Universal Journal of Communications and Network 1(4): 142-150, 2013
DOI: 10.13189/ujcn.2013.010403 http://www.hrpub.org

On the impact of the initial peer list in P2P live streaming applications: the case of Sopcast

Imane Mnie-Filali[1], Soufiane Rouibia[1,*], Dino Lopez-Pacheco[2], Guillaume Urvoy-Keller[2]

[1]TMG, 44230 Saint Sébastien sur Loire, France

[2]Laboratoire I3S UNS/CNRS UMR 7271, France

[*]Corresponding Author: rouibia@tmg.eu

Abstract The acronym P2PTV, which stands for Peer-to-Peer Television, refers to P2P applications that enable multimedia contents broadcasting through the Internet. These applications have become very popular due to their capability of delivering material with or without copyright protection to a large audience. In P2PTV applications, the start-up phase – from the moment the client joins an existing channel to the moment she can start viewing the content – is crucial. A long time before showing the first images of a video, can make a user switch to another channel or to an alternative P2P application. Later on, the same user behavior is expected in case of frozen images. In this paper, we consider the case of Sopcast, which is currently one of the most popular P2PTV applications. Sopcast recently modified its start-up algorithms. We uncover the details of the new strategy adopted by Sopcast and evaluate its performance as a function of the swarm size and the geographic source of the content.

Keywords P2PTV applications, SOPCAST

1 Introduction

The key idea behind Peer-to-Peer (P2P) is scalability. Unlike client/server architectures, the performance of P2P architectures increases as the number of users increases, since any peer (a host engaged in a P2P network) can act like both client and/or server.

In P2PTV, the data is a live stream video of a television program which is usually divided into pieces (video chunks) and transported through the IP network infrastructure. Several P2PTV freeware applications are available on Internet, such as Sopcast [1], TvAnts[2], PPlive[3], PPStream[4], etc.

Recent measurement studies, e.g. [5] have indicated that Sopcast is by far the most popular P2PTV application in Europe. We observed a similar result on several measurement campaigns that we made during several sportive events over the recent years. For instance, during the Champions League final match, FC Barcelona vs. Manchester United, in May 2011, Sopcast was used by around 66% of people using a P2PTV software to follow the match – see Figure 1.

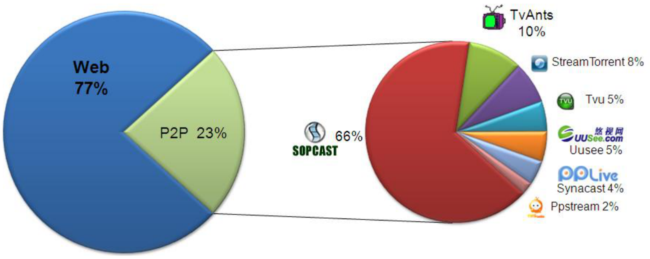


Figure 1. Media distribution mechanism breakdown

In early 2011, Sopcast modified its start-up phase algorithm. The main objective was to provide to a new arriving peer a list of peers that are presumably its nearest neighbors. This strategy is a major shift in the protocol, which was previously relying on a random approach to pick the addresses in the list maintained by the central tracker

[6] whose role is to maintain a list of active users and help new arriving peers to discover neighbors. The strategy – hereafter denoted as the peer-picking algorithm – used by Sopcast can be interpreted as a Peer-for-Peer (P4P) approach, that aims at keeping the path between two peers as short as possible [7].

Given the key importance of the start-up phase in P2PTV, we analyze, in this work, the behavior of Sopcast during the start-up phase and the peer list (PL). We rely on an experimental approach to uncover the impact of the change in the initial peer picking algorithm of Sopcast. We use several peers at various locations around the world and we connect to streams of various size to assess its efficiency. Our contributions are as follows:

- We uncover the new strategy that is applied by Sopcast to select the initial neighborhood of a new arriving peer. This is a mandatory step in the analysis since the algorithms of Sopcast are not publicly available.
- We demonstrate that the efficiency of the P2P algorithm depends on the size of the considered population (i.e., the popularity of a channel). We also demonstrate that another key parameter to take into account is the geographical relation between the content and the new peer.

The remainder of this paper is organized as follows: Section 2 presents the Sopcast architecture and analyzes key parts of its protocol. In Section 3, we detail the peer picking algorithm and the experimental set-up. Results are presented in Section 4. We review related work in Section 5. Discussion and concluding remarks are presented in Section 6.

2 Sopcast

Sopcast relies exclusively on the UDP transport protocol. It offers a variety of channels that can be grouped by language, ID, region, etc. Sopcast allows users to create their own broadcast channel if they are signed up. For each channel, Sopcast provides a quality indicator, shown as a blue bar next to the channel ID. By clicking on this bar, an information frame appears that displays a channel popularity indicator between zero and one[1], the encoding format (WMV, WMA, RMVB. . .), the broadcast starting time and the bit rate, typically ranging from ∼380 Kbps to ∼ 1168Kbps.

Sopcast is a closed-source application protocol. In addition, data exchanged between hosts is encrypted. We rely on an experimental approach to uncover its performance. As stated in Section 1, the start-up time of a TV channel plays a key role in the Quality of Experience (QoE) of users. Hence, we began by analyzing the network level latency between a peer and its neighbors, as this parameter directly impacts the time needed to begin watching a given channel. Also, the latency directly impacts the lag between the original content source point of time and the one of the final users. A typical value for the maximum delay between any peer, with respect to the initial source of the stream is one minute. This means that no peer should be more than one minute late as compared to the original source of content [8]. It is then clear that the network latency between peers is a key metric to assess the Quality of Experience (QoE) of a user.

2.1 Architecture

In Sopcast, each channel corresponds to an independent stream, and a client joins a stream by contacting first a central server (called the tracker hereafter), which is similar to the BitTorrent tracker [9]. The tracker, that keeps track of the peers in each channel, provides to the new client a list of up to 32 IP addresses (the PL) to contact. Thus, the tracker plays a main role in the start-up phase. In case of failure of the tracker, it would be impossible for a new peer to join an existing channel.

In Sopcast, the peers downloading a content directly from the original broadcaster are known as super-peers [10]. Sopcast limits the number of peers that can connect directly to the initial broadcaster to 11 peers [11, 12]. Any peer can become a super-peer.

The data exchanged between peers is encrypted. Additionally, the algorithms that control both the chunk and peer selection are not publicly available. Hence, we are not able to confirm if Sopcast uses also a tit-for-tat and/or rarest-first algorithm[13] like BitTorrent

2.2 Protocol analysis

In order to understand the behavior of Sopcast peers, we have analyzed the traffic originated and at destination of our clients that we deployed in different sites. The results reported below, concerning the protocol profiling, confirm what others studies have also shown [6].

When a peer connects to a channel, different packets are exchanged, see Table 1. To contact new potential neighbors and also to contact the tracker, peers first send an initiation/HELLO packet. The content of this packet is always the same.When a client sends a HELLO packet to the tracker to join a channel, the latter replies with an 80-bytes packet followed by a 60-bytes packet, that we denote as peer list packets (PL packets). The PL packets contains a selection of peers already watching the channel and the current offset within the channel, i.e., the elapsed time between the broadcast start time and the time at which the client sends the request to the tracker.

[1] The popularity indicator is a function of the number of peers connected to the channel, though the exact mapping between the numbers and the indicator is unknown.

Table 1. Sopcast packets

Packets length (Bytes)	Description
94	Initiation / HELLO Packets
80 to 166	Response to HELLO Packets
70	Acknowledgment
84	Keep alive packets
62-76-82-88	Control packets
1000 to 1320	Data – video content

After the initial handshake with a peer, we observe a number of packets whose payload size ranges from 60 to 166 bytes, that we denote as control packets. The control packets contain a set of requests, which are sent in order to obtain a chunk from a given channel. Indeed, after exchanging those control packets, we observe the transmission or reception of data packets, whose size varies between 1000 and 1320 bytes, and that contain the video chunks. Each received data packet is acknowledged with a 70-bytes packet. Finally, our traces reveal a periodical exchange of 84-bytes packets between the client and the tracker, which seem to be used to indicate that the current peer is still active in the network. We refer to these packets as keep-alive packets. Note that the content of acknowledgments and keep-alive packets is always the same. Figure 2 depicts the messages exchanged between two Sopcast peers.

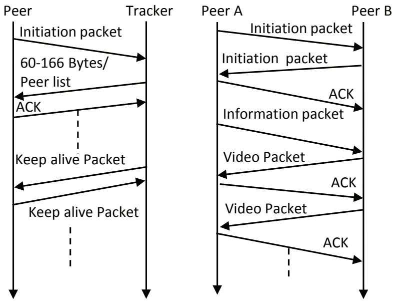


Figure 2. Sopcast Communication

In order to understand the tracker's algorithm to build the initial PL, we analyzed such a list from our Sopcast client running on a Linux machine. Indeed, when executing Sopcast by the command line, the content of the PL is displayed in the terminal.

3 Experimental set-up and Peer Selection Algorithm

Our measurement infrastructure consists of three hosts located in different networks. The first machine is a personal computer equipped with an Intel Pentium 2.70 GHz processor, 2 Go RAM running a Windows XP system. The two other computers are servers equipped with 3.8 GHz CPUs. In each host, the Sopcast client runs inside a virtual machine.

The PL received by our clients consists of 32 IP addresses, except in the cases where we connected to a channel with a small number of peers (according to the channel ratio), where it was smaller. We also observed that once a newly arriving client has contacted the peers of its initial PL, it contacts additional peers obtained through a peer exchange algorithm similar to the one used in BitTorrent [9]. This information (address obtained from the tracker or from a neighbor) is clearly indicated on the command line.

Before the modifications introduced to Sopcast in 2011, peers were selected based on their uploading rate, as observed in previous studies [14, 12]. Our experiments demonstrate that Sopcast now uses the distance between IP addresses in order to build the initial PL. Though the exact formula for this distance is unknown, it is giving more weight to most significant bits in the IP address. When calculating distance, we use the simplest form of such a metric between two IP addresses IP_1 and IP_2, where $IP_j(i), j \in \{1,2\}$ is the i-th bit in the IP addresses ($i = 1$ is the most significant bit):

$$d(IP_1, IP_2) = \sum_{i=1}^{32} 2^{32-i} XOR(IP_1(i), IP_2(i))$$

We refer to this distance as the IP distance in the remainder of this article. Our main objective is to investigate the relevance of the IP distance to build the PL. In particular, we analyze the relationship between the IP distance of two peers and the latency that they experience. The latter is measured using the ping protocol. We also use an *application ping* that we developed. Our application ping mimics the handshake procedure of Sopcast by sending a HELLO packet of 94 bytes, to elicit a 70-bytes length ACK [2] from the remote peer (see Table 1).

[2] Acknowledge receipt of a packet.

Table 2. Peers replying to ICMP requests

Channel	Host 1 (class./appl.)	Host 2 (class./appl.)	Host 3 (class./appl.)
Pro TV	10/5	14/32	14/12
Digisport	7/6	14/24	17/21
Digifilm	9/13	20/26	17/14
Kanal ID	11/12	12/24	18/22
Acasa TV	14/12	12/24	12/22
Realitatea	14/19	11/26	18/6

4 Results

4.1 Firewall configurationp

Before delving into the details of the peer picking algorithm used by the Sopcast tracker, we investigate if the tracker follows a strategy to control the number of peers that are behind a firewall to compose an initial PL. To estimate if a given peer is behind a firewall, we execute both a classical ping and our application ping to that node. If such a node answers to the legacy ping, then the probability of being behind a firewall is low. However, if the node does not reply to our legacy ping, but successfully replies to our application ping, then, the probability of that node being behind a firewall is high.

Table 2 reports the number of peers answering to classical and application pings, out of the 32 peers in the initial peers list. The channels in Table 2 feature varying popularity.

Our ping results show that the number of firewalled peers composing the initial PL did not follow a strict criteria. Thus, the tracker does not take into account the presence or not of a firewall in order to build such a list.

4.2 Peer Picking Algorithm

Sopcast does not offer any information about the algorithm used by the tracker to pick addresses and create a PL for a newly joining peer. Thus, the starting point of this study was the observation of the similarity between the IP address of the new peer and the ones composing the initial PL.

To verify if the IP addresses from the PL were always numerically close to the one of the new peer, we deployed three sopcast client on very different IP networks. Indeed, the most significant byte of the clients' network addresses was always different. Later on, we engaged the three hosts simultaneously on the same channel and we collected the IP addresses obtained from the tracker.

Finally, we computed the numerical distance (or Hamming distance) between each client IP address and the IP addresses from its associated PL, as well as the distance between each client IP address and the IP addresses from the PL of our remaining two clients. Our results, graphically available in Figures 3 and 4, show that the IP address similarity is the main criterion used by the tracker when building the PL. Indeed, the IP distance between a client and its associated PL is smaller than the IP distance between a client and the PL associated to other clients located in different networks. Note that in Figures 3 and 4, we use two different y-axes. The right y-axis corresponds to the distances between one peer and its own PL while the left one is used to show the distances with PL from our other clients.

Moreover, we want to highlight that number of peers involved in the channel play an important role on the estimated distance between a client and its PL. For instance, if the channel aggregate several peers whose addresses are close to the client, we will obtain a results similar to the Figure 3, where the function of the distance between the client and its peers increase smoothly. Otherwise, if only a few peers close to the client are connected to the channel (less than 32) and the other peers are located on very different networks (which can be the case if they are geographically far) then the function can experience a higher slope at a given point of the figure, similar to results shown in Figure 4. In the latter case, the impact of the tracker algorithm would not be easily noticeable.

The experiments already describes in this subsection were executed several times and we obtained similar results. Trough our multiple experiments, we also noted that when a client joins several times the same channel (BeIn Sport) over short periods of time, the initial PL remains unchanged, which reinforce our hypothesis that the IP similarity is the main criteria used by the tracker to create a PL.

4.3 The Sopcast network dynamic

After receiving the initial PL, a peer will contact each member of that list in order to obtain the multimedia content.Peers will use the Peer Exchange protocol (PeX) [15] to receive the neighbors PL like in BiTtorrent.

The objective of this mechanism is twofold. On the one hand, active peers of a channel will have an updated view of the network architecture and, on the other hand, the collaboration of the newest peers in the network can potentially improve the performance of the TV broadcasting system.

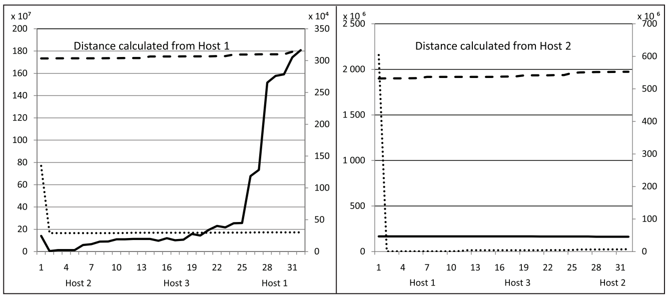


Figure 3. Peers IP distances

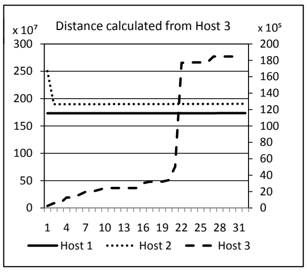


Figure 4. Peers IP distances

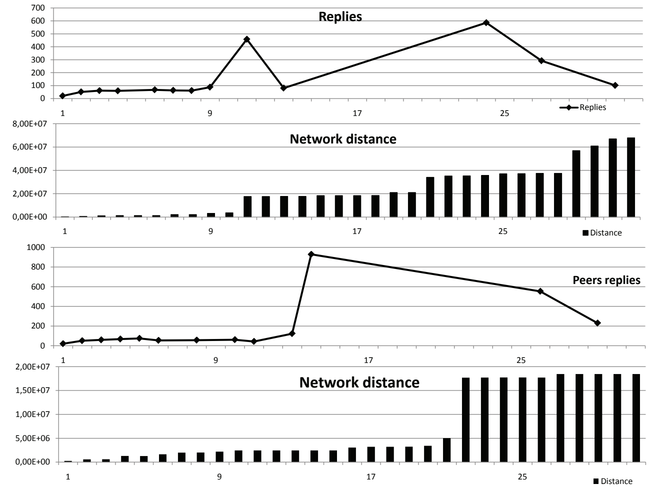


Figure 5. Correlation between the round-trip time and the peers reachability (PL from Germany vs. Greece game)

4.4 IP distance vs network latency

We next investigate the relationship between the numerical IP distance (referenced as network distance on figures) and the network latency. We estimated the round-trip time (RTT), averaged over a few tens of probes, between our sopcast clients and their associated PL collected during the experiment described above. Our results show that, for popular channels (channels with a popularity higher than 0.25 according to our sopcast clients), the experienced RTT between the client and the first ten peers of the PL (the closest peers) tends to be smaller than the experienced RTT between the client and the remaining peers. Figure 5, which shows the average RTT and the Hamming distance between host 3 and its peers from the initial PL, illustrates very well our observations.

However, if the popularity of the channel is low (channels with a popularity lower than 0.1 according to the popularity level given by our sopcast clients) and if the tracker returns less than 32 peers, the experienced RTT and Hamming distance seems to follow a random pattern. Such a behavior can be graphically seen in Figure 6(a), that presents the average RTT between host 3 and its initial PL, which has been retrieved during the half-time break of the same football match France vs Spain, already mentioned in Section 1. Indeed, frequently, channels used to follow a popular football soccer match have a high popularity during the active game periods, but their popularity drastically decreases during the half-time break period.

4.5 Peers Geographical Location

So far we have examined two different scenarios: in the first scenario the popularity of the channel was high and our client was located in the same geographical region as the bulk of the peers in the channel, and in the second scenario, the channel popularity was low and our client was still located geographically close to the peers

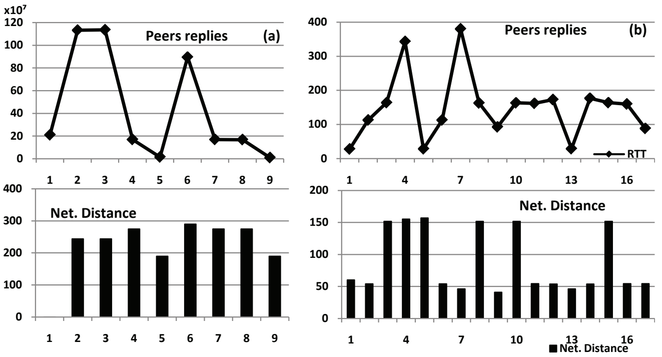


Figure 6. Peer reachability vs. Network distance

connected to that channel. In the first scenario we could observe a correlation between the Hamming distance and the experienced RTT (the RTT for the first tens peers is usually smaller than the one of the remaining 22 peers), while in the second scenario we were unable to find any correlation between the RTT and the IP distance of the peers.

Moreover, in this section we aim at analyzing a third scenario: the popularity of the channel is high but the client is geographically far from the rest of the peers connected to the channel. Thus our client still receives a PL containing 32 different addresses, which have the smallest Hamming distance with the newly joining peer. In order to analyze this case, we deployed a sopcast client in a network located in Asia that requested a connexion to an European channel (DiGi Sport 2, which is a Romanian channel) broadcasting a football soccer game of the Romanian League. The number of peers connected to the channel was close to 14,500, which produced a popularity of around 0.52 according to Sopcast.

Our experiments show that when the peer requesting a connexion is geographically far from the sources, the Hamming distance policy does not have a clear impact on the QoS/QoE. Indeed, we were unable to find once again any correlation between the experienced RTT and the Hamming distance between the client and its initial peers.

As a conclusion we can say that the sopcast peer-picking algorithm is pertinent when both the number of peers is high and the newly arriving peer is geographically close to the other peers already connected to the channel. However, the geographical location of the peers should be the main criteria to create the initial PL for that clients that are far from the bulk of peers in the channel.

We agree that taking into account the geographical location to build a PL would introduce more processing in the tracker. However, computing the average Hamming distance between the new client and its retrieved PL, and comparing this average to a given threshold, can quickly give an estimate if the new client is far from its possible peers. If so, the tracker should execute a geographical-aware algorithm to build an initial PL.

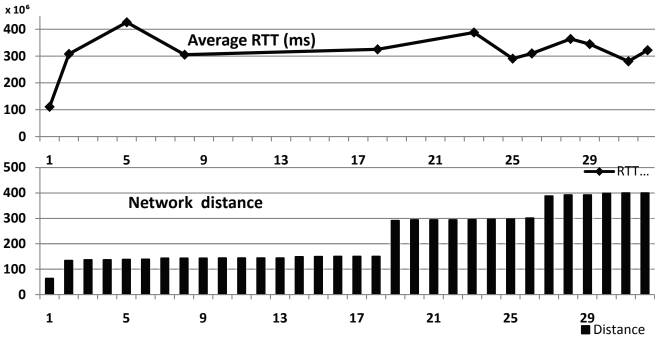
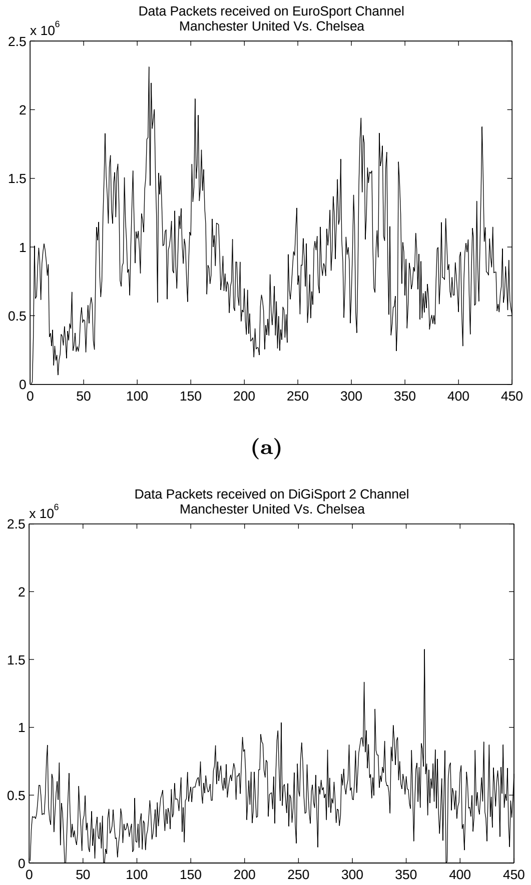


Figure 7. Peer reachability vs. Network distance

4.6 Impact of Peers Distance

We conducted another experiment to bring out the two aspects on section 4.3 and 4.4. We observe that the distance metric affect the PL selection and have a real impact on Sopcast's start-up, we investigate on this issue by measuring data packets received from the PL. The main idea is to confirm that peers within the PL distance value (the first IP distance value sent by the tracker and the last one on the PL) contribute on the proper start-up of the channel. We choose a premier league match broadcasted in October 2012, Manchester United Vs. Chelsea, turned in two different channel, EuroSport (in English) and DiGiSport 2 (in Romanian). The experiment lasted for roughly 8 minutes. We use Tcpdump to enable passive monitoring on the traffic transmitted at the Sopcast peers. On the first channel we received 77% of the total significant packets (more than 10) from peers within the PL distance value, whereas only 10.67% was received on the second channel. We present in Figure 8, data Packets received from those peers. We can notice that on figure 8-a the curve haven't reach a null value which means no



(a)

(b)

Figure 8. Received Data Packets

frozen images and the received packet on the 8 minutes was more considerable than figure 8-b and where we reach the null value many times.

The cultural aspect is very apparent in this case, on the first channel audience was about 7000 viewer while it count for 9742 for the second one. Normally we should received more from this latter but actually the diversity of peers - due to the cultural aspect- was conducive and allowed our IP address to be inserted in a suitable IP address set. The distance metric on the second channel places us on a IP address set where peers were from Romania and were too close from each others. In this case, even if peers send us their PL we were always on the same bulk and couldn't exchange with other sets.

5 Related Work

In [8], Sentinelli uncovered the key design parameters of Sopcast, PPLive, CoolStreaming and Anysee, based on early traffic measurements from those applications. The studied parameters included the overlay topology building, the buffer size and the playout delay. They propose to delineate control from data packets based on the size of the packet. A similar approach is used in all subsequent studies, including ours. For the case of Sopcast, they observed that the number of peers from which a peer download packets typically ranges from 2 to 5 peers, while the number of neighbors can be much higher.

Horvath et al. [14] analyzed PPlive, TVants and Sopcast and found that the last one (i.e. sopcast) does not favor peers with more bandwidth in the download process. Additionally, they observed that TVants favors close peers while building the overlay, unlike the two other protocols that seem to pick the neighbors at random. In the data transfer phase, both TVants and PPlive tend to favor data exchanges between close peers (same AS) unlike Sopcast.

Mendes et al. [12] report on measurements made in 2009 using a single end host connecting to an ADSL line with 12 Mb/s and 512 kb/s of downloading and uploading capacity respectively. They focused on three P2P streaming protocols: TVAnts, Sopcast and TVU Player2. In their experiment, their peer remained connected for several hours on a few channels. They argue that Sopcast does not obtain chunks from high bandwidth peers only as they upload more than 10GB with their modest 512 kb/s of upload capacity. However, Sopcast was featuring the highest performance in the sense that the download rate evolution described the lowest slope function, hence leading to less frozen images.Overall, they conclude that these three applications did not exploit any location information to reduce traffic load, contradicting the results provided by [14] in year 2009 (one year before).

In [10] Bermudez et al. focused on the performance of Sopcast only. Unlike previous studies, where a few P2PTV clients under the control of the experimenters were connected to some carefully chosen channels, they observed Sopcast using a passive approach with traces collected on the backbone of an European ISP. Sopcast was the most popular among all P2PTV applications they were able to detect in their traces, along with PPLive and TVants. They observed that peers with higher upload capacity exhibited a much higher peer discovery rate and concluded that the discovery rate depends on the peer upload capacity. They claimed that the peers discovery mechanism follows a random process in which the probability of contacting a peer is independent from other peers. They also show that there is apparently no preference based on the country the peers belong to in the exchange process.

Note that the above observations on the lack of locality of Sopcast do not contradict our results, as these listed studies were conducted before the change in the peer picking algorithm of Sopcast.

6 Conclusion

Previous studies on Sopcast outlined that this application is one of the dominant P2PTV applications in the Internet. Sopcast has recently modified the algorithm used to build the neighboring set of a new arriving peer. This algorithm is a crucial part of a P2PTV algorithm because it is strongly related to its startup time, i.e., the time needed to display the video stream of a channel.

We conducted a set of experiments in order to understand the features this new algorithm used by a peer to join a channel.

We first demonstrated that this new algorithm relies on the distance between the IP addresses of the newly arriving peers and the ones of the peers already present in the stream. The algorithm itself is executed at the tracker, with little attention paid to the NATed peers.

We outlined the degree of correlation between the IP distance and the actual network distance measured using legacy ping. This correlation depend on both the size of the stream (the number of connected peers) and the geographical relation between the new arriving peer and the content itself. If the new peer is far from the source of the content and also far from the peers connected to this stream, the start-up phase experienced by the new peer can be large as it takes time to this peer to find neighbors, through PeX algorithm.

REFERENCES

[1] www.sopcast.com, Std.

[2] www.ccants.com, Std.

[3] www.synacast.com, Std.

[4] www.pps.tv, Std.

[5] I. Bermudez, M. Mellia, and M. Meo, “Passive characterization of sopcast usage in residential isps,” in *Peer-to-Peer Computing*, 2011, pp. 1–9.

[6] B. I., M. M., and M. M., “Investigating overlay topologies and dynamics of p2p-tv systems: The case of sopcast,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1863–1871, 2011.

[7] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, “P4p: provider portal for applications,” in *ACM SIGCOMM '08*, 2008.

[8] S. A., M. G. andGerla M. andKleinrock L., and T. S., “Will iptv ride the peer-to-peer stream?” *IEEE Communications Magazine*, vol. 45, Issue: 6, pp. 86–92, June, 2007.

[9] X. Hei, Y. Liu, and K. Ross, “Inferring network-wide quality in p2p live streaming systems,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1640–1654, 2007.

[10] I. Bermudez, M. Mellia, and M. Meo, “Passive characterization of sopcast usage in residential isps,” in *Peer-to-Peer Computing (P2P),Conference on IEEE International*, 2011.

[11] C. Jiang, “Analyzing p2ptv traffic via measurement,” Master's thesis, TU Delft, 10th July 2009.

[12] J. Mendes, P. Salvador, and A. Nogueira, “P2p-tv service and user characterization,” *in Proceedings of IEEE International Conference on Computer and Information Technology (ICCIT)*, pp. 2612– 2620, June 29 2010-July 1 2010.

[13] A. Legout, G. Urvoy-Keller, and P. Michiardi, “Rarest first and choke algorithms are enough,” in *Internet Measurement Conference*, 2006, pp. 203–216.

[14] A. Horvath, M. Telek, D. Rossi, P. Veglia, D. Ciullo, M. Garcia, E. Leonardi, and M. Mellia, "Dissecting pplive, sopcast," TVAnts. Technical report, NAPA-WINE project, Tech. Rep., 2009.

[15] D. Wu, P. Dhungel, X. Hei, C. Zhang, and K. Ross, "Understanding peer exchange in bittorrent systems," in *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on.* IEEE, 2010, pp. 1–8.