Characterizing Dynamics in P2P Live Streaming System

Francisco Henrique Cerdeira Ferreira[1], **Ana Paula Couto da Silva**[1], **Alex Borges Vieira**[1]

[1]Computer Science Department – Institute of Exact Sciences
Federal University of Juiz de Fora (UFJF) – Juiz de Fora – MG – Brasil

francisco.henrique@ice.ufjf.br, {anapaula.silva, alex.borges}@ufjf.edu.br

Knowledge Area: Computer Networks
Year of admission: 2011

Abstract. *Despite large number of works devoted to understand P2P live streaming applications, most of them rely on characterizing the static view of these systems. These works do not worry about either understanding the systems dynamics or analizing how it evolves over time. Moreover, they lack on characterizing communities formation during a streaming session. In this work, we characterize the SopCast, one of the most important P2P live streaming application. We focus on its dynamics behavior and we also investigate the community formation phenomena. Using data collected from private channels, we characterize aspects from the P2P overlay, from the communities and from the data exchange pattern among peers. Our results show that communities in SopCast are well defined by data exchange process. Moreover, the SopCast protocol and policies do not group peers from the same AS. The outcomes shown that SopCast application does not try to keep traffic inside the AS, since the peers are not organized according their AS. The presented characterization may be useful to developers create new protocols and algorithms that reduces the transmission cost on the P2P network.*

Keywords: *Internet, Peer-to-peer, SopCast, dynamics, communities.*

1. Introduction

During the last decade, there has been a wide interest in characterizing and modeling the structure of various networks, from neural networks, to the web, to Facebook friends. Real networks are inherently dynamic in the sense that both nodes and edges come and go over some time-scale. Besides, real data trace of these networks are huge and not easy of manipulating and extracting main and useful characteristics.

The challenge of understanding systems with different natures, behavior and dynamics leads us to applying the network science theory. In this paper, we are interested in studing P2P-TV applications under a set of network properties, taking into account the systems dynamics over time. In other words, we do not restrict our analysis on the static view of the systems. We consider the system as a set of views that can change over time. We focus on P2P-TV applications given their success over the last years, bringing TV line channels to the users home through the Internet. Among several P2P-TV applications found in literature, SopCast is currently the most popular P2P live application [Bermudez et al. 2011]; this application has hundred of thousands of users spread worldwide. The characterization of this application became a very important topic in networking field. However, this task is made particularly challeging by the fact that most of these applications have proprietary protocols.

The characterization presented in this work also shows the importance of group participants considering which network it participates (AS). The outcomes indicates no concern for traffic between networks.

2. Related Work

Despite the large number of works devoted to understand P2P-TV applications, most of them rely on characterizing the static view of these systems [Hei et al. 2007],[Ciullo et al. 2010] and [Vu et al. 2007]. These works do not worry about either understanding the sytems dynamics or analysing how it evolves over time. But, given the importance of characterizing systems dynamics, a small set of recent papers are focused on this topic. In what follows we cite those that are more in line with our work.

Borges et al. [Borges et al. 2012] provided a SopCast client characterization. The authors active crawls data from real SopCast channels, modeling the client behavior. The measurements are perfomed using the PlanetLab[1] infrastructure. Tang et al. [Tang et al. 2009] also use PlanetLab to investigate SopCast topology, but they focus on neighborhood metrics, for instance, link duration and nodes degree. Although the authors characterize and model users' behavior during a live transmission, they do not inspect groups formation. Moreover, as authors focus on client behavior (as client session time and partnerships) they do not further investigate topology properties and how they evolve over time.

Bermudez et al. [Bermudez et al. 2011] proposed a methodology that allow distinguish and investigate three different graphs from the overlay SopCast topology.

The authors introduce the definition of *group of peers*, based on the peers contribution (data upload and download). Different from them, we show evidences that SopCast

[1] www.planet-lab.org/

do not use any AS based algorithm to cluster their users. Moreover, our crawling methodology isolates external factors that may obscure the SopCast application behavior.

Different from a large number of works found in literature, we focus on characterizing SopCast applications under network properties, mainly the *community* formation process over time. It is worthing nothing that the mathematical modeling used in this paper always takes into account the SopCast dynamics.

3. System Modeling

We devote this section to present the mathemathical model used for representing the data collected over the set of experiments described on the Section 4. The mathematical model is importante for defining the set of network properties that will guide the SopCast's dynamics characterization. We also introduce how we identify communities in SopCast application.

3.1. Overlay Mathematical Model

SopCast overlay dynamic is mathematically represented by a family of graphs $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ indexed over time; $\mathcal{V}_t$ is the set of nodes and $\mathcal{E}_t$ is the set of directed links. Parameter t spans over $\{T_0 \leq t \leq T_N\}$. Let us define Δ as the interval between two consecutive *snapshots*. Each graph $\mathcal{G}_t$ aggregates the system information over δ seconds. W.l.o.g, let us consider two consecutive *snapshots* T_0, T_1, with $T_1 = T_0 + \Delta$. In this case, $\mathcal{G}_{T_0}$ aggregates information over $[T_0, T_0 + \delta - 1]$ seconds; $\mathcal{G}_{T_1}$, instead, aggregates information over $[T_1, T_1 + \delta - 1]$ seconds. In other words, we are aggregating information using a sliding window mechanism, with a δ-seconds duration. Characterization analysis is performed between two consecutive *snapshots*. Here, *network* and *graph* terms mean the SopCast overlay.

Let us focus on how links are established among nodes. Let us consider two nodes u and v. If node u has sent at least one chunk of video to node v (*vice-versa*) during the analysed δ seconds, link (u, v) (*vice-versa*) exists. Links are weighted by the total of traffic exchanged between any pair of nodes, during δ seconds. Given the application dynamics, it is clear that a given link can exist in *snapshot* T_i, but not in *snapshot* T_{i+1}.

3.2. Community Identification

Most of real networks show *community structure*, i.e., groups of nodes that have a high density of links among them, with a lower density of links between different groups. Communities can be arranged follow some rules based on specific characteristics related to the entities that formed them. The understanting of community exstance as well as the pattern formation is one of many important tasks in network science theory.

For idendifying communities, several methods can be found on literature. In this paper, we use the method called *FastGreedy* [Clauset et al. 2004], given that it is possible to assign weights to the conection between peers. Moreover, *FastGreedy* returns the total number of existing communities in each time interval.

Roughly speaking, in *FastGreedy*, it is possible to assign a sort of *connection weight* to nodes pair in the network. Here, we define *connection weight* as the total number of chunks a particular peer send to another one, for any pair of pees in the network. In this way, community identification is biased towards to put together peers that exchanged a large amount of data among them. Community identification and characterization are perfomed over all snapshots, with t spaning over $\{T_0 \leq t \leq T_N\}$. Each snapshot has a total number of c communities. Each community, in turn, has a population size of s peers.

In section 5.1, we will explore and justify our choice of putting together, in the same community, peers that excange more data among themselfs.

4. Experimental Setup and Crawling Methodology

Our work relies on a set of SopCast data we have collected from a controlled environment on PlanetLab. Its P2P architecture is based on a data-driven mesh overlay with a special peer known as *server*. The server peer generates the live content, splits it into data pieces (chunks) and starts the live streaming dissemination. To receive the live content, a peer explicitly requests each chunk it needs to one of its partners.

We set up a private SopCast channel in order to isolate our P2P swarm. Most important, we have the complete view of overlay network during all the experiment, which is essential for a better characterization of the SopCast's dynamics. Peers in our experimental environment have heterogenous resources and are spread over the world. We do not impose any additional constraint to peers. Then, all peers behavior and group dynamics we analyze are imposed by heterogenous peers bandwidth and by protocol issues.

Server peer encodes and transmits an one-hour 500 kbps video. We have performed 5 experiments, where all crawlers join the SopCast channel at the same time and remain connected through all the experiment. Each log experiment has about $16GB$ of total data.

During our experiments, the set of PlanetLab machines we used act as both regular SopCast peers and data crawlers. Each crawler collects and stores data regarding to all packets it exchanges with its partners, and these traces are later merged to rebuild the overlay network. We use Wireshark[2] to collect all SopCast related traffic. We have used the largest number of crawlers as possible, reaching a total of 389 PlanetLab nodes.

Every second, each crawler stores the packet time-stamp, its source and destination as well as its size information. During our analysis, as we focus on the content distribution process, a chunk of video is distinguished from a control packet based on its size: full payload packets with more than 400 bytes are classified as a chunk of video (see [Hei et al. 2007] and [Tang et al. 2009]). We use only data packets because when peers trade data, we are sure that they have interacted each other.

For each experiment, we have merged all log files to rebuild the SopCast overlay network as a sequence of snapshots. For accomplishing this task, we have used the packets characteristics gathered by the crawlers. Sliding window snapshots are five-seconds long and each experiment contains exact $3,295$ snapshots.

5. Characterization Results and Discussion

In this section we characterize the SopCast communities formation and its properties. We start by first discussing SopCast P2P overlay properties. Next, we analyze the communities key metrics as the number of communities and their size, communitie diameter and

[2]http://www.wireshark.org

the shortest path distribution between two communities members. Finally, we investigate the traffic pattern between peers and communities on SopCast.

5.1. SopCast Overlay Characteristics

SopCast is a mesh-like P2P live streaming application. In such application, peers do not organize themselves in a clear. When peers change their parternships, they change the overlay topology. Moreover, we observe that the SopCast P2P streaming (and overlay formation) protocol forces peers to constantly search for new partners. As natural consequence, the P2P overlay evolves during the live transmission.

A key propertie that evolves during the live transmission is the graph diameter. Large graph diamter impacts on overall streaming latency. Figure 2 shows the cumulative distributions of the P2P overlay. As we can see on this figure, the SopCast overlay diamater varies from very low values as 2, to large values as 12. But, in almost 90% of the cases, the SopCast overlay presents a mean diameter as low as 6 hops. Larger SopCast networks may have larger diameters but, the major number of SopCast channels are not crowded . According to our experience, channels tend to have 100 order users. We just observe crowed channels during events, as soccer matches. In sum, as we previously discussed, the experiments we have conducted represents the most commom SopCast channel.

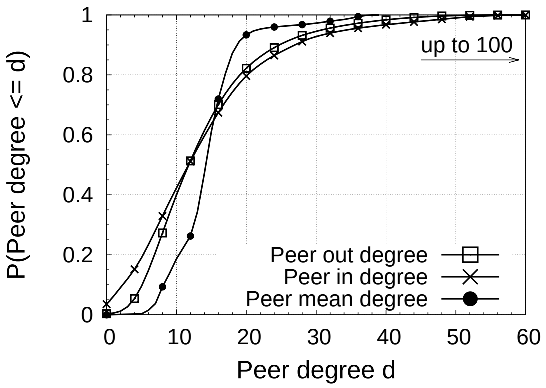

Figure 1. Peers In/Out Degree.

We also observe a low path size between peers. Figure 3 presents the cumulative distribution function of end-points shortest path size. In the SopCast overlay network, more than 96% of peers end-to-end connections presents only 3 hops. Both, low overlay diameter and low size end-to-end connections may indicate a fast streaming chunk dissemination.

Figure 1 shows the peer degree distribution during a live streaming session. Peers in SopCast presents 14.85 mean peer degree (whit a high C.V. of 0.34). Moreover, there is a non-negligible number of peers which present more than 20 partners during a snapshot. In all experiments we have conducted, we have observed the degree distribution pattern. This occurs because SopCast application does not provid to end users a customble configuration. In other words, the number of partners a peer tries to stabilish may be hardcoded in the SopCast application.

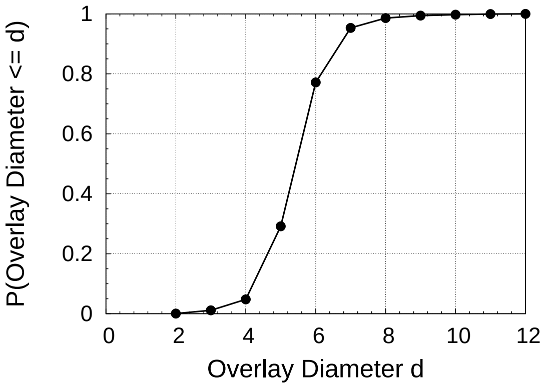

Figure 2. SopCast Overlay Diameter.

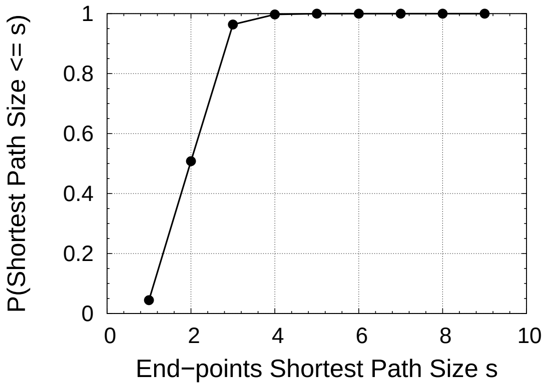

Figure 3. End-points Shortest Path Size.

Peers in/out degree distributions follow close the degree distribution. In fact, according to Figure 1, more than 90% of system peers present a in/out-degree lower than 15.

5.2. SopCast Communities Characteristics

The number of peers communities (or groups) changes during a live streaming session. Peers often change their partnerships, which in turn, alters the communities structure.

Figure 4 shows the cumulative distribution function of the number of communities during a SopCast live streaming session. In our experiments, the SopCast overlay presents 8.78 mean number of communities (std. dev. 2.8). More than 10% ovelay snapshots we analyze presents a small number of communities (less than 4). In this case, peers create large groups, reaching 250 members. Despite the existence of very big communities, more than 80% of communities have less than 50 members as show in Figure 5.

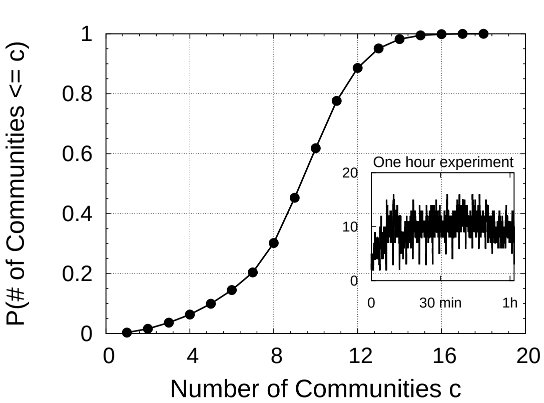

Figure 4. Overlay Number of Communities.

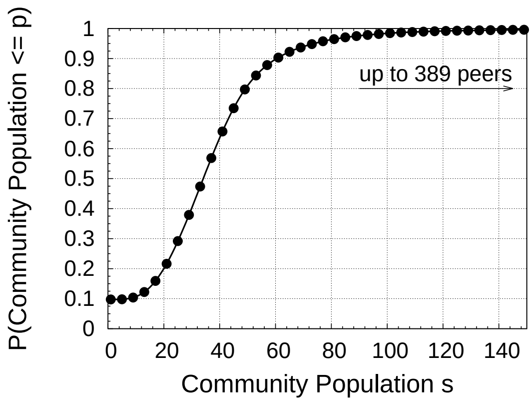

Figure 5. Community Population (number of peers).

As occurs to the SopCast overlay diameter, communities also present a low diameter. Figure 6 presents the probability distribution function of communities overlay diameter. The most commom diameter values to a comminity varies from 3 to 7, but there is a non-negligible number of communities that forms a complete graph (diameter equals to 1). In this case, peers exchange data each other keeping almost all data transfers whithin the community. Possible reasons to the low community diameter are: the small size of the community subgraph and the greedy peer behavior (peers try to gather data from lots of partners in a small time slot).

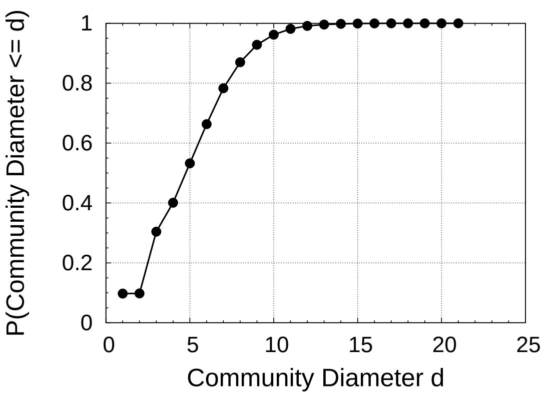

Figure 6. Communities Diameter.

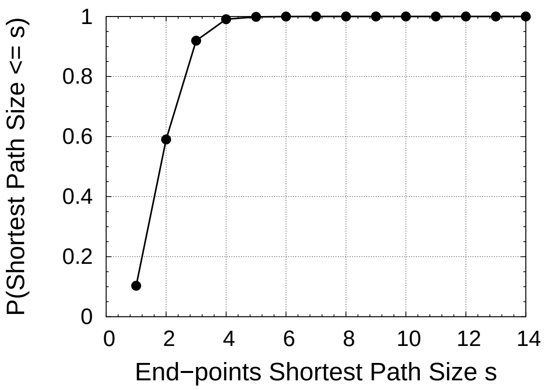

Figure 7. Community End-points Distance.

Figure 7 presents the distance between 2 end-points inside a community. Almost all nodes are only 2 hops distant each other. This low peers distance lead to a quick streaming dissemination and low latencies inside a community.

Finally, peers in a community have a much lower degree when compared to peers in the overlay network. In fact, as shown in Figure 8, the peer mean degree inside a community drops from 14.85 (mean network peer degree) to only $4.$ and the In/Out degree also gets lower values. For both, in and out degree, more than 90% of system peers present a in-degree (or out-degree) lower than 15, while in a community this number drops to 4. These lower values clearly shows that, peers exchange data to its partners inside a community, but they also remains communicating to peers outside theirs communities. In the next section, we discuss the peers and communities traffic pattern in more detail.

5.3. Community Traffic Pattern

Figure 9 shows peers link and data exchange summary. During our experiments, we observe that the larger portion of data flow remains in the community a peer behave. For instance, more than 60% of data exchange are between the members of a community.

Despite the data flow exchange pattern, there is a large number of links between members of distinct communities. In this case, only 30% of the links behave to 2 end-points from the same comunity. In this case, we reinforce that communities are formed among peers which trade a considerable amount of data each other.

Moreover, according to Figure 10, a small number of peers maintains the streaming inside a community. In this case, less than 10% peers from a community maintains more than 90% of the upload we observe inside the community. These higly altruistic peers may receive incentives and system protection against attacks as they acts as a community mantainer.

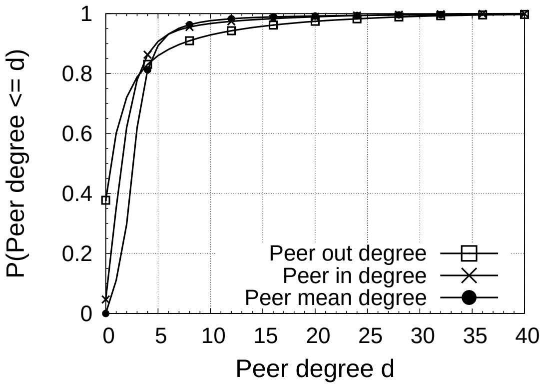

Figure 8. Peers In/Out Degree - Community

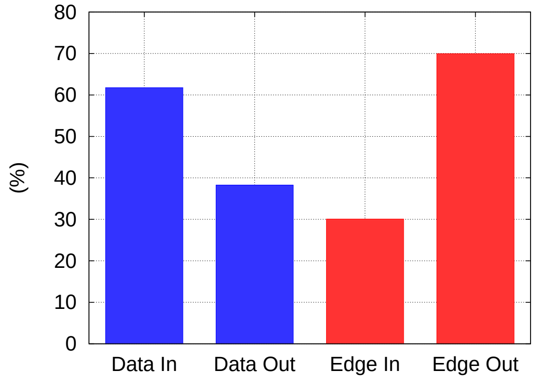

Figure 9. Data Exchange

During our experiments, we do not found any evidence that SopCast group its peers using theirs autonomous systems (AS) information. We have analized the 2 AS which have the largest number of peers among the PlanetLab nodes we use during our experiments. Figure 12 shows that, communities do not have a large proportion of peers from the same AS. For instance, less than 10% of communities we observe during our experiments present more than 50% of its members belonging to the largest AS (AS680). To the second largest AS (AS137), this probability drops to less than 5%.

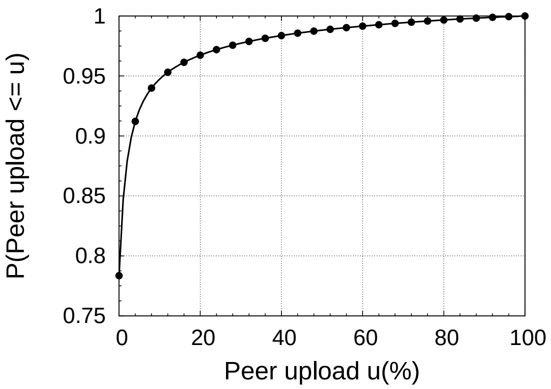

Figure 10. Peers Uploading Dist.

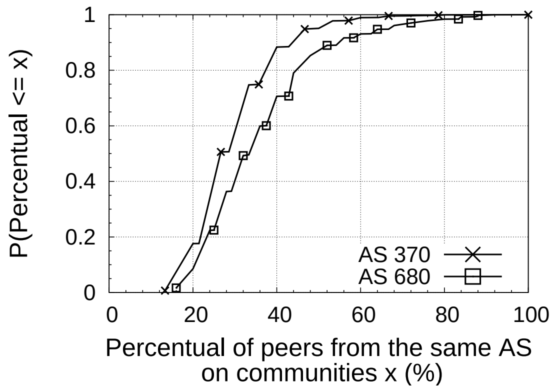

Figure 11. Percentual of Peers on Communities - Analisys using the 2 largest AS.

We also analize the data flow from peers belonging to the same AS. Figure 12 shows the traffic patterns we analize. First, we account the total data a peer exchange with a partner belonging to the same community (intra community traffic). In the opposite way, we check the amount of data a peer exchange to partners outside its community. Finally, we check the amount of traffic a peer exchange to its partners belonging to its AS (intra AS traffic). Note that intra AS traffic can be between peers belonging to 2 distinct communities.

Figure 13 and Figure 14 show the traffic pattern for both largest AS we found during 2 distinct experiments. For all other experiments, the traffic pattern follow close the traffic we plot in these 2 figures.

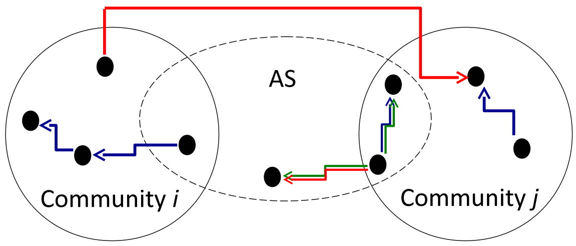

Figure 12. Max. peers in community

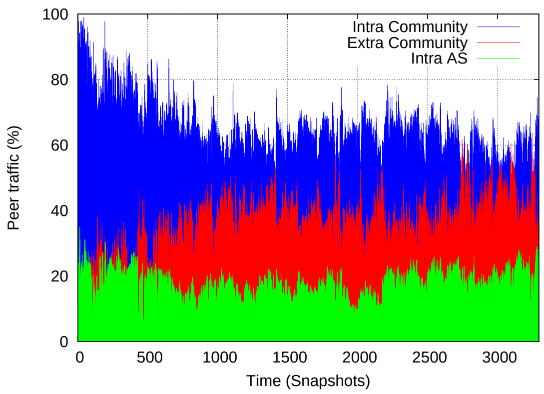

Figure 13. AS680

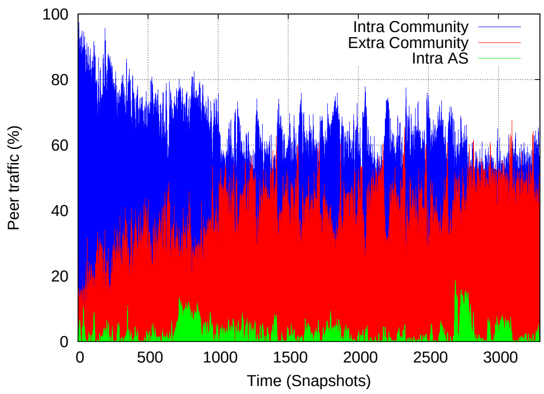

Figure 14. AS137

According to both figures, we note that intra-community traffic is larger than the extra-community traffic. For example, during the experiment from Figure 13, about 65% of the traffic remains inside the community. In both cases, the intra-AS traffic is low. During the first experiment, when we analize the largest AS, the intra-AS traffic is only about 18% off all traffic a peer exchange. For all other AS, including the second largest, the intra-AS traffic is much lower. For instance, as we present in Figure 14, the intra-AS drops to 2% mean.

In sum, peers group themselves into communities according to the amount of data they exchange each other. The larger portion of data peers exchange are with peers inside the same community they behave. The amount of data peers from the same AS exchange is low, and as consequence, we may infer that SopCast does not try to optimize the extra AS traffic, grouping peers according to their AS.

6. Conclusions

We observed that the SopCast protocol forces the peers to find new connections all the time, which directly implies in the size of diameter. The presented characterization shows that the graph formed by SopCast overlay has, for the most part, a low diameter.

We observed that the SopCast protocol forces the peers to find new connections all the time, which directly implies in the size of diameter and the shortest path. This values are so small when compared to its network size. With hundreds of peers, network diameter is smaller than 6 in almost of 90% time observation. Considering the shortest path size, 96% of peers end-to-end connections present only 3 hops. The mean degree is high, with value approximately equals to 5% of the total number of the peers. In this way, content fast spreads over the network and achievies the application real time constraints.

Communities follow the same pattern of SopCast overlay with respect to diameter and shortest path properties. Although, considering the mean degree, communities have a much lower degree when compared to peers in the overlay network. In and out degree are also smaller. These lower values clearly shows that, peers exchange data to their partners inside a community, but they also remains communicating to peers outside their communities.

Communities have different sizes. Peers create large groups, approximately reaching 75% of overall network. Despite the existence of very big communities, more than 80% of communities have less than 15% of peers of overall network.

Peers group themselves into communities according to the amount of data they exchange each other. The larger portion of data peers exchange are with peers inside the same community they behave. The amount of data peers from the same AS exchange is low, and as consequence, we may infeer that SopCast does not try to optimize the extra AS traffic, grouping peers according to their AS.

References

Bermudez, I., Mellia, M., and Meo, M. (2011). Investigating overlay topologies and dynamics of p2p-tv systems: the case of sopcast. *Selected Areas in Communications, IEEE Journal on*, 29(9):1863–1871.

Borges, A., Gomes, P., Nacif, J., Mantini, R., Almeida, J., and Campos, S. (2012). Characterizing sopcast client behavior. *Computer Communications*.

Ciullo, D., Mellia, M., Meo, M., and Leonardi, E. (2010). Understanding p2p-tv systems through real measurements.

Clauset, A., Newman, M., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Hei, X., Liang, C., Liang, J., Liu, Y., and Ross, K. W. (2007). A measurement study of a large-scale p2p iptv system. *IEEE Transactions on Multimedia*, 9.

Tang, S., Lu, Y., Hernández, J., Kuipers, F., and Van Mieghem, P. (2009). Topology dynamics in a p2ptv network. *NETWORKING 2009*, pages 326–337.

Vu, L., Gupta, I., Liang, J., and Nahrstedt, K. (2007). Measurement and modeling a large-scale overlay for multimedia streaming. In *in Proc. QShine*.

Submissão dos Artigos

Os orientadores Ana Paula Couto da Silva e Alex Borges Vieira estão cientes e de acordo com a submissão dos artigos dos seguintes alunos:

1) Bianca Portes;
2) Thiago Guarniere;
3) Rodrigo Duarte;
4) Abraão Guimarães;
5) Rafael Barra;
6) Thiago Boubee;
7) Francisco Henrique.

Atenciosamente,

Ana Paula Couto da Silva

Alex Borges Vieira