BemTV: Hybrid CDN/Peer-to-Peer Architecture for Live Video Distribution over the Internet

Flávio Ribeiro Nogueira Barbosa
and Guido Lemos de Souza Filho

CITE THIS ARTICLE
Barbosa, Flávio Ribeiro Nogueira; de Souza Filho, Guido Lemos; 2020. BemTV: Hybrid CDN/Peer-to-Peer Architecture for Live Video Distribution over the Internet. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2020.6. Web Link: http://dx.doi.org/10.18580/setijbe.2020.6

BemTV: Hybrid CDN/Peer-to-Peer Architecture for Live Video Distribution over the Internet

Flávio Ribeiro Nogueira Barbosa, Guido Lemos de Souza Filho

Abstract — **Assuming that video streaming is now responsible for the absolute majority of the Internet traffic and considering that the audience uses WebRTC-enabled web browsers and mobile devices to access and retrieve content, this work proposes the development of a peer-to-peer overlay network to assist the delivery of video streaming events that use HTTP-based protocols without the need to install additional software. Using the peer-to-peer network, the client/server model becomes hybrid, where network nodes that are watching the same event can retrieve portions of the video content directly from the server or neighboring nodes. This approach has two main objectives; decrease the client/server traffic and consequently the economic cost of delivery while improving the quality of the users' experience, given that communication between neighboring nodes can support the flow of better quality videos between the points**

Index Terms — **peer-to-peer, webrtc, http live streaming, HLS.**

I. Introduction

The growth of Internet use, followed by the increase of the preference for multimedia over text-based content consumption by the users in major portals, social networks, and video streaming services, supports the fact that the distribution of videos is now responsible for the majority of the data traffic on the Internet. In some countries, such as the United States, video consumption already exceeds 60% of all traffic on the network, and it is estimated that in 2022, 82% of all IP traffic in the world will be dedicated to the transmission of video content, 15 times higher than the video traffic in 2017.

The latest live events broadcast on the Internet have beat audience records, and it is expected that this phenomenon will continue to happen. For example, CBS' NFL Super Bowl 53 internet streaming for home users in the United States in 2019 reached 7.5 million unique devices. The users consumed more than 560 million minutes of video, increasing more than 20% when compared to the previous year, as shown in Fig 1. As another example, the FIFA Football World Cup in 2018 also set a new audience record for online streaming in Brazil. It peaked at more than 1.2 million simultaneous users, presented in Figure 2.

The high demand for bandwidth, quality of service (QoS), and high traffic inherent in video distribution applications have brought significant challenges to today's Internet. The implementation of large-scale infrastructures, the cost of transmission, and the QoS became a central issue for the industry. It is also observed that, as video consumption increases, rebuffering problems increase and users' session time decreases.

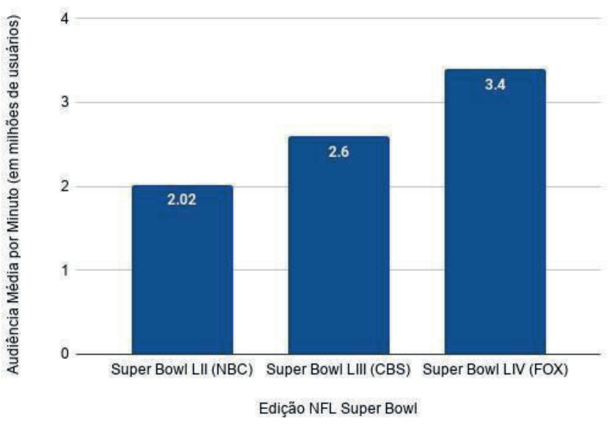


Fig 1: Average audience growth per minute between editions of the NFL Super Bowl.

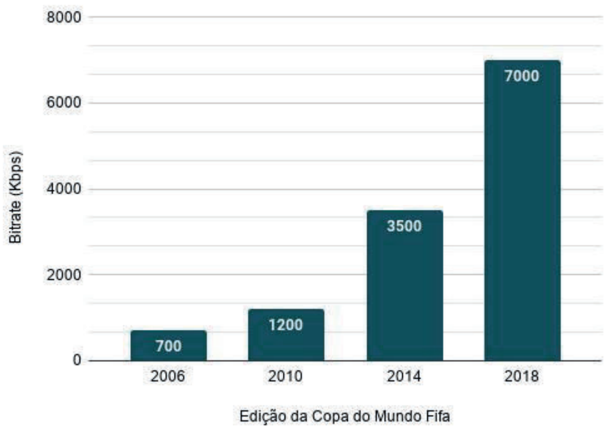


Fig. 2: Quality growth (bitrate) between editions of the World Cup, Grupo Globo.

Typically, large companies that stream video to hundreds, thousands, or even millions of people, use one or multiple *Content Delivery Networks* (*CDN*) to meet public demand. The main objective of *CDN* 's is to distribute content through sets of agglomerated servers called points of presence or *PoP*'s that are usually distributed geographically to guarantee a reliable, scalable, and efficient delivery to end-users [1]. This approach has some disadvantages:

- Scalability. CDN's use PoP's for delivery and load

F. R. N. Barbosa - Video Technology Group, CBS Interactive , New York, NY - United States (e-mail: flavio.ribeiro@cbsinteractive.com).

G. L. Souza Filho - LAVID, Universidade Federal da Paraíba (João Pessoa, PB - Brazil (e-mail: guido@lalavid.ufpb.br).

balancing. When a user is geographically far from the PoP, she depends on the Internet's communication channels and exchange points (PTT) to retrieve the requested content. In countries with poor telecommunications networks and slow exchange points, there is a need for an increase in the deployment of PoP's. In Brazil, the most popular CDN's concentrate their presence in the South and Southeast, damaging the experience of video consumers in the Midwest, North, and Northeast.

- Cost. The cost of building and managing a CDN is very high, both economically and in terms of complexity [12]. In addition to the need for specific hardware infrastructure, the point of presence needs to operate without interruption. According to [19], Google Youtube is believed to spend an average of $ 1 million a day on distribution infrastructure costs.
- Quality of Experience (QoE). The last significant event transmissions based on the traditional client/server model, using only CDN's, presented major scalability problems, making the audience's consumption experience unfeasible or degrading [27] [14]. In the context of Brazil, as shown in Fig. 3, in May 2013, users connected from the telecommunications provider Embratel in Caruaru - a city located in the Northeast of the country obtained an average of 9 2 buffer underflow events during the transmission of the semifinal of the European Champions League [5].

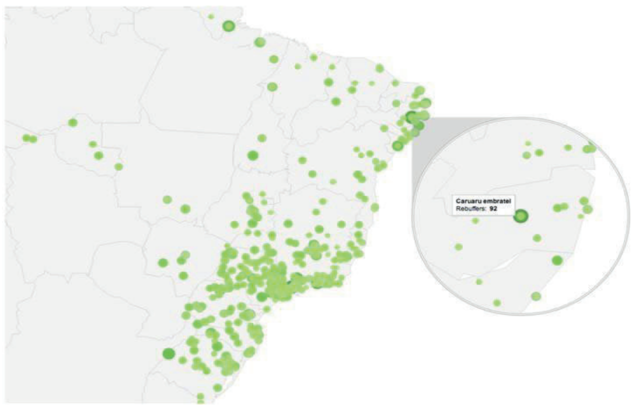


Fig. 3: User with 92 *rebuffers* events when watching video on Globo.com using Embratel provider in Caruaru, Pernambuco.

Given the considerably intensive requirements and resources imposed by streaming services in addition to the points already discussed, it can be said that traditional client/server systems do not scale sufficiently [25]. It is known that peer-to-peer architectures such as CoolStreaming, GridMedia, PPTV, PPLive, PPStream, UUSee, and TVAnts present themselves as alternatives to the problem of large-scale transmissions [8]. However, with the development of protocols for streaming video on the Internet, and migration in recent years of large companies (including Microsoft, Adobe and Apple) protocols *stateless* based on HTTP (English *Hypertext Transfer* Protocol), many state-of-the-art have become obsolete [16]. Since then, *peer-to-peer* models supporting HTTP delivery have been developed.

Assuming that the absolute majority of the audience consumes the videos in web browsers or mobile devices capable of operating and supporting traffic via WebRTC (*Web Real-Time Communications*), this work proposes the development of a *peer-to-peer* overlay network to assist delivery of video content in live streams that use HTTP-based protocols without the need to install additional software. Using the *peer-to-peer network*, the client/server model becomes a hybrid model, where network nodes that are watching the same transmission can retrieve portions of the video content directly from the server or neighboring nodes. This approach has two main objectives; decrease the client/server traffic and consequently the economic cost of delivery while improving the quality of the users' experience, given that communication between neighboring nodes can favor the flow of better quality videos between the points.

Thus, it is expected that in addition to the traffic between users and the CDN decrease, the system will become more consistent and fault-tolerant, increasing the throughput of the network, improving the user experience, and decreasing the cost of transmission as a whole.

II. THEORETICAL FRAMEWORK

Peer-to-peer meshes for solving scalability problems are widely studied and used in several segments. In the context of streaming services, the transmission of video streams where there is a high need for resource availability, the traditional client / server-based system proves to be insufficient for the growing demand [27] [14]. An alternative explored by the research and industry fields in recent years focuses on using users' upload bandwidth to exchange video segments through a peer-to-peer strategy in conjunction with the traditional client/server approach. According to [3], sharing data with the help of a peer-to-peer mesh can decrease traffic between users and servers by up to 96%, promoting a drastic decrease in transmission costs and reducing bottlenecks, resulting in improving the experience of broadcast users.

It is important to note that, for the exchange of video segments to occur, a fundamental requirement for users participating in the peer-to-peer system is the capacity to store portions of the watched content, to disseminate them to the other participants.

CoolStreaming [26] is one of the first streaming systems of large-scale video using P2P technology. The system was developed using the Python programming language and applies a gossip protocol [24] for the transmission of buffer maps between nodes. A buffer map reflects the portions of the video that are stored on a given node [6]. When starting the player, it connects to an origin node to receive a set of nodes available for that transmission. Upon receiving the set, the player randomly draws a subset of nodes and promotes them to partners. Once the partners are defined, the receiver receives their buffer maps and requests the first video clips from this subset. Since the choice of partners is random, the probability that the chosen nodes are not optimal for the recipient in question is high. Strategies such as the calculation of the RTT (Round- trip time), elapsed time for sending a message and arrival of the acknowledgment that the destination received it correctly in the handshake, the establishment of the initial connection between nodes or information about the location of the system components in the origin node could be implemented to optimize the choice of this subset.

PPlive [22] is one of P2P video streaming systems unstructured most popular today, especially in China, which serves about 3 million daily users in 300 different transmissions at a rate ranging from 250Kbps and 400kbps [6]. The system is non-structured (mesh), and the content exchange mechanism between nodes of the overlay network (overlay network) is done by breaking the streaming video into small pieces (chunks) [23]. Although widely used, PPLive has some shortcomings. The startup delay (the time between the press play until the first video frame) is in the order of 20 seconds for popular broadcasts and reaches up to a few minutes for less popular broadcasts. Another known limitation is the delay ratio between the nodes of the PPLive mesh, which reaches 150 seconds [7], degrading the experience in cases of live transmissions, such as a football game, for example. The protocol used in PPLive is proprietary, making it difficult to evolve. Still, [Li, Chen 2008] proposed a segment recovery strategy called Threshold Bipolar (TB). To conduct the study, it was necessary to reverse engineer the protocol without concrete conclusions about its improvement.

It is known that since the appearance of CoolStreaming and PPLive, several other architectures, models and peer-to-peer systems have presented themselves as alternatives to the problem of large-scale transmissions. Among the published models, we can highlight:

- P2PSP [13] uses a structured model where the nodes are classified into three layers: Broadcasting, Integrity, and MultiChannel. Broadcasting layer nodes are responsible for segmenting and adapting the video stream before being propagated over the network. The users that make up the Integrity layer manage the input and behavior of the nodes, detecting poisoned and hostile peers. Users who belong to the MultiChannel usually t is m grid conditions and resources to consume more than one stream video, serving as a relay for more than one transmission.
- mTreeBone and CorePeer [4] [Want 2007] are hybrid systems that utilize an array of unstructured networks (mesh) and structured (trees). Both classify the component nodes of the network as stable or not based on different criteria. mTreeBone considers the time assisted in distinguishing and promoting as the CorePeer analyzes the width of upload bandwidth thereof to promote them. A node, when promoted to stable, connects directly to CDN functioning as an extension of it, serving the other nodes of the mesh.
- BiToS [21] uses the sharing service network architecture files peer-to-peer best known globally, BitTorrent, to assist the flow of video segments between users. BiToS changes the order in which *chunks* are to be trafficked. Common BitTorrent uses a rarest-*first* strategy (where less popular chunks are prioritized, seeking a balance in their availability). In contrast, BiToS prioritizes chunks in an order similar to the progressive download, where the most critical segments to be sent and received are the next to be played in the video player.
- OpenSatRelaying [20] is a hybrid model where geographically distributed users can receive and repeat a broadcast signal via satellite and serve as entry points for regional IP transmission. Users who receive the IP stream also act as a relay for others. The main challenge of this approach is the synchronism between users since the transmission via satellite promotes a low delay of sending and receiving, unlike IP transmission.

With the evolution of protocols for streaming video on the Internet and the migration in the last years of large companies (including Microsoft, Adobe, and Apple) to stateless protocols based on HTTP, lots of the progress on the state of the art became obsolete [17].

A. Hybrid Peer-to-Peer Systems for HTTP-based Transmissions

SmoothCache [18] system is considered the first peer-to-peer built-in support for content transmissions on-demand and live via the HTTP protocol and model your solution using the concepts caching distributed.

In the first approach of the system, classified as Baseline Caching, each node maintains a table with a subset of the network's nodes and which video clips each has. Thus, when requesting a segment that does not exist in the distributed cache, the player directs its request directly to the server at the CDN and, when retrieving the requested segment, reports to a large number of other nodes so that they update their table using the method of gossip. The construction of the connections between the nodes and their maintenance is done at random, in an unstructured scheme (mesh). It is important to note that, as previously mentioned, the scheme takes advantage of the lack of synchronism between users inherent in the implementation of the players in this protocol - making the exchange of segments between them possible.

The Improved caching the SmoothCache adds an array of features. First, in addition to the natural desynchronization between users, the model positions users with greater direct bandwidth with CDN ahead of the transmission - limiting this positioning to up to 4 seconds ahead of other users. Each node that can connect to another node is defined as a neighbor (neighbor), and from the set of neighbors are randomly chosen node partners (partner), a subset of neighbors. Each chosen partner receives an initial score composed of the amount of data transferred successfully so far, and the current quality of the video streaming consumed in the case gave an adaptive streaming scenario. It is important to note that favoring nodes that are in the same quality as the user is dangerous: Once the node changes quality, it will be isolated. Therefore, the version Improved Cached the SmoothCache keeps its set of partners, we are in all the qualities available in a Gaussian distribution centered on the current quality of the given node.

In a multi-bitrate stream, the user's player maintains a periodic calculation of the band with the CDN - by measuring the time required to download a segment compared to its size in seconds and its quality in bits per pixel (bpp) to define which correct quality of the stream to be consumed at that time. Since SmoothCache shares segments between nodes, if the desired segment is already in the local cache, the implementation adds a delay in delivery to prevent the calculation from resulting in a false positive for the quality of the band at the moment, maintaining the download time of the segment approximately equal to the time it would take if it were recovered from CDN.

The implementaçãoo the SmoothCache uses the standard Smooth -Streaming Microsoft and *player headless* that runs natively in the operating system.

The pDASH [10] is a P2P overlay proposal to assist in the delivery of video clips on-demand using the DASH (Dynamic Adaptive Streaming over HTTP) standard. Similar to other HTTP standards and drafts, a DASH video stream is segmented on the server and distributed among users. To coordinate the order of the segments, the server builds a playlist - known as the Media Presentation Description (MPD) in the DASH standard - that will be accessed by the player before requesting the segments. In pDASH, this server that builds and serves the MPD also monitors accesses and records the addresses of users who have successfully downloaded a segment. When successfully serving a stretch, the server adds the user's address to the MPD, allowing users who request the MPD after this modification to choose which source they want to retrieve that stretch from. It is important to note that this addition is allowed by the standard. For pDASH to work, clients need to run an HTTP server locally in addition to having a public IP or NAT route for external access from this server. The implementation of simulation of pDASH used parts of HTTPTools Framework [9] to the HTTP server, and parts of the DASH VLC Plugin [2] to the native video playback.

[15] presents a P2P network using APIs from WebRTC. The publication does not clarify whether the transmission is made using any HTTP protocol, but describes how the network is built. The system uses a middleware called PeerJS that abstracts the direct calls to the native API's WebRTC and maintains compatibility if the method signatures are changed. The system also uses a server to coordinate and connect incoming users. Thus, when rendering the video player, the peer generates a unique user ID and registers with the WebRTC coordinator. The coordinator, in turn, selects two other users from the same transmission and promotes the connection between them. When closing the connection, the nodes use the RTCDataChannel API to exchange transmission data. Since the coordinator randomly selects the peers, an unstructured mesh is created and the users who are in the transmission the longest are selected more often and, therefore, are more stable and maintain more connections.

III. PROPOSED ARCHITECTURE

Based on the academic contributions and technological developments on the Internet already mentioned, and given the market scenario, where the adoption and migration to HTTP-based protocols are being made and based on the successful migration of the RTMP protocol to HTTP Live Streaming (HLS) by Globo.com, which resulted in 33% more time watched, we believe that the future of live broadcasts on the Internet will be using stateless protocols.

Analyzing the state of the art hybrid models CDN/Peer-to-Peer, we believe that non-structured models are more effective in dealing with highly dynamic networks and an alt turnover rate (node churn), an inherent feature of online live broadcasts large-scale. With the support for peer-to-peer communication added by WebRTC, we believe it is possible to implement a model entirely within the browser, without the need to install clients or plug-ins on devices, using only JavaScript technologies, already supported in most browsers used on the Internet. Therefore, we propose a hybrid architecture model CDN/Peer-to-Peer unstructured (mesh) for live video streams using only the technologies available in modern browsers, including WebRTC where we consumers help to CDN delivery of video segments that use the HTTP Live Streaming (HLS) protocol.

To validate our research work, we decided to create two architectures. One, initial, for proof of concept, and a second where we improve the protocol with the election of partners.

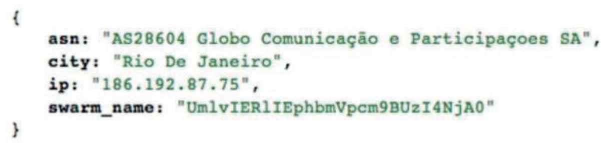
The goal of the development of this proof of concept was to validate the technologies chosen for the evolution of a peer-to-peer overlay network where users who consume the video cannot have their consumption experience impaired. In this way, the work focused on a hybrid model adding what already exists in current infrastructures, enabling the evolution of a distribution platform with the adoption of the described system without offering impact or downtime on the infrastructure.

A. BemTV Baseline

1) Creation of the Peer-to-Peer Mesh

For nodes to be able to establish a connection, it is necessary to a stage where the nodes become aware of each other. For BemTV Baseline, it was defined that only nodes from the same geographic region and Internet provider would meet each other. This set of nodes was given the name of a location-aware swarm (geo-aware swarm).

The node then makes an HTTP GET request to a server that will identify you based on your city and provider and will return a swarm name as shown in Fig. 4.



```
{
    asn: "AS28604 Globo Comunicação e Participaçoes SA",
    city: "Rio De Janeiro",
    ip: "186.192.87.75",
    swarm_name: "UmlvIERlIEphbmVpcm9BUzI4NjA0"
}
```

Fig. 4: HTTP Response by swarm name.

To identify the ISP and city, the request's IP is used as the input for libraries GeoIPASNum and MaxMind's GeoLiteCity. The server consists of a Python application using the Flask framework and is served by an Nginx server.

In possession of the swarm name, the node uses the RTCPeerConnection API to publish. An rtc-switchboard server receives the publication and initiates the attempt to switch via STUN (Session Traversal Utilities for NAT) to find a route through NAT's (Network Address Translation Port Mapping Protocol) where the two nodes can exchange data. It is important to note that, with each new node in the swarm, a connection to each node of the swarm will be stabilized, increasing the number of commutations exponentially, that is, considering n as the number of nodes the number of connections is equal to n x (n - 1).

2) *Requesting Video Segments*

Once the nodes already have connections with the others in the same cluster, the video player starts playback by requesting the playlist for CDN. The direct request from the player to the CDN, in this case, is not intercepted, since we want to keep the nodes updated with the live stretches, reducing the playback delay for the live event.

After receiving and interpreting the data (*parsing*) of the playlist, the player initializes the request to the chunks. Instead of making HTTP requests directly to the CDN, the node sends a DESIRE message to all nodes in the cluster, and each receiving node searches its local cache for the requested video segment. The implementation stores the last 10 stretches of video watched, and if the chunk is contained in the cache, the node sends back a DESACK message.

The node that wants the segment receives all DESACK's and analyzes which is the most appropriate to receive that segment. In the current implementation, the node that sends the first DESACK is chosen. Considering the node that wants the chunk to be Peer A, and the node that sent the first DESACK being Peer B, the negotiation process follows as shown in Fig. 5. A REQ message is sent from Peer A to Peer B, which in turn sends an OFFER message with the content of the chunk.

When Peer A sends the DESIRE message to the cluster, it waits 0.7 seconds until some other node sends the DESACK. If it does not, Peer A requests the chunk of the video directly n the CDN using the common HTTP scheme. The same is true for sending the REQ. If the node does not send the content of the chunk within 1 second, the node will resort to CDN.

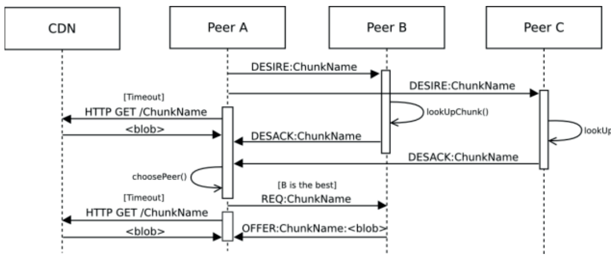


Fig. 5: BemTV Baseline Video Snapshot Trading Scheme

3) *BemTV Baseline Results*

To validate the proposed architecture, an experiment was carried out with the transmission of a live video stream lasting 60 min. 10 Apple MacBook White with 2 CPU cores and 2GB of SDRAM were used. The browser chosen was Mozilla Firefox 27.1, compatible with the latest draft of the WebRTC specification. The comparison test was also done with the same computers and with the Apple Safari 6.0.5 browser, capable of reproducing HLS streams natively.

All computers were on the same 10/100 Mbps Wireless network, which means that they were in the same geographic location and internet provider and, consequently, in the same cluster. The video stream was broken into 5-second chunks at a rate of 600 Kbps bitrate using the HTTP Live Streaming protocol. The CDN was represented by a server with 1 core and 512MB of SDRAM. Fig. 6 shows computer screens during the experiment.

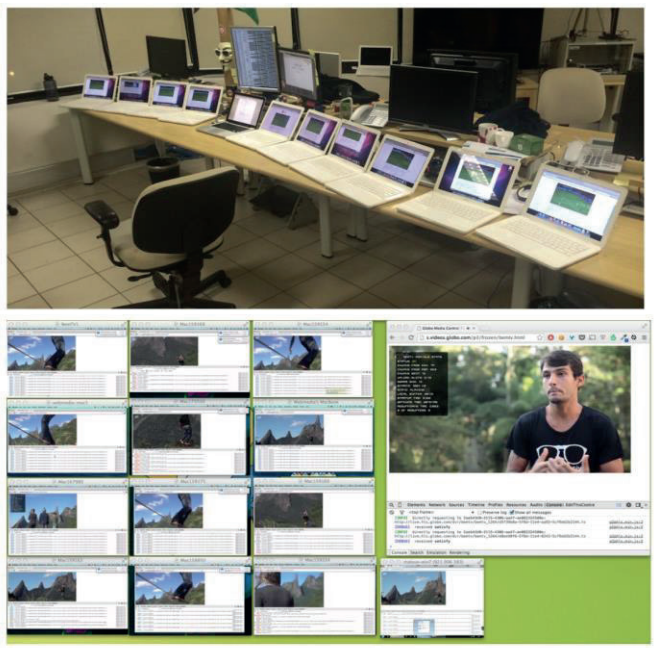


Fig. 6: BemTV Baseline Controlled Experiment

Computers were subjected to an hour of streaming using the native Apple Safari browser and then compared to a streaming consumption of time using the implementation of BemTV Baseline and were analyzed by the number of requests made directly in the CDN. The smaller the number of requests in BemTV Baseline, the more effective the peer-to-peer mesh of changing video chunks will be.

Fig. 7 shows 745 requests made by customers in an hour of experiment, of which 448 were directed to CDN and 297 were directed to BemTV peers. This result indicates promising results of the proposed strategy because, in this experiment, savings of 39.86 % of the egress traffic from CDN were found pointing to an important reduction in the cost of operation of the evaluated video distribution service.

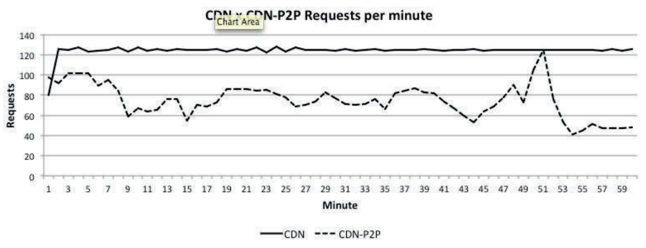


Fig. 7: Comparative Result of Direct Requests to CDN in the conventional model and BemTV Baseline.

B. *BemTV Enhanced*

With the proof of concept validated, we evaluated several protocols for the dynamic promotion of partners and created a balancing heuristic. The final idea is that each node in the peer-to-peer cluster has an optimized view for its current state of connectivity with both the CDN and neighboring nodes in an unstructured (mesh) architecture.

1) Creation of the Peer-to-Peer mesh

A reputation-based mechanism was created, where all users of the P2P loop have different scores associated with each point-to-point connection. When a user starts their session to watch the live video, they request information from a central server that returns up to 50 other peers that are part of that cluster, based on what was described in BemTV Baseline. Upon receiving the connection information with the 50 nodes, the peer will try to establish a WebRTC DataChannel with all of them and assign an initial score of 1000.

During this process, various connectivity problems happen (traverse *NAT, firewalls, proxies),* and the final quantity of successfully established connections is significantly lower than the number of attempts, as illustrated in Fig. 8.

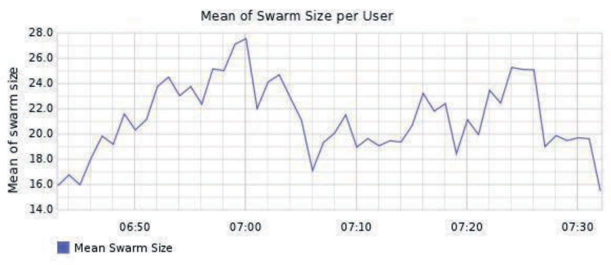


Fig. 8: Swarm size by users during the transmission of Bom Dia São Paulo, Globo.com

With the connection established with other nodes, the user sends a PING and is expected to return a PONG During this stage, the user counts the Round- Trip Delay Time (RTT) or the time it took between this initial handshake and the score to be shot The higher the RTT, the greater the impact on the user's score, which directly translates to the distance in the network topology between the two users. After exchanging PING and PONG messages the user has a picture of the current topology and promotes the top 10 scores to partners

2) Video Segment Requisition and Scores Management

The request to *manifest* file with the playlist of chunks video available in the partner (partners) remains similar to that used in the basic architecture (*baseline*) and does not add delay to the end-user, but the request of the video segments was broken into two stages

3) Requesting Video Segments

To request segments (chunks) of video, the user sends an INTERESTED message to the swarm, and waits for answers from the peers. If it receives CONTAIN back, *interested* and awaiting response s *shock* of partners. When you receive a message *containing the node* decrements of 1 the score of the peer and promotes the candidate partner. When it does not receive a response from the *peer,* the node removes it from the swarm.

Once the candidate with the highest score is identified, the node sends him a REQUEST message. If he receives a choke as a reply, he decreases that partner's score by 1. If, on the other hand, you receive a SATISFY message in response, it is already accompanied by a binary file with the content of the video chunk. This content is finally sent to the playback buffer and the score of the partner is incremented by 1. On the other side, and the partner does not respond it is removed from the cluster.

Then the partner with the highest score is promoted to sender avoiding sending INTERESTED messages to him.

Finally, with each new segment request, the array of peers is reordered promoting peers to partners.

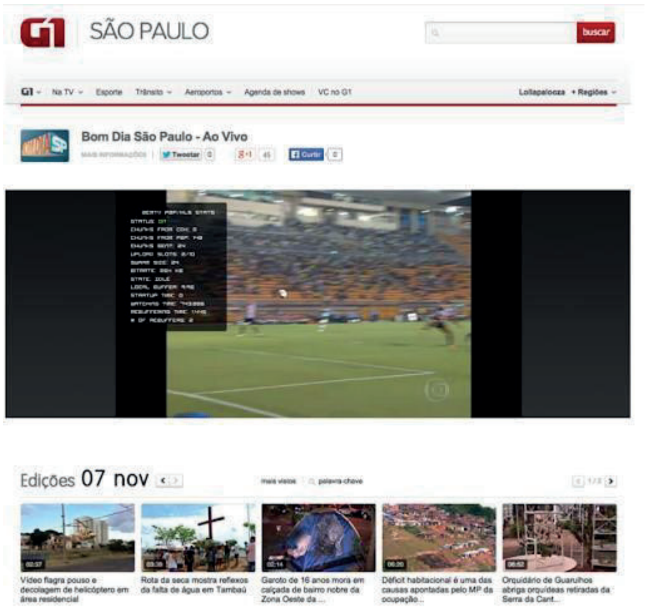


Figure 9: BemTV Enhanced in a production environment for Bom Dia São Paulo on Globo.com.

IV. FINAL RESULTS

The implementation of the advanced architecture was tested in a production environment at Globo.com during the exhibition of a Bom Dia SP show. In this experiment, the number of chunks trafficked in the overlay of pairs was measured, which reached the value of 19759 video chunks. The number of chunks trafficked at the CDN was also measured, resulting in a total value of 22522 video segments. These figures point to an economy of 46.7 3 % of chunks video transmitted during the experiment. Fig. 9 illustrates the screen of a browser running BemTV Enhanced code during the experiment and Fig. 10 shows the number of chunks received by users of CDN and other nodes in the BemTV overlay network.

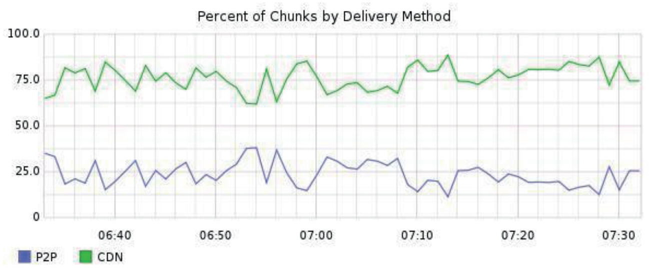


Figure 10: Percentage of Video Segments by Delivery Method in BemTV Enhance.

V. CONCLUSIONS AND FUTURE WORK

Based on the experiment and considering that browsers are constantly evolving, it is possible to conclude that the use of WebRTC as a basis for implementing a hybrid CDNP2P architecture to offer a live streaming video service is not only feasible but a very promising strategy. Optimizations in the data exchange protocol can increase the percentage of chunks negotiated between peers, reducing the cost of transmissions, increasing the scalability of the system, and offering a better user experience for the service consumers. However, several challenges remain open and will be addressed in future work:

- Convergence between pairs and over swarming. The approach described in this article proved to be effective for a relatively small number of users. However, building swarms that aggregate all users in a given location or connected to the same Internet access service provider can generate demands for grouping hundreds or thousands of users leading to an excessive exchange of DESIRE and DESACK messages. The use of reputation [Xiong and Liu 2004], partnership [11], or election of leader [Kutten 2013] are strategies that can be applied to address these problems.
- Video Chunks exchange protocol. With the introduction of improvements in the formation of swarms, we believe that it will be necessary to improve the protocol's stability, and the number of chunks over the P2P network tends to increase. However, research to find the ideal chunks size, video transmission rate, and node cache size will be necessary. The protocol must additionally ensure that the negotiation and transfer of chunks do not interfere with the experience of s users.
- Content security. The protocol presented in this article does not guarantee that the content exchanged between peers is identical to that distributed by CDN. Algorithms for detecting chunks adulterated by DoS starvation as presented [13] need to be analyzed and used.

REFERENCES

[1] Bronzino, F. Gaeta, R. Grangetto, M. Pau, G. (2012) "An Adaptive Hybrid CDN/P2P Solution for Content Delivery Networks". VCIP, páginas 1-6, IEEE.
[2] C.' Müller,' C.' Timmerer,' "A' VLC' Media' Player' Plugin' enabling' Dynamic' Adaptive' Streaming' over' HTTP,"' In' Proceedings' of' the' ACM Multimedia 2011, Scottsdale, Arizona, November 28, 2011.
[3] Cho, S., Cho, J., Shin, S. (2010) "Playback Latency Reduction for Internet Live Video Services in CDN-P2P Hybrid Architecture". 2013 IEEE International Conference on Communications.
[4] Deng, H., Xu, J. (2013). CorePeer: A P2P Mechanism for Hybrid CDN-P2P Architecture. In WebAge Information Management (pp. 278-286). Springer Berlin Heidelberg.
[5] Globo.com (2013) "Rebuffers por Cidade e Operadora – Bayern x Barcelona". HYPERLINK http://goo.gl/VwqXFK, acessado em Outubro, 2014.
[6] Hei, X., Liang, C., Liang, J., Liu, Y., Ross, K. W. (2007). A measurement study of a large-scale P2P IPTV system. Multimedia, IEEE Transactions on,9(8), 1672-1687.
[7] Hei, X., Liu, Y., and Ross, K. (2008). IPTV over P2P streaming networks: the mesh-pull approach. Communications Magazine, IEEE, 46(2):86–92.
[8] Jurca, D., Chakareski, J., Wagner, J. P., Frossard, P. (2007). Enabling adaptive video streaming in P2P systems. IEEE Communications Magazine,45(EPFL-ARTICLE-98484), 108-114.
[9] K. Jonsson, "HttpTools: A Toolkit for Simulation of Web Hosts in OMNeT++", In proceedings of the 2nd OMNeT++ workshop, Rome, Italy, 2009.
[10] Lederer, S., Muller, C., Timmerer, C. (2012, May). Towards peer-assisted dynamic adaptive streaming over HTTP. In Packet Video Workshop (PV), 2012 19th International (pp. 161-166). IEEE.
[11] Li, B., Keung, G. Y., Xie, S., Liu, F., Sun, Y., Yin, H. (2008, November). An empirical study of flash crowd dynamics in a p2p-based live video streaming system. In Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE (pp. 1-5). IEEE.
[12] Lu, Z., Wang, Y., Yang, Y. R. (2012) "An analysis and comparison of CDN-P2P-hybrid content delivery system and model". journal of communications,7(3), 232-245.
[13] Medina-López, C., Naranjo, J.A.M., García-Ortiz, J. P., Casado, L. G., González-Ruiz, V. (2013) "Execution of the P2PSP protocol in parallel environments". XXIV Jornadas de Paralelismo. Madrid, Spain.
[14] Nordyke, K, (2014) "HBO Go Crashes During 'True Detective' Finale". http://goo.gl/zixUr7, acessado em março, 2014.
[15] Rhinow, F., Veloso, P. P., Puyelo, C., Barrett, S., Nuallain, E. O. (2014, January). P2P live video streaming in WebRTC. In Computer Applications and Information Systems (WCCAIS), 2014 World Congress on (pp. 1-6). IEEE.
[16] Roverso R, El-Ansary S., Hogqvist M. (2013-a) "On HTTP live streaming in large enterprises". SIGCOMM '13 Proceedings of the ACM SIGCOMM 2013.
[17] Roverso R. (2013-b) "A System, Tools and Algorithms for Adaptive HTTP-live Streaming on Peer-to-peer Overlays."Doctoral Thesis in Information and Communication Technology, KTH - Royal Institute of Technology Stockholm.
[18] Roverso, R., El-Ansary, S., Haridi, S. (2012) "SmoothCache: HTTP-Live streaming goes peer-to-peer."In Proceedings of the 11th international IFIP TC 6 conference on Networking - Volume Part II (IFIP'12), Robert Bestak, Lukas Kencl, Li Erran Li, Joerg Widmer, and Hao Yin (Eds.), Vol. Part II. Springer-Verlag, Berlin, Heidelberg, 29-43.
[19] Spangler, T. (2009) "YouTube May Lose $470 Million in 2009: Analysts". http://goo.gl/oNgAzY, acessado em Março, 2014.
[20] Tommasi, F., Melle, C., De Luca, V. (2014). OpenSatRelaying: a Hybrid Approach to Real-Time Audio-Video Distribution over the Internet. Journal of Communications, 9(3), 248-261.
[21] Vlavianos, A; Iliofotou, M.; Faloutsos, Michalis, "BiToS: Enhancing BitTorrent for Supporting Streaming Applications," INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, vol., no., pp.1,6, 23-29 April 2006. doi: 10.1109/INFOCOM.2006.43
[22] Vu, L., Gupta, I., Liang, J., Nahrstedt, K. (2006). Mapping the PPLive network: Studying the impacts of media streaming on P2P overlays.
[23] Vu, L., Gupta, I., Nahrstedt, K., Liang, J. (2010). Understanding overlay characteristics of a largescale peer-to-peer IPTV system. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP),6(4), 31.
[24] Wuhib, Fetahi, Rolf Stadler, and Mike Spreitzer. "A gossip protocol for dynamic resource management in large cloud environments." Network and Service Management, IEEE Transactions on 9.2 (2012): 213-225.
[25] Yu-Kwong Kwok, (2011) "Peer-to-Peer Computing: Applications, Architecture, Protocols, and Challenges", Chapman and Hall/CRC Computational Science Series, CRC Press, Taylor and Francis Group, 200 pp., ISBN: 978-1-4398-0934-1.
[26] Zhang, Xinyan, et al. "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming." INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE. Vol. 3. IEEE, 2005.
[27] Zimmerman, A., (2014) "ABC Promised to Livestream the Oscars and Totally Failed". http://goo.gl/sTNb7d, accessed in March, 2014.

Flavio Ribeiro is an MSc. Student in Computer Science at the Federal University of Paraiba and Senior

Director of Engineering at ViacomCBS Inc. working on research, development and deployment of large-scale online video systems. Flavio participated on the design and delivery of large-scale video projects, such as the live streaming stack responsible for the 2014 FIFA World Cup for Grupo Globo, the U.S. Presidential Election Debates for The New York Times and the NFL SuperBowl 53 for CBS.

Guido Lemos de Souza Filho, is a Full Professor in the Department of Computer Systems at the Center for Informatics at the Federal University of Paraíba (DI-CI-UFPB) holds a PhD in Informatics from the Pontifical Catholic University of Rio de Janeiro (PUC-RIO). Coordinates LAVID (Center for Research and Extension in Digital Video Applications) where he works in research on the following topics: digital television, digital cinema, distributed multimedia applications, video distribution networks, distributed artistic performances, accessibility, information security, fakenews and blockchain applications. Worked on the development of the Ginga middleware, published as ITU-T and ITU-R recommendations, and adopted as a standard in the Brazilian Digital Television System and in several countries in Latin America, whose implementation is software currently installed on about 100 million devices TV. The results of their research also include the development of a system for the storage, transmission and display of 4K 3D videos called Fogo Player, the development of a platform to support the realization of distributed dance, theater and music shows called Arthron, the development of video servers for live transmission and on demand, called DLive and DVod, which were used in RNP 's Digital Video Network and in the IPTV service of USP - SP, the VLibras accessibility software (used on www. brasil.gov.br, senado.leg.br and Câmara.leg.br) output 1.5 billion times a year; the development of technologies for registration, validation and preservation of Digital Diploma based on blockchain that will be used in 170 Brazilian public universities; finally, the development of the V4H health video system that uses digital signature technologies, blockchain registration and preservation to add security in the use of videos generated in appointments. He was coordinator of REUNI at UFPB and participated in its creation and implementation foreseen in the increase from 20 to 45 thousand in the number of undergraduate and graduate students. He also serves as a member of the Deliberative Council of the Brazilian Digital Television System Forum and guest of Ancine's Technical Accessibility Chamber.

Received in 2020-11-17 | Approved in 2020-11-21