution while the number of search attempts helps peers find adoptive peers. Without using substreams, the overlay becomes less resilient and increasing the number of search attempts does not help find available adoptive peers.

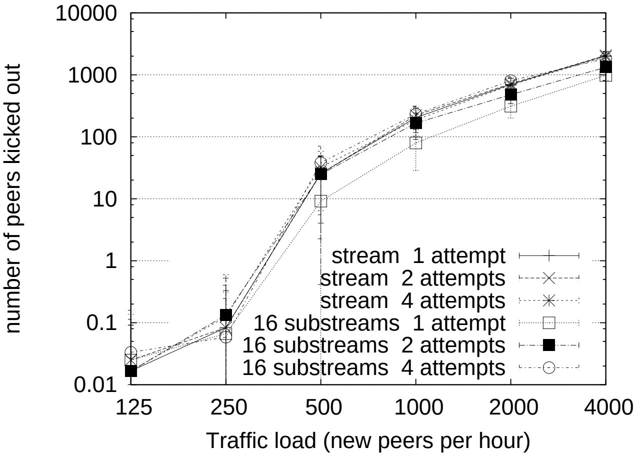


Fig. 20. Impact of the number of search attempts on the kicked out peers.

V. Conclusion

P2PTV network is a content delivery architecture that is particularly attractive due to the relative ease of deployment and its potential bandwidth savings. However, as it gains in popularity, we need to evaluate the behavior of large scale P2PTV network under heavily loaded conditions. Such heavy-load scenarios have not been fully investigated in the literature due to the lack of measurement data. In this paper, we studied several factors that can affect the scalability and efficiency of typical P2PTV network by performing detailed simulations. In order to instantiate realistic simulation settings, we analyzed 9.8 million sessions collected from the professional-grade Zattoo's P2PTV network. In this paper, we demonstrated that both the redistribution factor and peers' available upload capacity have a strong impact on the P2PTV network's scalability. Contrary to the intuition, we found that the peer selection process plays a relatively marginal role in improving the P2PTV network scalability. Instead, we observed that the redistribution factor and session lengths have far more significant effect on the maximum capacity of the P2P overlay. Finally, we demonstrated the influence of the buffer size and peer search parameters on the overall efficiency of the overlay. Building a scalable and efficient P2P overlay thus requires a careful consideration of the parameters we have studied in this paper, including peer characteristics as well as overlay configuration parameters. Our contribution is therefore multifold and can be summarized as follows:

- We have analyzed 9 million P2PTV sessions from a real production network and we have derived the probability distributions of the most important parameters such as session length, redistribution factor, inter-arrival time, NAT type. These distributions can be useful to other researchers for realistically simulating P2PTV networks.
- We have simulated a P2PTV network and have shown that the redistribution factor is the parameter that has the highest impact on the viewing time and the amount of kicked out peers.
- We have also shown that peer selection strategies have nearly no influence upon these values.

- We have finally shown that configuration parameters such as the number of substreams, the buffer duration and the number of search attempts do have an impact but only when the traffic load is high.

Our future work will be aimed at studying new methods to limit peer churn and relieve the limitations imposed by low upload capacity peers. For example, the positioning of long lived peers at the top of the distribution tree could help to reduce churn. Moreover, the filtering of low upload capacity clients could also help to achieve a redistribution factor close to one thus making the overlay scalable. We plan to carry out further experiments to study these important issues.

References

[1] L. Bracciale, F. L. Piccolo, D. Luzzi, S. Salsano, G. Bianchi, and N. Blefari-Melazzi, “A Push-based Scheduling Algorithm for Large Scale P2P Live Streaming,” in *Proceedings of the 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*, 2008, pp. 1–7.

[2] S. Xie, B. Li, G. Y. Keung, and X. Zhang, “CoolStreaming: Design, Theory, and Practice,” *IEEE Transactions on Multimedia*, vol. 9, no. 8, December 2007.

[3] H. Chang, S. Jamin, and W. Wang, “Live Streaming with Receiver-based Peer-division Multiplexing,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 1, February 2011, a shorter version of this paper appeared in Proceedings of the ACM Internet Measurement Conference, Oct. 2009.

[4] M. Zhang, Q. Zhang, and S.-Q. Yang, “Understanding the Power of Pull-based Streaming Protocol: Can We Do Better?” *IEEE JSAC*, vol. 25, no. 9, pp. 1678–1694, 2007.

[5] K. Shami, D. Magoni, H. Chang, W. Wang, and S. Jamin, “Impacts of Peer Characteristics on P2PTV Networks Scalability,” in *Proceedings of the 28th IEEE Conference on Computer Communications – Mini-Conference*, April 2009.

[6] C. Cassagnes, D. Magoni, H. Chang, W. Wang, and S. Jamin, “On the scalability of p2p-based push-driven live streaming systems,” in *Proceedings of the IEEE International Conference on Communications*, May 2010.

[7] D. Tran, K. Hua, and T. Do, “ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming,” in *Proceedings of the IEEE Infocom*, 2003.

[8] R. Rejaie and S. Stafford, “A Framework for Architecting Peer-to-Peer Receiver-Driven Overlays,” in *Proceedings of the ACM NOSSDAV*, 2004, pp. 42–47.

[9] X. Liao, H. Jin, Y. Liu, L. Ni, and D. Deng, “Anysee: Peer-to-Peer Live Streaming,” in *Proceedings of the IEEE Infocom*, 2006.

[10] F. Pianese, “PULSE: A Flexible P2P Live Streaming System,” in *Proceedings of the 9th IEEE Global Internet Symposium*, 2006.

[11] F. Pianese, D. Perino, J. Keller, and E. Biersack, “PULSE: An Adaptive, Incentive-Based, Unstructured P2P Live Streaming System,” *IEEE Transactions on Multimedia*, vol. 9, no. 6, 2007.

[12] K. Sripanidkulchai, B. Maggs, and H. Zhang, “An Analysis of Live Streaming Workloads on the Internet,” in *Proceedings of the ACM Internet Measurement Conference*, 2004, pp. 41–54.

[13] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, “On Next-Generation Telco-Managed P2P TV Architectures,” in *Proceedings of the 7th International Workshop on Peer-to-Peer Systems*, 2008.

[14] S. Xie, G. Keung, and B. Li, “A Measurement of a Large-Scale Peer-to-Peer Live Video Streaming System,” in *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW’07)*, 2007, p. 57.

[15] X. Zhang, J. Liu, B. Li, and T.-S. Yum, “CoolStreaming/DONet: A Data-driven Overlay Network for Live Media Streaming,” in *Proceedings of the 24th IEEE Infocom*, 2005, pp. 2102–2111.

[16] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, “A Measurement Study of a Large-Scale P2P IPTV System,” *IEEE Transactions on Multimedia*, vol. 9, no. 8, 2007.

[17] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, “Measurement and Modeling of a Large-scale Overlay for Multimedia Streaming,” in *Proceedings of the Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2007.

[18] S. Ali, A. Mathur, and H. Zhang, “Measurement of Commercial Peer-to-Peer Live Video Streaming,” in *Proceedings of the ICST Workshop on Recent Advances in Peer-to-Peer Streaming*, 2006.

[19] S. Thomas, F. Olivier, B. Alessio, D. Alberto, P. Antonio, V. Giorgio, and S. Kavé, “Traffic Analysis of Peer-to-Peer IPTV Communities,” *Computer Network*, vol. 53, no. 4, pp. 470–484, 2009.

[20] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H.Zhang, “The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-points,” in *Proceedings of the ACM SIGCOMM*, 2004.

[21] J. Douceur, J. Lorch, and T. Moscibroda, “Maximizing Total Upload in Latency-Sensitive P2P Applications,” in *Proceedings of the 19th ACM SPAA*, 2007, pp. 270–279.

[22] F. Pianese and D. Perino, “Resource and Locality Awareness in an Incentive-based P2P Live Streaming System,” in *Proceedings of the ACM P2P-TV*, 2007.

[23] R. Kumar, Y. Liu, and K. Ross, “Stochastic Fluid Theory for P2P Streaming Systems,” in *Proceedings of the IEEE Infocom*, 2007.

[24] M. Feldman, K. Lai, I. Stoica, and J. Chuang, “Robust Incentive Techniques for Peer-to-Peer Networks,” in *Proceedings of the 5th ACM Conference on E-Commerce*, 2004, pp. 102–111.

[25] Y. Chu and H. Zhang, “Considering Altruism in Peer-to-Peer Internet Streaming Broadcast,” in *Proceedings of the ACM NOSSDAV*, 2004.

[26] Y.-W. Sung, M. Bishop, and S. Rao, “Enabling Contribution Awareness in an Overlay Broadcasting System,” in *Proceedings of the ACM SIGCOMM*, 2006.

[27] Z. Liu, Y. Shen, K. Ross, S. Panwar, and Y. Wang, “Substream Trading: Towards an Open P2P Live Streaming System,” in *Proceedings of the IEEE International Conference on Network Protocols*, 2008, pp. 94–103.

[28] C. Wu, B. Li, and S. Zhao, “Magellan: Charting Large-Scale Peer-to-Peer Live Streaming Topologies,” in *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS’07)*, 2007, p. 62.

[29] T. Small, B. Liang, and B. Li, “Scaling Laws and Tradeoffs in Peer-to-Peer Live Multimedia Streaming,” in *Proceedings of ACM Multimedia06*. ACM, 2006, pp. 539–548.

[30] S. Guha and P. Francis, “Characterization and Measurement of TCP Traversal through NATs and Firewalls,” in *Proceedings of the ACM Internet Measurement Conference*, 2005, pp. 199–211.

[31] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, “RFC 5389: Session Traversal Utilities for NAT (STUN),” October 2008.

[32] D. Magoni, “Network Topology Analysis and Internet Modelling with Nem,” *International Journal of Computers and Applications*, vol. 27, no. 4, pp. 252–259, 2005.

[33] D. Magoni and M. Hoerdt, “Internet Core Topology Mapping and Analysis,” *Computer Communications*, pp. 494–506, 2005.

Cyril Cassagnes received a M.S. degree in computer science from the University of Bordeaux in 2008. He is currently a Ph.D. student at the University of Bordeaux working under the supervision of Prof. Magoni. His research interests include P2P overlay networks as well as distributed addressing and greedy routing algorithms.

Damien Magoni received a Ph.D. degree in computer science from the University of Strasbourg in 2002. Since 2008, he is a full professor at the University of Bordeaux and a member of the LaBRI-CNRS joint laboratory. He is an IEEE senior member since 2007. He was a visiting scientist at the AIST, Tsukuba, Japan in 2005 and at the University of Sydney in 2007. His research interests focus on the Internet architecture and protocols.

Hyunseok Chang received the B.S. degree in electrical engineering from Seoul National University in 1998, and the M.S.E. and Ph.D. degrees in computer science and engineering from the University of Michigan in 2001 and 2006, respectively. He is currently a member of the Network Protocols and Systems Department, at Alcatel-Lucent Bell Labs, Holmdel, NJ. Prior to that, he was a Software Architect at Zattoo, Inc., Ann Arbor, MI. His research interests include content distribution network, mobile applications, and network measurements.

Wenjie Wang received his B.S. degree from Peking University and his Ph.D. degree in computer science and engineering from the University of Michigan in 2006. He co-founded a P2P live streaming company named Zattoo in 2005 where he served as CTO. He worked for IBM Research China in 2009. He is now with PPLive Inc., serving as P2P Chief Architect. His research interests include large scale self-organizing network, data collection and modeling, P2P video streaming and end-host multicast.

Sugih Jamin Sugih Jamin is an Associate Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. He received his Ph.D. in computer science from the University of Southern California, Los Angeles in 1996. He received the National Science Foundation (NSF) CAREER Award in 1998, the Presidential Early Career Award for Scientists and Engineers (PECASE) in 1999, and the Alfred P. Sloan Research Fellowship in 2001. He co-founded a peer-to-peer live streaming company, Zattoo, in 2005.