Join rate improvements in P2P live streaming based on topological aspects during flash crowds

Eliseu César Miguel[1], **Fernando C. S. Coelho**[2],
Bruno Morgan[1], **Murilo Calado Junior**[1],
Ítalo Cunha[2], **Sérgio Campos**[2]

[1]Instituto de Ciências Exatas
Universidade Federal de Alfenas (UNIFAL-MG)
Alfenas, Brasil

[2]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Brasil

{eliseu,bruno.morgan,murilo.cesar}@bcc.unifal-mg.edu.br,
{fccoelho,cunha,scampos}@dcc.ufmg.br

Abstract. *In P2P networks peers share content in a topological overlay above the physical network. This is fundamental to network's performance. However, the simultaneous arrival of many peers, known as* flash crowd*, can affect network topology and disrupt content transmission. Current studies frequently separate flash crowd from topology arrangement techniques. In this work, we set topological partnership restrictions to preserve the existing mesh during a flash crowd. Using a* parallel network *technique, incoming peers are isolated in sub channels to avoid new relationship interference between newcomers and existing peers. This is particularly important because incoming free riders may affect existing cooperative peers significantly. In our experiments, we show that this restriction has allowed the transmission to remain unaffected in the presence of a flash crowd six times greater than would have been possible without the proposed technique.*

Resumo. *Nas redes P2P para vídeo ao vivo, os clientes compartilham o conteúdo em uma organização topológica sobreposta à rede física, fundamental para o bom desempenho das redes. Contudo, a chegada de grande número de clientes simultaneamente, o* flash crowd*, em casos extremos, pode desestruturar a topologia da rede e interromper a transmissão do conteúdo. Em geral, os trabalhos separam estudos sobre flash crowd dos estudos de topologia. Neste trabalho, configuramos restrições topológicas de parcerias para preservar a malha durante o flash crowd. Com uso da técnica de* redes paralelas*, os clientes recém chegados foram isolados em subcanais de transmissão para evitar que os clientes não cooperativos interferissem nas novas parcerias. Em nossos experimentos, mostramos que estas restrições de parcerias permitiram ingressar simultaneamente uma quantidade de clientes seis vezes maior que a quantidade ingressada sem o isolamento dos clientes não cooperativos.*

1. Introduction

Differently from the client/server content distribution model, P2P networks allow live streaming to a large audience without reliance in server's upload bandwidth. In these networks, each client needs to share the received content, what makes these systems scalable and with low running costs. Real systems allow thousands of simultaneous users in many video distribution channels [UUSee 2008, SopCast 2008, PPLive 2008, TVU 2013]. In these systems, peers establish partnerships in a decentralized way what creates a mesh topology over the physical network to exchange the media content. In this case, the content is a real time video splited in *chunks*.

Many problems may arise under the P2P network during topology maintenance and during chunks distribution. P2P networks are constantly hit by the peer's dynamism, known as *churn*, which is caused by peers leaving and joining the network. Churn may harm the systems [Zheng et al. 2011]. Furthermore, the presence of uncooperative peers, known as *free riders*, reduces the chunks availability what decreases chances of good partnerships to receive data [Adar and Huberman 2000, Moltchanov 2011, Karakaya et al. 2009, Krishnan et al. 2004, Meulpolder et al. 2012].

Many P2P strategies were proposed over the time in order to ensure a media transmission with low chunk latency and low chunk discontinuity. In this case, latency accounts for the time gap between the chunk generation on the server up to it's visualization in the peer. Basically, these strategies seek to establish good relationships among peers and at the same time to bring cooperative peers close to the media server [Payberah et al. 2011, Lobb et al. 2009, Fortuna et al. 2010]. These topology interventions may require time so that peers acquire neighborhood awareness until they establish promising partnerships.

On the other hand, flash crowd happens suddenly. So, there are techniques to control the joining rate which are capable of dealing with the huge amount of newcomers what prevents the network from disruption [Liu et al. 2009, Chen et al. 2011, Li et al. 2008, Liu et al. 2012]. Among these techniques, in [Liu et al. 2012], the flash crowd is controlled by splitting new peers in *joining batches*[1] which receives joining permission for joining up the network in time slots. Thus, these time slots allow the network to establish new partnerships. In this way, this procedure is repeated until the end of all the flash crowd event. In this technique, many peers await in queues for accessing the network. In this context, upload bandwidth is wasted while potential cooperative peers are waiting their turn to join the mesh. In cases where the waiting time is longer, peers may give up from the waiting queue, what becomes a major issue.

Another issue with flash crowd control technique is to estimate network's resources in order to know it's joining potential. Network's surplus resources are parameters to mathematical models which manages *joining batches* creation and are used to define each joining time slot. Although, as happens in [Liu et al. 2012], previous works about flash crowd focus in resources related to the idle upload bandwidth capacity and some aspects such as the partnership amount allowed to each peer, without any consideration around network's topology.

[1]Normally, batch is used for tree topology, and *slot* for mesh topology. However, we use batch in this context to distinguish from *time slot*

Differently from previous works, we fixed the bandwidth resources and peer's amount of partnerships in the network to evaluate elements related to the topology that can increase the amount of newcomers' peers in a joining batch without compromising the network's stability. We found that it is possible to change partnership criteria and create greater joining batches without any loss to cooperative or uncooperative peers. In our experiments, changes imposed to partnerships insulate newcomer free riders from cooperative peers resident in the network during the flash crowd event. This has allowed a peer joining rate six times greater in a single joining batch, without compromising the network's service.

As contributions of this work, we highlight: first, we show that it is possible to mitigate free riders negative presence effects during the flash crowd event by only setting restrictions to partnerships between free riders and cooperative peers. Furthermore, it is shown that these restrictions allow that both, cooperative peers and free riders, can join simultaneously the network without disrupting the live media transmission.

In a second place, we present a parallel network approach for handling flash crowd event. In this way, even with resources constraints, as will be discussed in this work, a large number of free riders peers are able to join the network without competition with cooperating peers resources. This scenario is possible because parallel networks were set to not allow partnership formation among free riders and cooperative peers that joined the network before the flash crowd event, what is enough for wiping out the risks offered by free riders exhausting network's resources.

Finally, we present *Free Rider Slice* as a simple way to restrict partnerships among free riders and cooperative peers without the need to implement parallel networks. We believe that free rider slice concept can be implemented in tandem with topological techniques to improve the network's live media distribution and to prepare the network overlay to receive a sudden flash crowd event.

The remainder of this work is organized as follows: Section 2 discusses related works. We describe TVPP, the real P2P streaming system that we use in our work, in Section 3. We explain our experimental method in Section 4 and present our results in Section 5. We offer conclusions in Section 6.

2. Related works

The P2P model is characterized by peers directly connecting to each other without intermediaries. The partners for a given peer can be classified into in-partners and out-partners. P2P systems may impose constraints on the number of in-partners (in-degree) and/or out-partners (out-degree). An in-depth look at the literature reveals that most of the works, such as [Traverso et al. 2015, Lobb et al. 2009, Payberah et al. 2011], do not impose fixed limits for the out-degree implicating in a non-zero chance that the number of out-partners exceed the bandwidth capacity of a given peer.

Therefore, additional techniques for choosing out-partners priority for chunk sending may be used [d. Silva et al. 2008, Wu et al. 2012]. Thus, low upload bandwidth peers can manage a large number of partners without cracking the video transmission. Unlimited out-degree may be an optional topology approach to improve overlay arrangement. However, according to [Lobb et al. 2009], large neighborhoods deliver a burden

to the network performance due to the need of each peer maintain and exchange a large amount of information with their neighbors, thus, wasting bandwidth and increasing the complexity of the scheduling algorithm.

Furthermore, overlay topology organization should dispose a distributed or centralized way to detect peers chunk upload contribution. Free rider identification is a challenge and demands time to be accomplished. *Conscious Free rider* was introduced in [e Oliveira et al. 2013]. In cases that upload bandwidth is limited, as happens with mobile peers, it is preferable that these peers join as conscious free riders. Thus, cooperative peers do not waste time requesting chunks for these uncooperative peers neither choose them for in-partnerships. On the other hand, in the flash crowd event, approaches such as in [Liu et al. 2009], ask for peer's bandwidth before bootstrapping each peer in the network. Then, peers are sorted by upload bandwidth and uncooperative peers, such as free riders, can be stalled for long time. This time is managed by *Deadline Alarm Queue* which avoids peers from giving up the joining stage. These works show that it is important to improve topology overlay organization during the flash crowd.

The majority of flash crowd handling techniques are based on peers joining by time slots. In this case, these time slots constraints allow networks to scale with the increase in the number of users, given the absence of multicast support in global network-layer [Rückert et al. 2015]. TOPT, presented in [Rückert et al. 2015], is built upon the state-of-the-art hybrid streaming approach called TRANSIT [Wichtlhuber et al. 2014] and consider to prepare the system overlay, beyond using only the time slots, for a flash crowd event. TOPT is a rare approach with this focus.

To mitigate flash crowd effects without advanced flash crowd control techniques, streaming systems such as COOLSTREAMING or PPLIVE rely on a large number of dedicated servers or the use of CDNs [Wu et al. 2012, Liu et al. 2012]. Thereby, it increases the systems cost, besides the challenges in dealing with the unforeseen flash crowd's occurrences. Differently, in this work we want to study uncooperative relationship constraints to make robust overlay with or without flash crowd event. We define a limited network bandwidth and fixed value of out-degree configurations to understand whether it is possible to mitigate flash crowd degradation without disrupting media transmission for all peer classes.

Despite other works, we believe that once defined both enough small in-degree and out-degree, the flash crowd consequences are softened, without sophisticated techniques for choosing out-partners priority. Furthermore [Liu et al. 2012] shows that neighborhood size does not boost peers' joining time. In this case, it have been compared neighborhood sizes in the range of 6-100 and found out that joining time of 20 against 100 peers neighborhood sizes, had similar joining time behavior.

Regarding free rider issues, we consider that their presence are allowed in network, but conscious free riders reduce the complexity of joining techniques and overlay management techniques. Then we chose these free riders to constrain network's resources on the experiments. In the same way we avoid any technique for choosing out-partners to send chunks and also, differently from [Liu et al. 2009], we allow continuous joining of peers in our experiments.

Moreover, we investigate whether it is possible to reach good P2P system's result,

without the need of sophisticated techniques, but through just configuring parameters such as neighborhood size. Finally, we believe that free rider's relationship constraints may help new topology organization works to obtain robust overlay meshes for facing flash crowd events.

3. P2P Live Streaming System

We define a P2P live streaming system as a system with a set of *peers* that collaborate with each other to watch a live media transmission. The *server* S is a special peer that encodes the video, splits the video into *chunks* (a chunk can contain multiple frames), and starts the transmission.

Each peer p has a set of *partners* $\mathcal{N}(p)$ for exchanging video chunks. To get more control over partnerships, $\mathcal{N}(p)$ is split between two subsets of partner peers: $\mathcal{N}_i(p)$ containing *in-partners* (partners that provide chunks to p) and $\mathcal{N}_o(p)$ containing *out-partners* (partners that receive video chunks from p).

The maximum number of in-partners is denoted by $N_i(p)$. Similarly, the maximum number of out-partners is denoted by $N_o(p)$. For S, $\mathcal{N}_i(p) = \emptyset$. In order to join a live streaming channel, a peer p registers itself at a centralized *bootstrap* server B, which returns to p a subset of all peers currently active in the system as potential partners. Peer p selects peers from this subset and tries to establish partnerships with them.

All relationship $p \in \mathcal{N}_o(p')$ requires $p' \in \mathcal{N}_i(p)$. Successfully established partnerships determine $\mathcal{N}(p)$. When p detects that one of its partners $p' \in \mathcal{N}(p)$ has been silent for longer than a predefined time period, p removes p' from $\mathcal{N}(p)$. It is not a problem because peer p periodically contacts the bootstrap server to obtain a new list of potential partners to replace the lost partnership.

Several works do not impose a fixed value for N_o of peers in the network. It ensures that if p is chosen by p' for $\mathcal{N}_i(p')$ implies that p should accept p' in $\mathcal{N}_o(p)$ [Lobb et al. 2009, Traverso et al. 2015]. On the other hand, we have imposed known values for both, N_i and N_o, while peers randomly choose in-partners. To organize the network topology, peers are able to accept a new out-partner even when $\mathcal{N}_o$ is full. In this case, p disconnects a random out-partner $q \in \mathcal{N}_o(p)$ with less out-partnerships than the new peer n, i.e., $N_o(q) < N_o(n)$. Afterwards, peer p remains unable to disconnect more out-partners to accept incoming partnership requests during the next 60 seconds to prevent overlay instability.

Each peer has a local buffer to store its video chunks. Periodically, peers exchange *buffer maps* with their partners $\mathcal{N}_o$ to inform them which chunks they have available. Each peer periodically checks which chunks it needs, identifies which partners $\mathcal{N}_i(p)$ can provide missing chunks, and then sends chunks requests accordingly. In this work, we schedule chunk requests using the earliest deadline first policy associated with *Simple Unanswered Request Eliminator* (SURE) [e Oliveira et al. 2013]. SURE allows each peer p to hold a black list containing non responding peers to avoid new requests faults. We do not limit the number of simultaneous (pending) requests. However, we limit the number of request retries to six to control each peer's opportunities to download a chunk and make download opportunities independent of buffer size. A peer considers that a request has timed out if it is not answered within 500 milliseconds. Finally, cooperative peers immediately serve their received requests in the order of their arrival.

Peers send their monitoring reports to the bootstrap server every 10 seconds. These reports, actually, include the number of chunks generated (only reported by the video server), sent, received, and the ones that missed their playback deadline; the number of requests sent, answered, and retried; the average forwarding path length, retry count, and time of arrival of received chunks; neighborhood size; and the number of duplicate chunks received. Thus, we use these peer reports to compute the performance metrics being evaluated: chunk latency and the chunk playback deadline miss rate.

4. Experimental Method

Our evaluation relies on real experiments that we have conducted on PlanetLab [PlanetLab 2009], using the P2P system, TVPP [Oliveira et al. 2013], and with five repetitions for each. We configured video and bootstrap servers in our university's network and used about 108 PlanetLab nodes as peers. The video server streams a 420 kbps video (about 40 chunks per second). Even though PlanetLab nodes have their own bandwidth and CPU restrictions, we impose additional upload bandwidth constraints on each peer to archive a restricted network in order to approach a realistic scenario.

Our experiments were performed in 1350 seconds. Firstly, the first 70 seconds initialize bootstrap and media servers. Then we construct a *first network*[2] with a group of 108 peers to support the suddenly arrival of 1080 new peer group from flash crowd that happens 350 seconds after each experiment beginning. In a total, we run around 1100 peer instances (i.e. 11 instances in each PlanetLab's node). To preserve packet loss and delay in a realistic scenario we do not allow peers' communication within the same PlanetLab's node.

To assess topological aspects we fixed some peers resource parameters. We are interested in understanding whether any free rider partnership restriction better improves flash crowd joining rate peers through all experiments with the same bandwidth and $\mathcal{N}(p)$ size network configurations. We set four types of upload bandwidth and partnership size for each peer. Table 1 shows peer's configuration. Since $N_o = 0$ for *class 0*, it defines a *conscious* free rider that informs to no one their *buffer maps*. Thereby, all free rider's partners know that they are unwilling or unable to upload data [e Oliveira et al. 2013]. Finally, all join peer groups have the same resources configuration.

Table 1. Network Peers Configuration

Peer Classes	Mb/s	Proportion	N_i	N_o
class 0	0.0	40%	10	00
class 1	1.5	27%	10	09
class 2	2.5	22%	10	20
class 3	4.0	11%	10	23

In order to study the flash crowd in this paper, we define: (i) *master network*, composed by the set of peers that joined on P2P network before the flash crowd's event; and (ii) *whole network* composed by the set of all active peers. Thus, before the flash crowd, *master network* and *whole network* are the same. Finally, we have a unique server media with $\mathcal{N}_o(s) = 20$ and no upload bandwidth constraint.

[2]In this work, the first network is called *master network*

5. Evaluation

In this section we evaluate the impact of free rider partnership limitation during the flash crowd event on P2P live streaming. We compare two scenarios, with and without free rider isolation, and show joined amount variation for the same network configuration. First, we present results for *common flash crowd technique* approach introduced in [Liu et al. 2012] whose results are previous benchmarks for this work. After it, we evaluate a parallel network to impose free riders' partnership limits and show that it is possible to allow larger simultaneous number of peers joining the network without disrupting the media distribution. Parallel networks are our first attempt to assess that free rider isolation can be another way to improve the amount of joined peers in P2P live streaming networks. Finally, despite parallel networks issues we present *Free Rider Slice*: an easy way to setup the network in order to mitigate free rider negative impact on the mesh.

5.1. Common Flash Crowd Technique

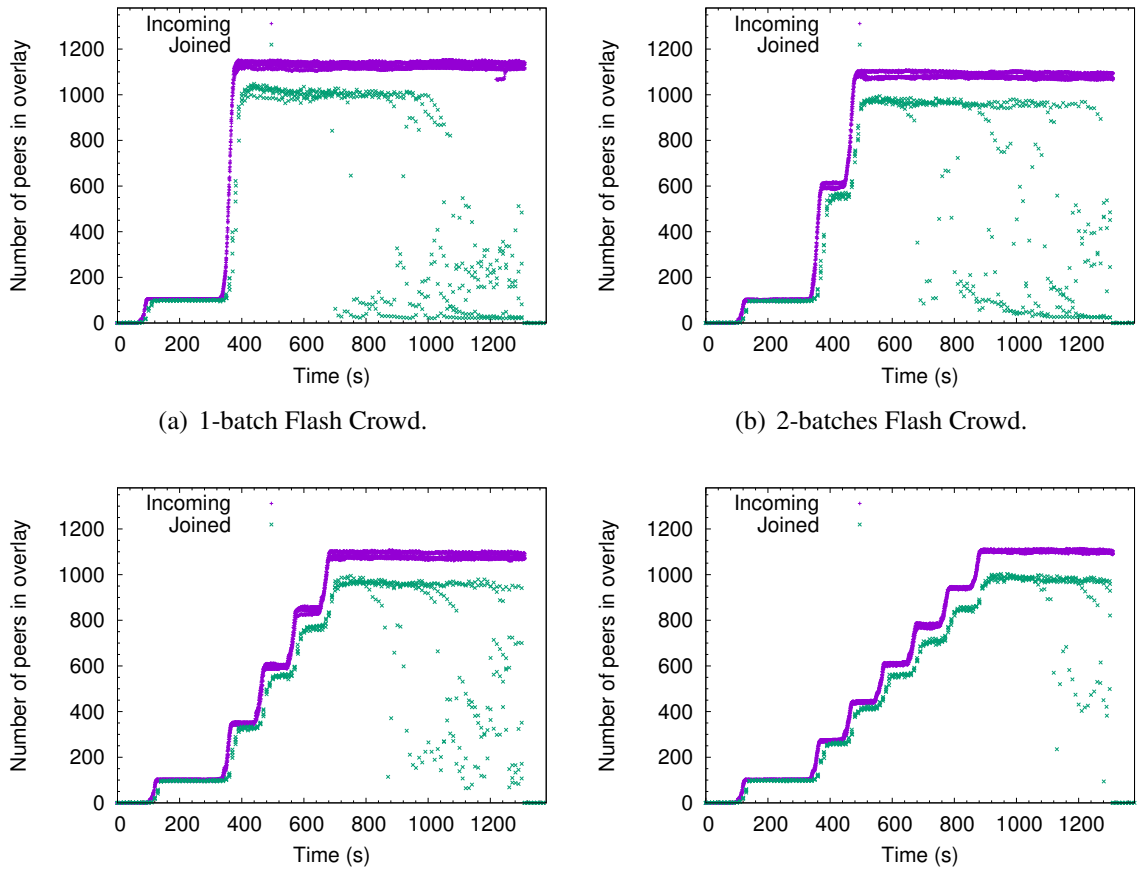
To evaluate the behavior of our network configurations we implemented and executed a flash crowd technique approach presented in [Liu et al. 2012]. Based on peer's joining by batches, this technique holds newcomers peers in a larger group $\mathcal{P}$ to control the joining process. Basically, in each i-iteration, the system evaluates the network's resources and establishes both $\mathcal{R}_i \subseteq \mathcal{P}$ and τ_i. It means that each peer $p \in \mathcal{R}_i$ is allowed to join the network and the next $(i+1)$-iterator happens after network stabilization time τ_i. Several iterations are repeated until $\mathcal{P} = \emptyset$.

To simplify understanding our network resources, we fixed $\tau_i = 100s$ and we experimented [1,2,4,6] as i-iterations limits. It is enough to take a preliminary network behavior without spending a long time implementing the common flash crowd technique in its essence since it is not the scope of our work.

Figure 1 shows the joining results charts for the common flash crowd technique. The instability in the joined amount of peers is shown in all charts on Figure 1. This demonstrates the negative effect of the flash crowd event on peer joining. Thus, we can see that no configuration had provided the desired joining amount. However, we consider the relative improvement of each iteration enough to study this hard network configuration scenario.

Once Figure 1(d) approaches the desired network joining rate, we chose this configuration to show the values of discontinuity and latency metrics for chunks in Figure 2. We observe in both Figure 2(a) and Figure 2(b) that until 1000 seconds, the network was stable. After it, since joining was not sustained, the network service was compromised and has reached high discontinuity and latency.

Our expectation is to reach at least the *6-batch* joining rate shown in Figure 1(d), but with the *1-batch* incoming peer behavior from Figure 1(a). For it, we dispose that parallel networks technique that holds newcomers in isolated correlated networks. We believe that is possible to join peers keeping distance from master network and, thus, ensuring existing peers relationships. Moreover, we expect to successfully increase peer's joining rate, by limiting the relationship amount among free riders and cooperative peers.



(a) 1-batch Flash Crowd.

(b) 2-batches Flash Crowd.

(c) 4-batches Flash Crowd.

(d) 6-batches Flash Crowd.

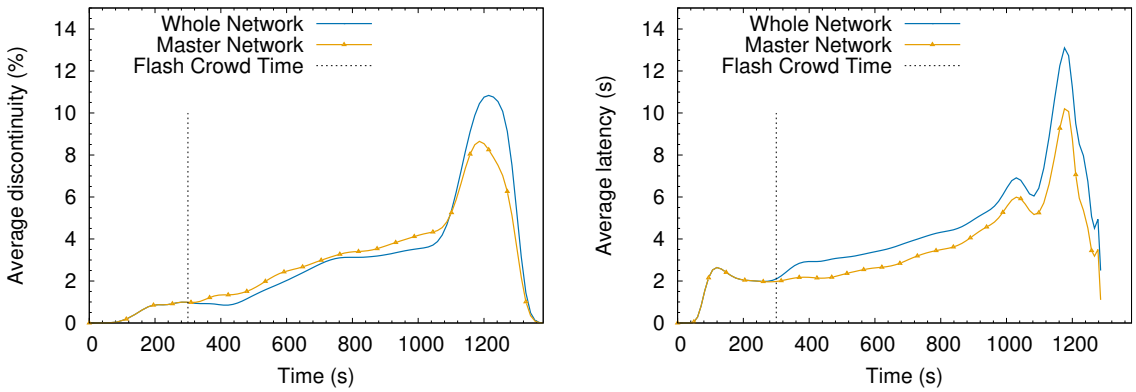
Figure 1. Common Technique Overlay Size.

5.2. Parallel Networks Technique

Parallel networks technique, introduced in [Miguel et al. 2016], are useful to construct new networks from an existent network. Let M be the *master network*, as in Section 4, the constructed P2P mesh overlay around media server S. In this technique, the bootstrap server B select a set of peers $S_{aux} \subset M$. Each $p \in S_{aux}$ becomes a especial *auxiliary server* peer. The master network ensures media transmission for auxiliary servers, but auxiliary servers are unable to give chunks for peers in the master network. This stage is called *server isolation*.

The server isolation prepares network for the flash crowd event. When the peers' herd appears, peers are distributed around each auxiliary server, establishing the parallel networks. New relationships are established only among peers in the same parallel network and their auxiliary server. It provides a first relationship limit controlled by bootstrap server. The last step is *parallel network merging*. We permit, in this step, that peers from parallel merging networks to establish new relationships among cooperatives peers from the master network. As a second limit, during merging phase we kept free riders around old partners. Thus, the merging state of parallel networks preserves the master network to sustain all the joined peers.

Figure 3 presents the stages until parallel networks formation. In the first stage, the



(a) 6-batches Playback Quality. (b) 6-batches Distribution Latency.

Figure 2. 6-batches Common Technique Metrics.

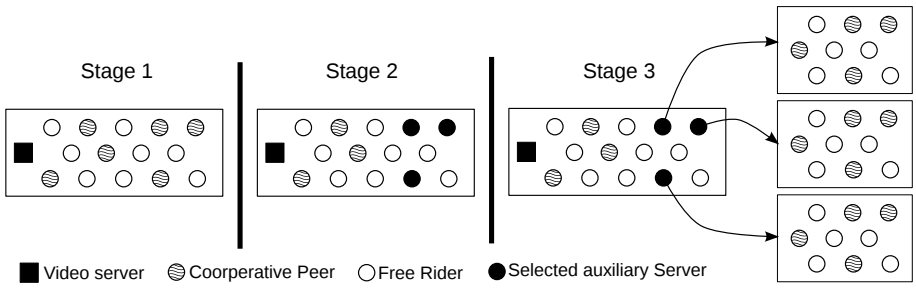


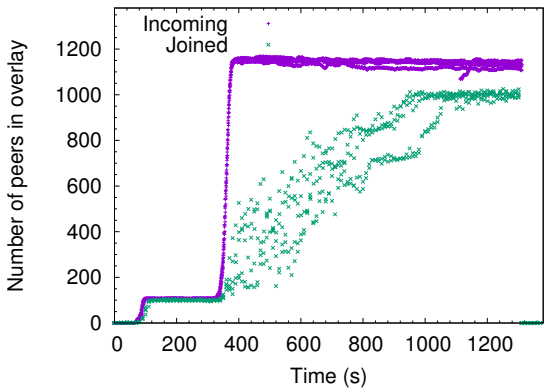
Figure 3. Parallel Networks Formation Flow.

master network is stable with the video server serving content to the peers, as described earlier in this section; in the second stage there is a flash crowd surge and the auxiliary servers are selected to attend the demand; finally, on the third stage, there is the formation of the parallel networks where selected auxiliary servers behaves as the video server for each newly formed network.

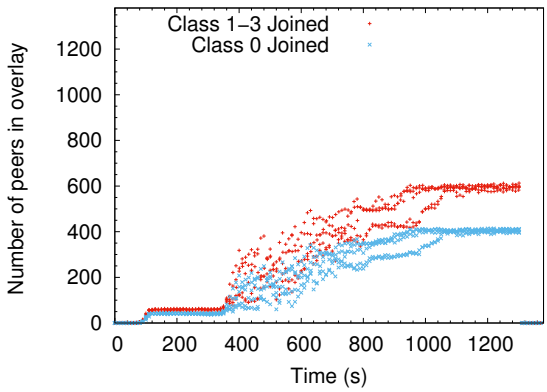
Six auxiliary server from peer *class 3* were configured in our experiments. The newcomers peers were split for constructing balanced parallel networks. After the flash crowd, we wait 200 seconds to construct the parallel network topology and then, to begin merging each of these networks in steps happening in 100 seconds interval. So, from the beginning of the experiment, since there are six parallel networks and the flash crowd event happens at 350 seconds, the first network merges at 550 seconds, the second at 650 seconds and so on.

As result, in Figure 4(a) the incoming curve is equivalent to the *1-batch* from Figure 1(a) and we have reached the same joining behavior of the *6-batches* common flash crowd technique, as shown in the joined curve in Figure 1(d). Figure 4(b) shows separated joined common peers and joined free rider. We observe balance in joining rate for all peer classes. The small joining gap between class 1-3 and class 0 is explained because free riders are 40% versus 60% from common peers.

Figure 5 shows parallel networks' discontinuity and latency. It is clear that the master network was stable during the whole experiment's steps. This fact is important



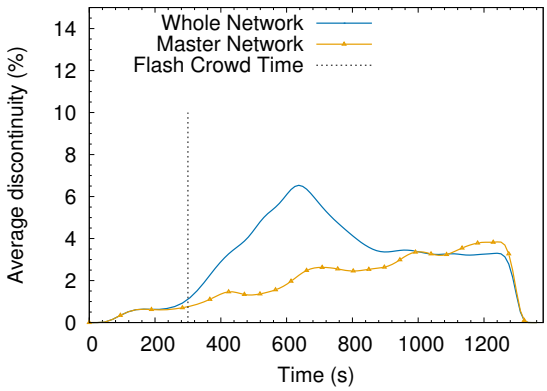
(a) Overlay Size.



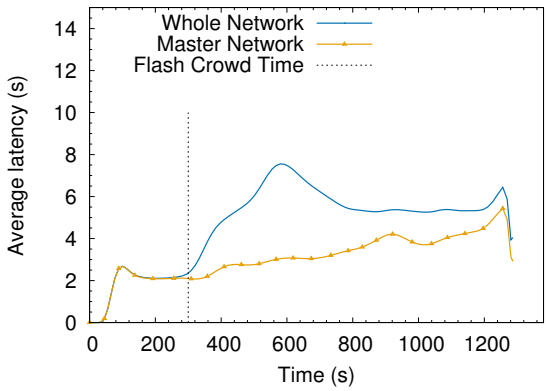
(b) Overlay Size of Free Rider and Common Peers.

Figure 4. Parallel Networks Technique Overlay Size.

and we suspect that it explains why parallel network technique supported large number of incoming peers without postponing the joining event. In the case of the whole network, chunk latency and discontinuity peaks at around 600 seconds as expected. The poor auxiliary server upload bandwidth is a serious parallel networks constraint. Compared with non bandwidth limited server S in the master network, parallel networks received twice more incoming peers. Even so, the peaks were controlled and they stabilized during the merging steps.



(a) Playback Quality.



(b) Distribution Latency.

Figure 5. Parallel Networks Metrics.

Parallel networks technique is a novel technique. Thus, we suspected that these systems work better with high bandwidth auxiliary servers. However, isolating high bandwidth peers can disrupt promising topological partnerships which may compromise the master network. Besides that, it is required a deep understanding of the stabilization time, merging phases and amount of parallel networks before they can be applied commercially. Furthermore, we issue that auxiliary server clusters and the newcomers amount in each parallel network is a challenge. Finally, we consider it was challenging to implement this technique.

Even with the lack of knowledge about these parallel networks techniques, free

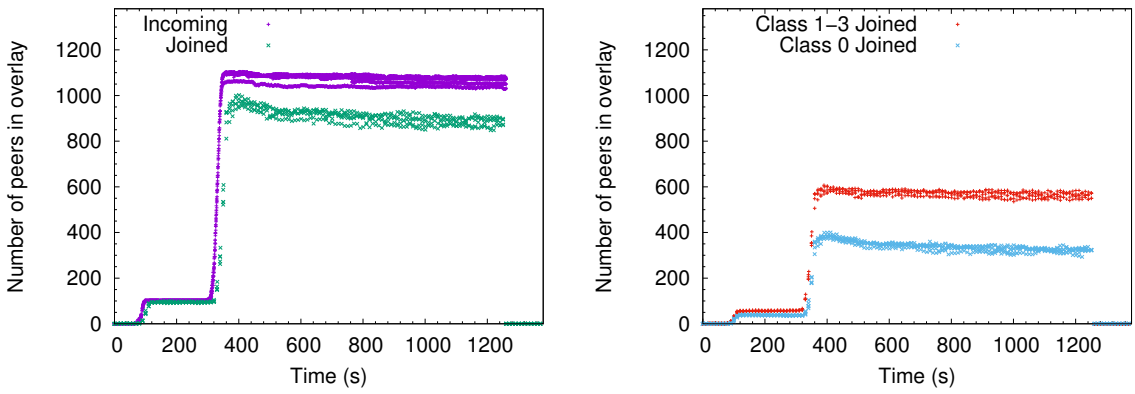
rider relationship limitation was successfully investigated in our experiments. We want, from now on, to make new topology configurations in order to explore free riders' partnership limitations in a realistic scenario. If this experiments become successful, new topology organization may contemplate this solution without service disruption and should be robust for the flash crowd events.

5.3. Free Rider Slice Setup

Based on results from parallel networks in Section 5.2, we have realized that splitting free riders from cooperative peers delivers good network performance. Furthermore, parallel networks technique requires complex implementation and is insufficiently researched yet. Thus, our goal is to decrease competition among free rider and cooperative peers imposing a simple partnership constraint on peers. Therefore, by considering peer's classification, we can define which partnerships are allowed and which are denied among peers.

Then, the free rider slice is defined following the partnership constraint: peers from class 3, that have high upload bandwidth, are allowed to accept out-partners requests only from cooperative peers, that is, peers classified on class 1-3, (i.e. classes 1, 2 and 3). On the other hand, low upload bandwidth peers from class 1 accept out-partners requests from class 0 peers (i.e. free riders). Finally, to balance the network relationship distribution, no rule was set up for peers from class 2.

As with parallel networks, we evaluate experiments considering *1-batch* incoming peers. Figure 6(a) shows joined peers amount versus incoming peers on network configured with free rider slice. We observe fast joining effect preserved until the experiment ends. Joined free riders and common peers are shown in Figure 6(b). In this case, we consider that all peers (from all classes) joined without preference nor privilege. Once server S do not accept free riders and we reach full class 0 joining, we understand that our partnership constraint rules direct new relationships among peers pushing free riders to the edge of the network topology. We believe that overlay topology is desired, once free rider and low upload peers amount from class 1 are sufficient to disrupt cooperative peers media propagation, as occurred in Section 5.1.

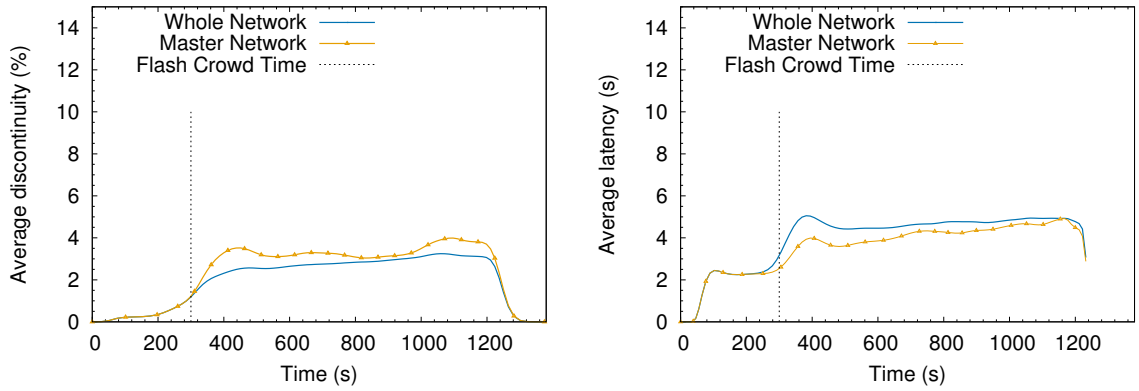


(a) Overlay Size. (b) Overlay Size of Free Rider and Common Peers.

Figure 6. Free Rider Slice Technique Overlay Size.

New topology organization improvements are observed on chunk latency and chunk discontinuity metrics, Figure 7. Both metrics stayed stable with low values for

both master and whole networks. In this case, we reach the same values of metrics observed on master network for parallel network technique, Figure 5. However, we adjust the high values of whole network's metrics found with parallel network experiments with free rider slice without large parallel network complexity. We also observe that free rider slice provides a average discontinuity below 4% throughout the experiment, which is a fair value for discontinuity, as considered by [Traverso et al. 2012].



(a) Playback Quality.

(b) Distribution Latency.

Figure 7. Free Rider Slice Metrics.

We consider that free rider slices needs more studies before being used in commercial applications. It is because the relationship among network set up parameters and its consequences are not well understood to allow dynamic classes configuration yet. Among these parameters we can mention: (i) peers' chunk distribution; (ii) peers' out-degree; and (iii) peers' upload bandwidth. However, we believe that to aggregate the concepts of free rider slice to topology organization is not complicated. This concept can inspire new future works.

6. Conclusion

Currently, there are no detailed studies about free rider relationship isolation done yet. In this work, we expose constraint for free rider neighborhood to assess whether it is a relevant issue to improve overlay organization advances. First, based on *parallel networks*, we show that free rider isolation allows growing the number of simultaneous peers joining without crashing the network transmission. Thus, we propose a new setup, called *free rider slice*, for networks' parameters set up that results a better peer joining without the *parallel networks'* complexity.

Despite other works that only identify free riders and converge the topology to push them to the mesh's edges, in this work we show that splitting the free riders and the cooperative peers' relationships may avoid media resources competition. Thus, this resource preservation contributes to the network media transmission stability. Furthermore, assigning a limited neighborhood size for each peer can become a parameter for new topology overlay techniques to avoid the negative consequences arising from non limited size.

We believe that many future works may be developed from our observations. Our experiments facing flash crowd events are a clue that simple network settings can pro-

mote major improvements in P2P live media distribution. Then, we want to study the dynamic network mechanism in order to facilitate the free rider isolation to provide a robust network topology organization.

Acknowledgment

This work was partially funded by FAPEMIG, CAPES, and CNPq.

References

Adar, E. and Huberman, B. (2000). Free Riding on Gnutella. *First Monday*, 5(10-2).

Chen, Y., Zhang, B., and Chen, C. (2011). Modeling and Performance Analysis of P2P Live Streaming Systems under Flash Crowds. In *ICC'11*, pages 1–5.

d. Silva, A. P. C., Leonardi, E., Mellia, M., and Meo, M. (2008). A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems. In *2008 Eighth International Conference on Peer-to-Peer Computing*, pages 279–288.

e Oliveira, J. F., Cunha, Í., Miguel, E. C., Rocha, M. V., Vieira, A. B., and Campos, S. V. (2013). Can Peer-to-Peer Live Streaming Systems Coexist With Free Riders? In *IEEE P2P 2013 Proceedings*, pages 1–5. IEEE.

Fortuna, R., Leonardi, E., Mellia, M., Meo, M., and Traverso, S. (2010). QoE in Pull Based P2P-TV Systems: Overlay Topology Design Tradeoffs. In *P2P '10: IEEE International Conference Peer-to-Peer Computing*, pages 1–10, Washington. IEEE Computer Society.

Karakaya, M., Korpeoglu, I., and Ulusoy, O. (2009). Free Riding in Peer-to-Peer Networks. *IEEE Internet Computing*, 13(2):92–98.

Krishnan, R., Smith, M., Tang, Z., and Telang, R. (2004). The Impact of Free-Riding on Peer-to-Peer Networks. In *Annual Hawaii International Conference on System Sciences*.

Li, B., Keung, G., Xie, S., Liu, F., Sun, Y., and Yin, H. (2008). An Empirical Study of Flash Crowd Dynamics in a P2P-Based Live Video Streaming System. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5.

Liu, F., Li, B., Zhong, L., Li, B., Jin, H., and Liao, X. (2012). Flash Crowd in P2P Live Streaming Systems: Fundamental Characteristics and Design Implications. *Parallel and Distributed Systems, IEEE Transactions on*, 23(7):1227–1239.

Liu, F., Li, B., Zhong, L., Li, B., and Niu, D. (2009). How P2P Live Streaming Systems Scale over Time Under a Flash Crowd? In *Proceedings of the 8th International Conference on Peer-to-peer Systems*, IPTPS'09, pages 5–5, Berkeley, CA, USA. USENIX Association.

Lobb, R. J., Couto da Silva, A. P., Leonardi, E., Mellia, M., and Meo, M. (2009). Adaptive Overlay Topology for Mesh-based P2P-TV Systems. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '09, pages 31–36, New York, NY, USA. ACM.

Meulpolder, M., Meester, L., and Epema, D. (2012). The Problem of Upload Competition in Peer-to-Peer Systems With Incentive Mechanisms. *Concurrency and Computation: Practice and Experience*, 25(7):899–917.

Miguel, E., Cunha, I., and Campos, S. (2016). Ingressos em Redes P2P para Vídeo ao Vivo. In *SBRC 2016 - WP2P+ ()*.

Moltchanov, D. (2011). Service Quality in P2P Streaming Systems. *Computer Science Review*, 5(4):319–340.

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

Payberah, A. H., Dowling, J., and Haridi, S. (2011). GLive: The Gradient Overlay as a Market Maker for Mesh-Based P2P Live Streaming. In *Parallel and Distributed Computing (ISPDC), 2011 10th International Symposium on*, pages 153–162.

PlanetLab (2009). An Open Platform for Developing, Deploying, and Accessing Planetary-Scale Services. `http://www.planet-lab.org/`.

PPLive (2008). `http://www.pplive.com`.

Rückert, J., Richerzhagen, B., Lidanski, E., Steinmetz, R., and Hausheer, D. (2015). TOPT: Supporting Flash Frowd Events in Hybrid Overlay-Based Live Streaming. In *2015 IFIP Networking Conference (IFIP Networking)*, pages 1–9.

SopCast (2008). Deliver Your Media to the World! `http://www.sopcast.com`.

Traverso, S., Abeni, L., Birke, R., Kiraly, C., Leonardi, E., Cigno, R. L., and Mellia, M. (2012). Experimental comparison of neighborhood filtering strategies in unstructured P2P-TV systems. In *2012 IEEE 12th International Conference on Peer-to-Peer Computing (P2P)*, pages 13–24.

Traverso, S., Abeni, L., Birke, R., Kiraly, C., Leonardi, E., Lo Cigno, R., and Mellia, M. (2015). Neighborhood Filtering Strategies for Overlay Construction in P2P-TV Systems: Design and Experimental Comparison. *IEEE/ACM Transactions on Networking (TON)*, 23(3):741–754.

TVU (2013). TVU VR. `http://www.pplive.com`.

UUSee (2008). UUSee Inc. `http://www.uusee.com/`.

Wichtlhuber, M., Richerzhagen, B., Rückert, J., and Hausheer, D. (2014). TRANSIT: Supporting Transitions in Peer-to-Peer Live Video Streaming. In *2014 IFIP Networking Conference*, pages 1–9.

Wu, H., Liu, J., Jiang, H., Sun, Y., Li, J., and Li, Z. (2012). Bandwidth-Aware Peer Selection for P2P Live Streaming Systems under Flash Crowds. In *Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International*, pages 360–367.

Zheng, Q., Long, Y., Qin, T., and Yang, L. (2011). Lifetime Characteristics Measurement of a P2P Streaming System: Focusing on Snapshots of the Overlay. In *Intelligent Control and Automation (WCICA), 2011 9th World Congress on*, pages 805–810.