Overlay Convergence Analysis in P2P Networks: An Assessment of the 2PC Algorithm

Adriel Cardoso dos Santos
Dep. de Ciência da Computação
Universidade Federal de Alfenas
Alfenas, Brazil
a15001@bcc.unifal-mg.edu.br

Cristiano M. Silva
Dep. de Tecnologia (DTECH)
Universidade. Federal de São João del-Rei
Ouro Branco, Brazil
cristiano@ufsj.edu.br

Eliseu César Miguel
Dep. de Ciência da Computação
Universidade. Federal de Alfenas
Alfenas, Brazil
eliseu.miguel@unifal-mg.edu.br

***Abstract*—Peer-to-Peer (P2P) Networks for live streaming require low latency and low discontinuity in media transmission among peers. When latency is high, multiple users watch the video at different times, and in the case of high discontinuity, various parts of the media are not viewed by network users. Also, other factors compromise the quality of service of a P2P network of live streaming, such as the presence of a large number of peers that do not contribute to the distribution of the media, known as free riders, and the constant arrival and departure of peers during transmission, known as peer churn. An alternative to preserve the quality of a P2P network of live streaming is the usage of algorithms for construction and maintenance of the overlay network. One of these algorithms is the Peer Classification for Partnership Constraints (2PC), proposed to allow a large number of free riders in the network. 2PC acts by imposing constraints on partnerships between peers according to their contributions in the media transmission. 2PC was successfully tested on PlanetLab and its authors states the algorithm attracts high-contribution peers close to the server, while pushes low-contribution peers to the edge of the overlay. However, the authors have not demonstrated that this peer organization in the overlay actually happens. In this work, by analyzing the logs of the execution of the 2PC algorithm, together with the application of graph structures, we evaluated the application of the 2PC and identified that the partnership relationships between peers imposed by the algorithm organizes the overlay as expected.**

***Index Terms*—P2P, live streaming, mesh overlay construction, graphs, algorithms.**

I. Introduction

Peer-to-Peer Networks (P2P) are content distribution networks based on cooperation between their users. They aim to reach a large number of users, which we call *peers*. In them, peers establish partnerships with each other in order to exchange parts of the content of interest, reducing the need for direct connection between them and the transmission server, as in the server-client model. The topological organization formed by peers and their partners is called *overlay*. P2P networks can be used to transmit any type of content. In this work, we focus on the P2P network of live streaming, in which the data transmitted among peers are fragments of the video, known as *chunks*, generated and distributed by the media server.

In this case, it is desired that the media transmission be performed with low *latency* between peers and low *discontinuity* in the receipt of chunks, in order to maintain the quality of the user experience. Among the factors that can compromise the quality of service of a P2P live streaming network are: the constant arrival and departure of peers, known as churn [1]; the sudden arrival of a large number of peers in a short period of time, known as *flash crowd* [2]; and the presence of peers who refuse or are unable to contribute in the distribution of chunks, known as *free riders*.

The occurrence of these factors is common and, in many cases, inevitable. As an example, we can mention the growing number of mobile devices connected to the Internet. Such devices influence churn, because due to their mobility, they are constantly entering and leaving the network [3]. An option to preserve the quality of media transmission in P2P live streaming, even facing the mentioned problems, is the usage of algorithms for construction and maintaining the overlay.

Among several algorithms, *Peer Classification for Partnership Constraints* (2PC) [4] performed well in terms of stability of the P2P live streaming network in the experiments carried out, while maintaining a low media discontinuity and low latency between peers even in extremely unfavorable conditions. In the experiments, a flash crowd event was generated with an excessive number of free riders on the network.

2PC offers stability by imposing constraints on partnerships between peers on the overlay. These constraints are updated for each peer periodically by the algorithm, which are based on the contribution to the individual media distribution of each peer. The 2PC authors consider that the success of the algorithm is achieved because it attracts peers that present a high-contribution of media close to the server, while, at the same time, pushes low-contribution peers and free riders to the edge of the overlay.

Perhaps, the success of the algorithm may be due to another reason unknown to the authors of the 2PC. In this work, we studied the organization of the overlay promoted by 2PC by analyzing the *logs* of the experiments carried out in the work in which the 2PC algorithm was proposed.

As a contribution, we show that the success of 2PC can be explained by the way in which the algorithm constructs and maintains the overlay. This study was carried out with the application of graph techniques to observe the movement of peers in overlay throughout the experiments. Even though there was a rupture of partnerships between peers during the maintenance of the network, the algorithm demonstrated to attract the high-contribution peers close to the server and to push the low-contribution peers to the edges of the network, while preserving part of the partnerships already established to ensure the stability of the network.

The remainder of this work is organized as follows: Section II presents related works. It is organized to present the most important efforts in the area of interest, as well as to synthesize the 2PC algorithm. Section III describes the methodology used in this work to analyze the logs generated by the application of the 2PC algorithm. Section IV presents the results obtained in this work, and Section V concludes the work.

II. Background & Related Work

This section was organized into two parts. The first one presents the works related to research on P2P live streaming networks that are important for our study. Then, the second part presents the 2PC algorithm.

A. General questions about P2P live streaming networks

The topic of P2P networks has been widely researched in last decades and continues to encourage many works today. Presently, studies show that P2P strategies: can mitigate scalability problems in cloud services [5]; allow sharing chains of blocks in modern systems, such as Bitcoin [6]; and can be combined with CDN systems to increase scalability in media distribution, at low cost [7].

Many of the oldest researches aim to increase the robustness and quality of service offered by these networks. Cooperation is a premise for the smooth functioning of a P2P network. Thus, one of the many ways to increase the robustness of a P2P network is to implement techniques to encourage cooperation between peers [8]. Typically, studies point to two main strategies to deal with the problem of lack of cooperation from peers: the usage of mechanisms to encourage cooperation; and techniques for identification and punishment of uncooperative peers [8].

In terms of works related to encouraging the contribution of peers, we highlight: Xin Jin and Yu-Kwong Kwok [9] who propose a Striker strategy to coerce peers to cooperate based on the analytical insights derived from the repeated game, and Liu et al. [10] who present a mechanism capable of providing the peer with video quality proportional to its contribution to the network.

There is also the well-known Tit-for-Tat, a mechanism present in the P2P software for file sharing known as BitTorrent [11]. Some works use Tit-for-Tat on P2P live streaming networks, where uncooperative peers are penalized with a lower quality of service [12]. Such mechanisms that aims to encourage cooperation have disadvantages: they increase the network's overhead and the complexity of the system or impose constraints that limits the performance of the system [8]. These disadvantages become even more significant in P2P live streaming networks, which demand low latency and low media discontinuity.

However, instead of avoiding freeriding behavior, the 2PC algorithm was designed to construct and maintain the overlay without punishing the peers with low or no media contribution. Based on the *conscious free riders concept*[1], 2PC proposes that free riders declare they will not contribute to the media transmission upon joining the network. Thus, 2PC organizes the overlay to stream media to a large number of peers, supported by the media contribution of cooperative peers. The objective of this work is to study the evolution of the overlay organization carried out by 2PC in relation to the distance between peers and the media server.

B. Overview of the 2PC Algorithm

2PC algorithm performs the construction and maintenance of overlay by defining classes of peers and classifying them in such classes dynamically, according to their contribution to the media distribution in the overlay. Each of these classes establish partnership constraints between its peers and peers from other classes. The constraints are intended to organize the peers in the overlay in order to maximize the efficiency of the each peer's upload bandwidth usage.

In the experiments carried out by the authors of 2PC, four classes were proposed, known as: *hot*, *warm*, *cold* and *free rider*. Hot class is reserved for high-cooperative peers; warm class accommodates peers that present media cooperation, but not enough to be classified as hot; cold class accommodates low-cooperative peers. Finally, free rider class is reserved only for peers who do not contribute to the media transmission.

In this work, each peer p connected to the P2P network has a set of *neighboring* peers, Z, referring to the peers that p knows on the network, and a set N of *partner* peers ($N \subset Z$); p only receives and sends chunks to peers $q \in N$. N is divided into two sets: the input set $N_{in}(p)$, from which a peer p can receive chunks; and the output set $N_{out}(p)$, to which a peer p can send chunks.

For the 2PC execution, $N_{out}(p)$ was divided into two other sets: $N_{out}^{high}(p)$ reserved for peers whose class is considered to be high-contribution (hot and warm), and $N_{out}^{low}(p)$, reserved for peers whose class is considered to be low-contribution (cold and free rider). This separation of $N_{out}(p)$ ensures that there is no competition between high-contribution and low-contribution peers.

Partnership constraints defined by the 2PC algorithm between peers of different classes are applied when a peer p wants a peer q to provide him chunks of media. Traditionally, if q accepts the p's request, a partnership is established in which p is an out-partner of q, that is $p \in N_{out}(q)$. As a definition, 2PC determines that: peers from the hot class only accept peers from the hot and warm class in their N_{out}^{high} set; peers from the warm class accepts peers of any class in their N_{out}^{high} and N_{out}^{low} sets; peers from the cold class accepts only cold and free riders in their N_{out}^{low} set. Fig. 1 shows the possible out-partners (N_{out}) of a peer for each class, considering the

[1]Conscious free riders are those who notify their partners that they will not contribute to the P2P network [8].

separation of N_{out} in N_{out}^{high} and N_{out}^{low}: for each peer (on the left), the first column of the table (center) represents the set of partners reserved for high-contribution peers ($N_{out}^{high}(p)$), while the second one represents the set of partners reserved for low-contribution peers ($N_{out}^{low}(p)$). With this separation, there is no partnerships competition between high-contribution peers and low-contribution peers. Therefore, as the number of high-contribution peers increases, so does overlay's capacity to handle low-contribution peers.

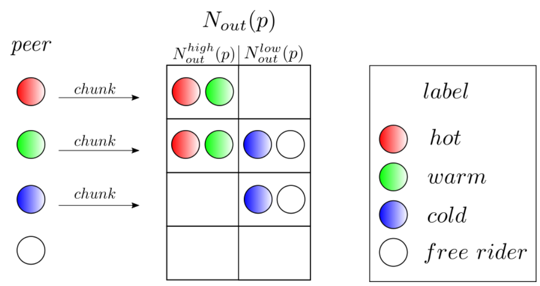


Fig. 1. Constraints imposed by the 2PC algorithm in partnerships between peers of different classes. On the left, we see the peers for each class. They accept out-partners according to the peers in the table (center of the figure).

2PC periodically analyzes the chunk contribution of each peer and reclassifies it, promoting or demoting it between classes. The reclassification of a peer p may lead to disconnection between it and some of his partners $q \in N_{out}(p)$, if the class of q is in disagreement with the constraints imposed by the new class of p. Also, 2PC determines that high-contribution peers have priority in establishing new partnerships, even when they request partnership to a peer with a full out-partner N_{out} set. Thus, if a peer p of the class warm, for example, with its two sets N_{out}^{high} and N_{out}^{low} complete, receive a partnership request from a peer r of the cold class, p will randomly disconnect a free rider from the set N_{out}^{low} (low contribution) to accept the connection of the new requesting partner r. The same can happen when a peer s from the hot class requests a partnership from p, in which case that p randomly disconnects a partner from the warm class. This reclassification and imposition of constraints promotes the maintenance of overlay, while avoiding competition between high-contribution peers with low-contribution ones.

The 2PC algorithm, in the configuration studied in this work, has the following characteristics: (i) when a peer requests to join the network, 2PC classifies them in the warm class; (ii) peers are reclassified throughout the entire network's execution time; and (iii) peers classified as free rider or cold cannot connect directly to the media server.

1) Computational Setup: The 2PC algorithm was implemented in TVPP [13], an open-source P2P live streaming system. TVPP defines a media server S, which generates the chunks of the transmitted video; a bootstrap server B; and the peers that will be part of the P2P network. The server S is also considered a peer.

The bootstrap server B manages the lists of active peers on the P2P network. Upon joining the network, a peer is registered by B and periodically receives a list of active peers from the network (its neighbors Z), with which it can form partnerships (N). Thus, in a network of thousands of users, a peer comes to know a subset of others peers that are on the network. B is also used to run the centralized 2PC algorithm.

Another assignment of the bootstrap B is to receive individual reports from each peer. Throughout the experiment, B elaborates logs with the reports received. These logs allow you to assess the state of the network at each instant. There are two types of logs in TVPP: the *overlay log* and the *performance log*, both generated periodically.

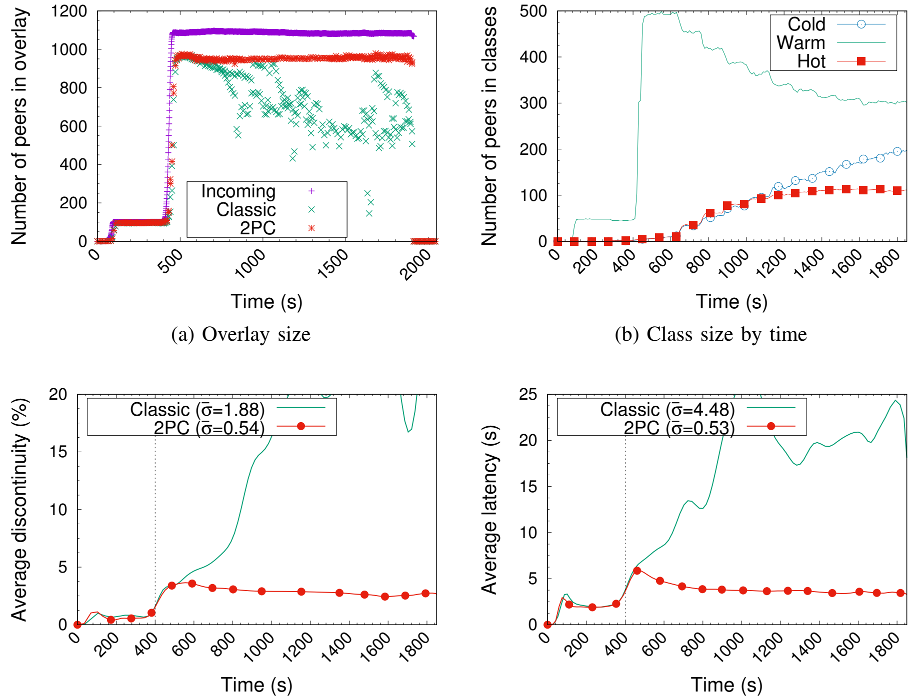
The overlay log reports in each of its entries a record that identifies a peer and its partners. This log consists of $T(n)$ intervals of approximately ten seconds. In a $T(n)$ interval, each active peer on the network generates an entry in the log. For our analysis, we defined a interval $T(n)$ by the media server S entries appearances. In other words, each $T(n)$ is composed of all occurrences of active peer entries on the network between the entries $E_{(n)}$ and $E_{(n+1)}$ of S.

On the other hand, the performance log reports, on each line, a record that identifies a peer and its individual performance information, such as: the state of the assisted media; statistics on the amount of chunks received, transmitted, lost and delayed; the number of partnerships established; and the classification of peer performed by 2PC, among other fields. Through media status information, it is possible to determine whether a peer watches the media continuously at any given time, in addition to determine the media latency for each peer in relation to the time that S generated the media.

The experiments were carried out at PlanetLab with about 110 nodes (computers). To generate a overlay with many peers, multiple instances were run on the same computer. In total, approximately 1000 instances of TVPP peers were run in five replications of experiments. In order to impose the latencies and packet losses present on the Internet, peers running on the same computer did not know each other. This eliminates the occurrence of partnerships that do not need to transmit media on the physical computer network. This constraint was implemented on the bootstrap server, when generating the lists of neighbors to be sent periodically to each peer.

As a time delay for the experiments, a overlay with approximately 100 peers is construct initially, and, at time of 400s, the peers waiting to join the already existent overlay arrive concurrently, forming the flash crowd event. Finally, Table I shows the upload bandwidth distribution of the peers present in the experiment. The values shown in Table I were defined in the original work of the 2PC algorithm [4]. To create a representative experiment setting, the authors of the 2PC used an existing flash crowd control method (proposed by Liu et al.) as their baseline [14]. The experiments demonstrated that the configuration of the Table I provided a challenging scenario for the P2P overlay, not supported by the baseline.

2) Results presented by the algorithm: In [4], the 2PC algorithm presented positive results regarding the stability of the network under adverse conditions such as flash crowd and the presence of a large number of low contribution and



(a) Overlay size

(b) Class size by time

(c) Media discontinuity over time

(d) Latency over time

Fig. 2. Comparison: P2P networks with and without the application of the 2PC algorithm. Fig. 2(a) shows the number of peers in the network, that is, the amount of peers that were executed (purple line) and the amount that remained connected to the network (red and green lines). Fig. 2(b) shows the amount of peers in each class over time. Note that the class warm loses peers to the other classes throughout the experiment, which tends to stabilize. Fig. 2(c) shows the average discontinuity for the experiments, presenting the standard deviations of each one, at the top. And finally Fig. 2(d) presents the average media latency. (Source: [4]).

TABLE I
NETWORK PEER CONFIGURATION

Upload	*Upload* (Mb/s)	*Share*
high	4.0	9%
average	2.5	17%
low	1.5	24%
nonexistent	0.0	50%

free rider peers. Such results were obtained by analyzing the performance logs of the experiments carried out.

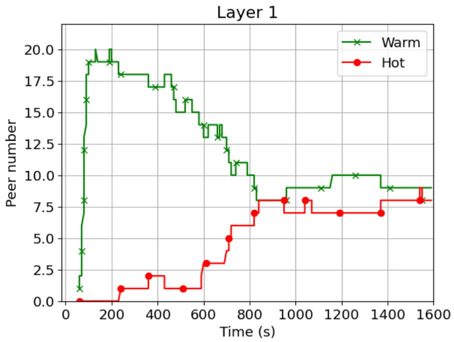
Fig. 2 presents the results for two types of experiments: One with the application of the 2PC algorithm and the other without the algorithm, referenced by the 'Classic' label. Fig. 2(a) shows the number of peers connected to overlay. The purple dots (at the top) indicate the total of peers performed in the experiment. It is visible that in the 'Classic' experiment (without 2PC), the number of connected peers is unstable over time (green dots). The number of peers connected with the 2PC remains stable throughout the experiments (red). The green and red dots are the result of overlapping the dots from all repetitions of the experiments. Fig. 2(b) shows the evolution in the classification of peers over the course of the experiments by class. We can see in time 400s the occurrence of the flash crowd with all peers entering the warm class. In this case, the lines are the average of all experiments.

Figs. 2.c and 2.d describe, consecutively, the average of the media discontinuity and the average of the latency of the chunk distribution. Those metrics describes the quality of the transmission of the media on the network. It is observed in these figures that both metrics remain low and stable for the network with the application of the 2PC algorithm, which does not happen in the 'Classic' case. In the figures label we present the standard deviations values.

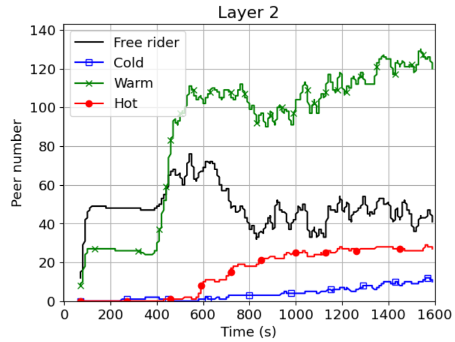
Although positive effects of 2PC are observed in the presented results, it is not possible to observe the evolution of the overlay organization. In this work, through the analysis of partnerships from the overlay logs, which were not evaluated in [4], we intend to show whether 2PC provides these good results due to the approximation of high-contribution peers to the server, and the drifting of low-contribution peers to the peripheries of the network.

III. METHODOLOGY

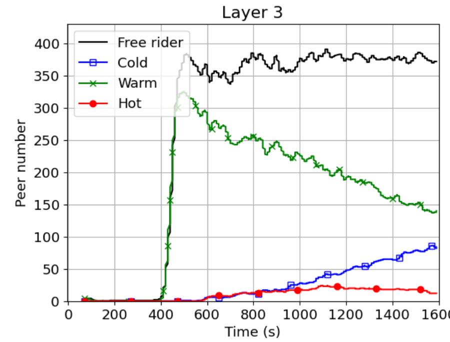
We use the overlay logs to graphically reconstruct the P2P network while observing the class defined by 2PC for each peer. For every interval $T(n)$ in the overlay log, it is possible to generate a graph $G(V, E)$ that represents a P2P overlay, V being the set of peers on the overlay whose media is received with little to no discontinuity and E the set of edges, where $\forall (p, q) \in E \iff q \in N_{out}(p)$. Considering the potential



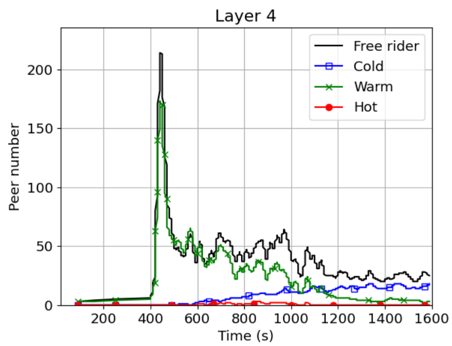
(a) Q_1: peers 1-distance away from S



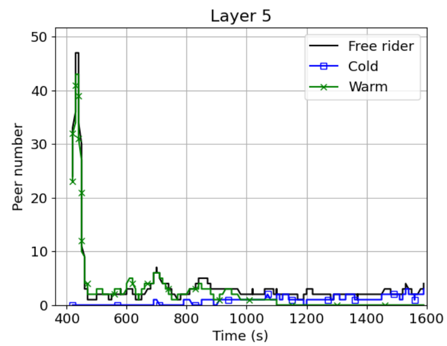
(b) Q_2: peer 2-distance away from S



(c) Q_3: peers 3-distance away from S



(d) Q_4: peers 4-distance away from S



(e) Q_5: peers 5-distance away from S

Fig. 3. Figure shows the five layers of the P2P overlay. For each layer, we show the number of peers per class as a function of time. As a summary, we observe the migration of peers between the

number of N_{out} of a peer, the graph G is considered a dense graph and there is an exponential number of paths in which a chunk can be transmitted from the server S to a given peer p. Given the large number of paths and our objective of observing the distance of the peers per class from S, we chose the shortest path as a metric, that is, the distance $d(u, v)$ is the minimum number of edges connecting the u vertex to the v vertex.

The shortest path $d(u, v)$ is not necessarily the path through which the chunks were transmitted between the u and v peers. In this case, the actual transmission path was not registered in the logs. Other distance metrics were also evaluated in this work, such as: (a) average of the K-shortest paths, which was discarded because it did not present discrete values, making it unfeasible to define layers in the overlay; and (b) median of the K-shortest paths, which was discarded as it presents low variation in the length of the median paths and, thus, tends to group all peers in a few layers We therefore adopt the shortest path as a distance reference for our purposes.

Given the graph $G(V, E)$ generated from an interval $T(n)$, we apply the Dijkstra's algorithm using the server S as source. The result is a tree R in which S server is the root. Let P be the set of all peers belonging to the P2P overlay and $e(p)$ the distance from the S server to the peer p, we can partition P into Q_i layers where $p \in Q_i \iff e(p) = i$. By definition $\bigcup_i Q_i = P$. Note that the further the layer is from the server, the more peers it can contemplate.

For each tree R generated from an interval $T(n)$, we count the number of peers in each layer according to their classes defined by the 2PC algorithm. Thus, we compared the trees between the intervals $T(i)$ and $T(i+1)$ throughout the experiments to identify a convergence in the overlay network in relation to the new partnerships between peers, which is consequence of the reclassification promoted by 2PC.

IV. Results

Fig. 3 presents five graphs that describe the number of peers over time for each Q_i layer. These graphs are the results of the analysis of the overlay logs generated by the experiments carried out in [4], the same experiments that show the results of the application of 2PC observed in Fig. 2. The curves in the graphs shown in Fig. 3 have been normalized using the Gaussian filter.

Q_1 layer, of peers connected directly to the server, is exclusively composed of peers classified as hot and warm. This is due to the constraint imposed by the 2PC algorithm: only high-contribution peers can be partners of S. In the Q_1, until $800s$ of execution, we observe that there is a decrease in the number of warm peers and an increase in the number of hot peers. This effect is the result of two characteristics of the 2PC algorithm, which classifies every peer as warm when entering the network, and reclassifies them throughout the experiment based on its individual media contribution to the network. Consequently, peers that were initially classified as warm, regardless of their upload bandwidth, are reclassified into more appropriate classes when proving their contribution. Fig. 2(b) shows this effect.

The peers that make up the Q_2 layer are mostly classified as warm. The number of hot peers, in this same layer, exceeds the number of cold peers. In this layer the number of free riders fluctuates. This fluctuation is explained by the fact that cooperative peers prefer to disconnect a random free rider from their $N_{out}^{low}(p)$ set to accept a cold peer partnership request. The decrease in the number of free riders in the Q_2 layer also brings, as a consequence, the migration of these peers to more distant layers from the server. Thus, we observe the tendency of the Q_2 layer to be populated mainly by high-contribution peers.

Q_3 layer is mostly composed of free riders, whose number remains stable. Most of the network's free riders are in this layer for most part of the experiments. A decrease in the number of warm peers and an increase in the number of cold peers are visible, due to the initial classification and the 2PC reclassification routine. The trend of warm peers and cold peers curves implies that, at some point, there may be more peers from the cold class than peers from the warm class. We conclude that the Q_3 layer is composed mostly of low-contribution peers.

Q_4 layer has a peak in the number of peers at the moment of $400s$, when the flash crowd event happens, and subsequently converges into a small number of peers, compared to the previous layers. This is because the peers tend to approach the server as the network stabilizes. The layer is composed mostly of free riders and peers from the cold class, indicating that it is a layer for low-contribution peers and free riders.

Finally, Q_5 layer only exists after $400s$ time, when the flash crowd happens. The layer shows a total absence of hot peers, and an absence of warm peers after $1100s$ time. This layer is exclusive for low-contribution peers and free riders.

In summary, it is observed that the 2PC algorithm promotes for the layers Q_1 and Q_2 the composition mostly by high-contribution peers and for the layers Q_3, Q_4, and Q_5 (more distant from the servers) the composition mostly of peers of low or no contribution.

V. Conclusion

In this work, we sought to prove the form of overlay organization carried out by the 2PC algorithm. For this, we analyzed the overlay logs of the experiments carried out by the authors of the algorithm in [4]. In this analysis, we developed a software called TVPP Log Parser (TLP), capable of parsing the overlay and performance logs, in order to build the graphs that represent the P2P network for each moment of the experiment. With these graphs, we could observe the evolution of the overlay network at each moment, which made clear the migration of high-contribution peers close to the server, while low or no contribution peers were pushed to the edges of the network.

We conclude that the 2PC algorithm constructs and maintains the overlay of the P2P networks as expected, but so far not confirmed in other works. Thus, this work contributes to completing the understanding of the 2PC algorithm, even with the complexity observed in the structure of the overlay generated by the large number of different paths that the media chunk can take.

As future work, we propose that the same methodology described in this paper should be used again, but instead of considering the shortest path to generate the media distribution tree, the real chunk transmission path among the peers should be used, which is not currently registered in the logs. With the real path, it will be possible to generate a much refined overlay representation, in this way, the study of the construction and maintenance of the overlay can be more precise.

Acknowledgments

This research was partially funded by the CNPq, CAPES, and FAPEMIG.

References

[1] H. Terelius and K. H. Johansson, "Peer-to-Peer Gradient Topologies in Networks With Churn," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 2085–2095, 2018.

[2] T. Y. Chung and O. Lin, "A Batch Join Scheme for Flash Crowd Reduction in IPTV Systems," in *Proc. of 2011 IEEE 17th International Conference on Parallel and Distributed Systems*. IEEE, 2011, pp. 823–828.

[3] Z. Ou, E. Harjula, O. Kassinen, and M. Ylianttila, "Performance Evaluation of a Kademlia-Based Communication-Oriented P2P System Under Churn," *Computer Networks*, vol. 54, no. 5, pp. 689–705, 2010.

[4] E. C. Miguel, "Overlay Construction Strategies for Peer-to-Peer Live Streaming Systems," Ph.D. dissertation, Universidade Federal de Minas Gerais-UFMG, Brazil, 11 2017.

[5] N. W. B. A. Wahid, K. Jenni, S. Mandala, and E. Supriyanto, "Review on cloud computing application in p2p video streaming," *Procedia Computer Science*, vol. 50, pp. 185 – 190, 2015, big Data, Cloud and Computing Challenges.

[6] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks," *CoRR*, vol. abs/1904.04098, 2019.

[7] H. B. Hadj Abdallah and W. Louati, "Ftree-CDN: Hybrid CDN and P2P Architecture for Efficient Content Distribution," in *Proc. of 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2019, pp. 438–445.

[8] J. Oliveira, Í. Cunha, E. C. Miguel, M. V. Rocha, A. B. Vieira, and S. Campos, "Can Peer-to-Peer Live Streaming Systems Coexist With Free Riders?" in *Proc. of IEEE P2P 2013*. IEEE, 2013, pp. 1–5.

[9] X. Jin and Y.-K. Kwok, "Coercion Builds Cooperation in Dynamic and Heterogeneous P2P Live Streaming Networks," *Computer Networks*, vol. 81, pp. 1 – 18, 2015.

[10] Z. Liu, Y. Shen, K. W. Ross, S. S. Panwar, and Y. Wang, "LayerP2P: Using Layered Video Chunks in P2P Live Streaming," *IEEE Transactions on Multimedia*, vol. 11, no. 7, pp. 1340–1352, 2009.

[11] B. Cohen, "Incentives Build Robustness in BitTorrent," in *Proc. of Workshop on Economics of Peer-to-Peer systems*, vol. 6, 2003, pp. 68–72.

[12] T. Locher, R. Meier, R. Wattenhofer, and S. Schmid, "Robust Live Media Streaming in Swarms," in *Proc. of the 18th international workshop on Network and operating systems support for digital audio and video*, 2009, pp. 121–126.

[13] J. Oliveira, R. Viana, A. B. Vieira, M. Rocha, and S. Campos, "TVPP: A Research Oriented P2P Live Streaming System," *SBRC 2013-Salão de Ferramentas*, 2013.

[14] F. Liu, B. Li, L. Zhong, and B. Li, "Flash Crowd in P2P Live Streaming Systems: Fundamental Characteristics and Design Implications," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1227–1239, July 2012.