



A Data-Driven Analysis and Tuning of a Live Hybrid CDN/V2V Video Distribution System

Ishani Sarkar^{1,2(✉)}, Soufiane Roubia¹, Dino Martin Lopez-Pacheco²,
and Guillaume Urvoy-Keller²

¹ Easybroadcast, Nantes, France

`ishani.sarkar@easybroadcast.fr`

² Université Côte d'Azur, CNRS, I3S, Nice, France

Abstract. Video live streaming now represents over 34.97% of the Internet traffic. Typical distribution architectures for this type of service heavily rely on CDNs that enable to meet the stringent QoS requirements of live video applications. As CDN-based solutions are costly to operate, a number of solutions that complement CDN servers with WebRTC have emerged. WebRTC enables direct communications between browsers (viewers). The key idea is to enable viewer to viewer (V2V) video chunks exchanges as far as possible and revert to the CDN servers only if the video chunk has not been received before the timeout. In this work, we present the study we performed on an operational hybrid live video system. Relying on the per exchange statistics that the platform collects, we first present an high level overview of the performance of the system in the wild. A key performance indicator is the fraction of V2V traffic of the system. We demonstrate that the overall performance is driven by a small fraction of users. By further profiling individual clients upload and download performance, we demonstrate that the clients responsible for the chunk losses, i.e. chunks that are not fully uploaded before the deadline, have a poor uplink access. We devised a work-round strategy, where each client evaluates its uplink capacity and refrains from sending to other clients if its past performance is too low. We assess the effectiveness of the approach on the Grid5000 testbed and present live results that confirm the good results achieved in a controlled environment. We are indeed able to reduce the chunk loss rate by almost a factor of two with a negligible impact on the amount of V2V traffic.

1 Introduction

By 2022, the global video traffic in the Internet is expected to grow at a compound annual growth rate of 29%, reaching an 82% share of all IP traffic [2]. The video content is usually delivered to the viewers using a content delivery network (CDN). The huge amount of users puts a high pressure on the CDN networks to ensure a good Quality of Experience (QoE) to the users. It also leads to huge cost for the content owner. This is where a hybrid CDN/V2V (viewer-to-viewer) architecture plays an important role. It allows sharing of the data between different viewers (browsers) while maintaining the QoE for the users.

This paper focuses on a commercial hybrid V2V-CDN system that offers video live streaming channels, where each channel is encoded in different quality levels. More precisely, we focus on the operations of the library that acts as a proxy for fetching the video chunks for the video player. The library strives to fetch the video chunks from other viewers watching the same content and reverts to the CDN in case the chunk is not received fast enough. This operation is fully transparent to the player, which is **independent from the library** and decides the actual quality level based on the adaptive bitrate algorithm it implements, according to the network conditions and/or the buffer level occupancy.

Our hybrid V2V-CDN architecture uses Web-RTC [3] for direct browser communication and a central manager, see Fig. 1. The library is downloaded when the user lands on the Web page of the TV channel. It first uses the Internet Communication Exchange (ICE) protocol along with the STUN and TURN protocols to find its public IP address and port. The library then contacts the central manager using the session description protocol (SDP) to provide its unique ID, ICE data which includes reflexive address (public IP and port), and its playing quality.

The manager sends to the library a list of viewers watching the same content at the same quality level. Those candidate neighbors, called a swarm, are chosen in the same Internet Service Provider (ISP) and/or in the same geographic area as far as possible. The viewer will establish Web-RTC [3] channels with up to 10 neighbors. This maximum swarm size value of 10 in our production system offers a good trade-off between the diversity of video chunks it offers and the efforts needed to maintain those channels active.

When the video player asks for a new video chunk, the library selects the source from which the chunk will be downloaded, either another viewer or a CDN server if the chunk is not available in the swarm. We allow viewers to download data from other viewers within a specified time period which is generally in the order of the size of one video chunk. For example, in the channel used in this paper, the size of one chunk is 6 s for the three different encoding rates.

In terms of global synchronization of the live stream, there is no mechanism to enforce that clients stay synchronized within a given time frame, but a new video client, upon arrival, always asks for the latest available chunk whose id is in the so-called manifest file (list of available chunks, materialized as URLs) that the viewer downloads from the CDN server. Users have the possibility to roll back in time. For the channel we profile, the last 5 h of content is available from the CDN servers. The library maintains a history of the last 30 chunks, corresponding to about 3 min of content.

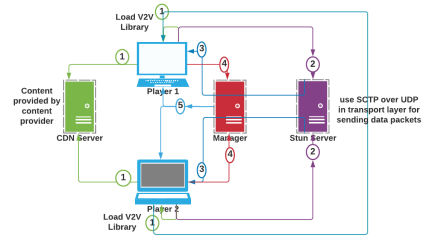


Fig. 1. Overall hybrid V2V-CDN architecture

Although hybrid V2V-CDN systems offer a cost effective alternative to a pure CDN architecture, they need to achieve a trade-off between maintaining video quality and a high fraction of video chunks delivered in V2V mode. Those requirements are somehow contradicting as the V2V content delivery is easier when the content (video chunks) is smaller in size, i.e. for lower video quality.

The contributions of this paper are as follows:

- (i) We present detailed statistics of a 3-day period – with over 34,000 clients and 6.TB of data exchanged – for a popular channel serviced by our commercial live video distribution. We follow an event-based rather than a time-based approach to select those days. Indeed, as the audience of a TV channel varies greatly over time depending on the popularity of the content that is broadcasted, we choose this 3-day period to offer a variety of events, in terms of connected viewers.
- (ii) We question the efficiency of the system using three metrics: V2V Efficiency, which is the fraction of content sent in V2V mode, (application level) Throughput and Chunk Loss Rate (CLR) which is the number of chunks not received before the deadline. These metrics allow to evaluate the efficiency of the library operations. They are specific to the evaluation of the library and differ from classical metrics used at the video player like the number of stalled events and quality level fluctuations.
- (iii) We demonstrate that the root cause of the high observed CLR rate lies at the uplink of some clients, rather than the actual network conditions. This allows us to devise a mitigation strategy that we evaluate in a controlled environment, to prove its effectiveness and then deploy on the same channel that we initially analyzed. We demonstrate that we are able to reduce the observed chunk loss rate by almost 50% with a negligible impact on the fraction of V2V traffic.

2 State of the Art

Several studies have demonstrated that Web-RTC can be successfully used for live video streaming, e.g. [5,6]. The V2V protocol used in this work relies on a mesh architecture to connect different viewers together [4]. The V2V content delivery protocol used applies a proactive approach, which means that the information is disseminated in the V2V network as soon as a single viewer downloads the information. The information is sent to other viewers by using the same Web-RTC channel with a message called *downloaded*. So even if a viewer has not yet requested the resource, it still has the information about all the resources present in its V2V network.

There have been some large scale measurement studies on live video systems done in the past. One of the most popular studies done on a P2P IPTV system is [7] dates back to 2008. In this paper, the authors demonstrate that the current Internet infrastructure was already able to support large P2P networks used to distribute live video streams. They analysed the downloading and uploading bitrate of the peers. They show that there is a lot of fluctuation in the upload

and download bitrate. They also found that the popularity of the content does affect the number of viewers and how easy or difficult it is to find other viewers.

In [8], the authors focused on the problems caused by P2P traffic to ISP networks. This concern is in general addressed in hybrid V2V-CDN architectures through a central manager that can apply simple strategies like offering to a viewer neighbors in the same ISP or geographic location.

3 Overall Channel Profiling

The TV channel we profile in this study is a popular Moroccan channel serviced by our hybrid V2V-CDN system, that offers regular programs like TV series and extraordinary events like football matches. Almost 50% of the clients are in Morocco. The second most popular country is France which represents 15% of the viewers. Italy, Spain, Netherlands, Canada, United States, Germany, Belgium each hosts approximately 4% of the viewers, for a total of about 28% of users. Watching the channel is free of charge. It is accessible using a Web browser only (all browsers now support WebRTC), and not through a dedicated application as can be the case of other channels. On average, 60% of the users use mobile devices to view this channel, whereas 40% of the users use fixed devices.

3.1 Data Set

Our reference data set aggregates three days (from Oct. 2020) of data. Two days have no special events thus the distribution and size of the clients throughout the day remains the same whereas on the third day there is an important event which changes the distribution and size of the clients throughout the day. The channel can be watched at three different quality levels corresponding to 3.5 Mb/s for the smallest quality, 7 Mb/s for the intermediate quality and around 10 Mb/s for the highest quality. These quality levels are selected by the content owner, not the library. Over these three days, we collected information on 34,816 client sessions. On a standard day, the total amount of data downloaded (in CDN or V2V mode) varies between 1.5 and 2 TB whereas in case of big events, the amount of data downloaded is between 6 and 6.5 TB. Figure 2 reports the instantaneous aggregate bit rate over all the clients connected to the channel. The average is at 34 MB/s (372 Mb/s) while for the peak event (a football match), the aggregate throughput reaches 479 MB/s (3,8 Gb/s).

The V2V library reports to the manager detailed logs for all the resource exchanges made by each viewer every 10 s. Over the 3 days, 4,615,045 chunks have been exchanged. The manager later stores those records in a back-end database. Each exchange is labelled with the mode (V2V or CDN) and in case of V2V, the id of the remote viewer. We also have precise information about the time it took to download the chunk or alternatively if a chunk loss event occurred. In addition to per chunk exchange record, we also collect various player level information as well like watching time, video quality level, operating system (OS), browser, city, country, Internet service provider (ISP), etc. We also collect

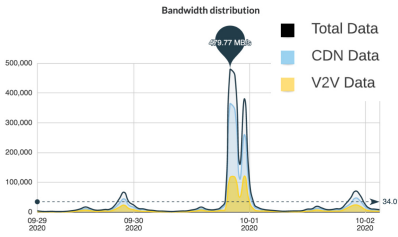


Fig. 2. Initial 3-day period

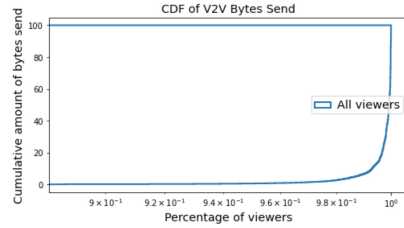


Fig. 3. V2V size bytes distribution

various other viewer information as well like to how many viewers a viewer is connected to simultaneously (swarm size), how many consecutive uploads to the other viewer has been done, rebuffering time, rebuffering count etc.

3.2 Clients Profiling and V2V Efficiency

The V2V paradigm directly inherits from the P2P paradigm where a significant problem was the selfishness of users [1]. We are not in this situation here as on one side, the V2V library is under our control and second, the choice of a viewer to request a chunk from, is done at random among the peers possessing this chunk. Still, we observe a clearly biased distribution of viewers contribution with 1% of the viewers responsible for over 90% of the bytes exchanged, as can be seen from Fig. 3. This bias in the contribution is in fact related to the time actually spent by the user watching the channel. We report session times in Fig. 4. Since most of the V2V data is sent by only 1% of the viewers, we compare the session time of all the viewers with these 1% of most active viewers. We can readily observe in Fig. 4 that the top 1% active viewers feature a bimodal distribution of session time with around 25% of clients staying less than 1 min and the rest staying in general between 30 min and a few hours. In contrast, the overall distribution (all users) is dominated by short session times with 60% of users staying less than 10 min.

Another factor that is likely to heavily affect the viewer ability to perform effective V2V exchanges is its network access characteristics. As part of the content is downloaded from the CDN servers which are likely to be close to the client and feature good network performance, the average throughput achieved during chunks downloads from the CDN provides a good hint on the network access capacity of the user. Note that as a chunk is several MB large, the resulting throughput should be statistically meaningful.

As we see from Fig. 6, there is a significant difference between the CDN bitrates of the overall viewers and most active 1% viewers, which experience way higher throughputs. The correlation coefficients between CDN bitrate and chunk loss rate (CLR) for overall viewers is -0.47 and for most active top 1% viewers, it is -0.7 . Ideally, one expects this value to be indeed negative as the better the access link of the user is, the less likely it is to miss the deadline when sending or receiving a chunk. From this perspective, the CLR is highly correlated

with the CDN throughput performance for the top 1% of users, hinting that this metric is a good estimator of the reception quality.

The actual chunk lost rate (CLR) of the overall viewers and most active viewers are reported in Fig. 5. We can clearly observe that for the most active 1% of users, the distribution is skewed to the left. Indeed, over 50% of these users experience less than 20% CLR, while the others experience a CLR roughly uniformly distributed between 20 and 75%.

To further understand the observed CLR, and how to reduce it, we carry a detailed study the CLR in the next section.

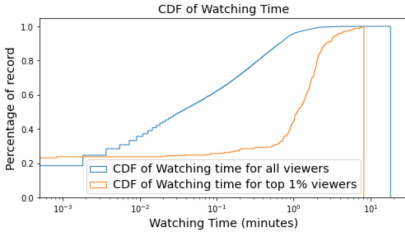


Fig. 4. Watching time distribution

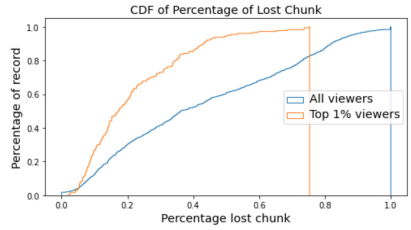


Fig. 5. Lost chunk rate distribution

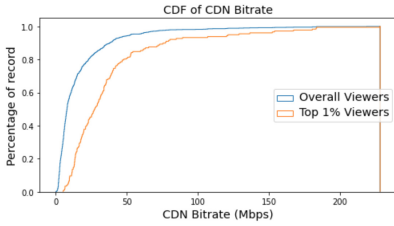


Fig. 6. CDN bandwidth distribution

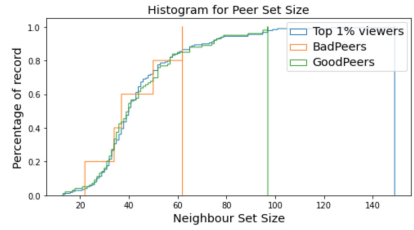


Fig. 7. Viewer’s neighbour set size

4 Detailed Analysis of Chunk Loss Rate (CLR)

We focus in this section on the 1% most active users viewers with more than 1 min session time. We formulated hypotheses to identify the root causes behind the observed lost data chunks:

- H_1 : The swarm size affects the lost chunk rate of a viewer, because the bigger the swarm size, the more control messages you receive, thus more network traffic resulting in a higher CLR.
- H_2 : The type of client access affects the lost chunk rate. Ideally, we would like to know the exact type of network access the client is using: Mobile, ADSL, FTTH. The library is not able (allowed) to collect such information. We can however classify clients as mobile or fixed lines clients based on the user-agent HTTP string.

- H_3 : The network access link characteristics directly affects the CLR. We already studied the download rate of the users using the transfers made with the CDN servers. The download and especially the upload rates achieved during V2V exchanges can also be used to understand the characteristics of the client access link.

Based on the observation we made on Fig. 5, we form two groups of users (for the top 1%) that we term good or bad. The viewers with less than 20% CLR are categorised as good viewers while viewers with more than 60% CLR values are categorised as bad viewers. The rationale behind this approach is to uncover key features of clients that can lead to small and large CLR so as to isolate ill-behaving clients and improve the V2C efficiency.

H₁ Hypothesis. The first hypothesis states that the neighbour set size of the viewers should affect the CLR. Figure 7 presents the CDF of the peer set size of good and bad viewers. We can observe that bad peers tend to have smaller peer set size than good peers. While this could hint towards the fact that bad peers have more difficulties to establish links with other viewers, we believe that the actual session times play a key role, as the longer the session, the more likely a peer is to establish more connections. This is indeed the case here as bad peers have an average session time of 22 min while it is 160 min for the good peers. We however also found that the correlation coefficient between neighbour set size and CLR is only 0.05 and 0.07 for good and bad peers respectively. Thus although we observe distinct distributions for good and bad viewers, the neighbour set size does not seem to have any direct correlation with the CLR.

H₂ Hypothesis. The second hypothesis is to check if the type of device affects the CLR. We have two families of devices: desktop devices and mobile devices. As a mobile (resp. desktop) device can send to a desktop or mobile device, we have 4 possible combinations to consider. We plotted the distributions of CLR for the good and bad viewers for all the four combinations in Figs. 8 and 9 respectively. For the good users, the type of device does not seem to play a significant role¹. For the bad viewers, we have very few cases of desktop senders, which is understandable as the worse network conditions are likely to be experienced on mobile devices. This hints towards putting the blame on the user access link that we investigate further with hypothesis H_3 .

H₃ Hypothesis. We now investigate the impact on the CLR of the access link characteristics of the users that we indirectly estimate based on the bandwidth achieved during transfers with CDN servers and other viewers. From Fig. 10, we observe that 50% of the bad viewers have just 10Mbps of CDN bandwidth whereas 50% of the good viewers have about 25Mbps of CDN bandwidth. The coefficients of correlation between CLR and CDN bandwidth for the good viewers and bad viewers are -0.45 and -0.4 respectively.

¹ Note that the good users in Fig. 8 can experience CLR higher than 20% for some categories, as the threshold of 20% applies to the average CLR and not per category.

Looking at the V2V download rates should enable to estimate the uplink of the users as it is likely to be the bottleneck of the path. From Fig. 11, we clearly see that the V2V downloading rate of good viewers is far better than the one of bad viewers. It thus appears that a key factor that explains the observed CLR is the uplink capacity of the peers. In the next section, we leverage this information to devise a simple algorithm, that can be applied independently at each viewer and helps reducing the CLR.

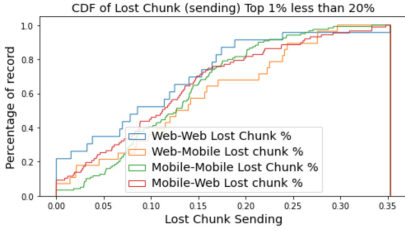


Fig. 8. CLR of good viewers

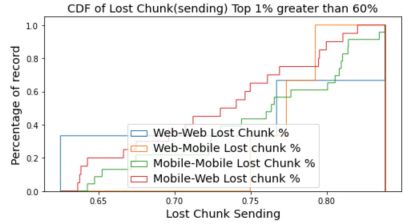


Fig. 9. CLR of bad viewers

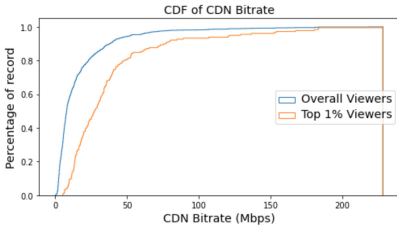


Fig. 10. CDN bandwidth distribution

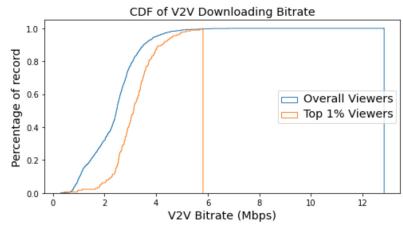


Fig. 11. V2V bandwidth distribution

5 CLR Mitigation Algorithm

Our objective is to achieve a trade-off between CLR reduction and a decrease of V2V traffic. Indeed, a simple but not cost effective way to reduce the CLR is to favor CDN transfers at the expense of V2V transfers. Results of the previous section have uncovered that a key (even though probably not the only one) explanation behind high CLR is the weakness of the uplink capacity of peers. We thus devised a simple approach that allows viewers to identify themselves as good or bad viewers by monitoring their chunk upload success rate. The algorithm checks every second the CLR, and if it goes above a threshold of $th\%$, the viewer stops sending the so-called *downloaded* control messages, which indicate to its neighbors that it has a new available chunk. As the viewers won't send a *downloaded* message, they will not receive a request for that resource, which will reduce their lost data rate. Note that viewers can still request and

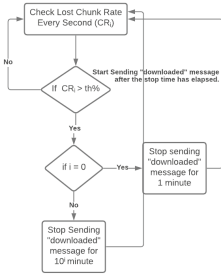


Fig. 12. Algorithm

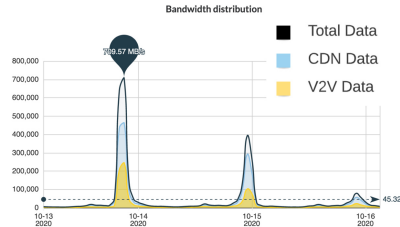


Fig. 13. Second 3-day period

receive chunks in V2V mode from other viewers. This is motivated by the fact that the access links tend to be asymmetric with more download than upload capacity.

The algorithm (Fig. 12) implements a backoff strategy where the viewer alternates between full V2V (receiving and sending) and partial V2V (only receiving) mode to account for possible channel variations or varying congestion in the network. The first time the threshold th is reached, the viewer stops sending *downloaded* messages for 10^0 min and then starts again monitoring the CLR every second for one minute afterwards. If a second consecutive period of CLR over the threshold is observed, the viewer stops sending downloaded messages for 10^1 min and so on (i consecutive events lead to a period of 10^i minutes long silence period). In between silence periods, the test periods, where the viewer is allowed again to upload, last one minute.

In the next section, we report on tests performed with our CLR mitigation algorithm on a test-bed and in production in the live channel used in Sect. 3.

6 Evaluation

We evaluate our CLR mitigation algorithm first in a controlled environment which features 60 viewers and second in our production environment. While modest in size, the controlled environment is useful as it enables to : (i) perform functional tests as the client code in the same as the one in production, (ii) emulate a variety of client network conditions by tuning the upload and download rate of clients, even though we cannot reproduce the full diversity of network conditions observed in the wild and (iii) perform reproducible tests, which is unfeasible in the wild.

6.1 Test-Bed Results

Our test-bed was deployed on 4 physical servers on the Grid’5000 experimental platform [9] which uses KVM virtualisation. Each server hosts 4 virtual machines with 15 viewers per VM, for a total of 60 unique viewers. The viewers are connected to a forked version of the channel presented in Sect. 3, where they operate in isolation, i.e. they can only contact the CDN server and the local viewers.

We relied on Linux namespaces to create isolated viewers. The download capacity of each virtual node is around 325 Mb/s. Each experiment lasts 40 min. To emulate bad viewers, we capped their upload capacity, using the Netem module of Linux, to 3 Mb/s, a value smaller than the smallest bitrate, corresponding to smallest video quality of the channel. In contrast, we impose no constraints on their uplink. We created three different scenarios: (i) Scen. 1: 15 bad viewers and 45 good viewers, (ii) Scen. 2: 30 bad viewers and 30 good viewers and (iii) Scen. 3: 45 bad viewers and 15 good viewers.

Table 1 reports the fraction of chunks downloaded from the CDN or in V2V mode as well as the CLR for the three scenarios with the CLR mitigation algorithm on and off. Clearly, the V2V efficiency is not affected (it even increases) when the algorithm is turned on while the CLR significantly decreases. The CLR does not reach 0 as when the bad peers are in their test periods (in between silence periods) they can be picked as candidates by the good peers.

Table 1. CDN V2V and LCR rate For V2V protocol with and without algorithm

	No Algorithm			Algorithm		
	CDN%	V2V%	LCR	CDN%	V2V%	LCR
Scen. 1	36.7	63.3	7.85	36.37	63.3	3.38
Scen. 2	46.72	53.28	13.52	55.78	64.21	5.58
Scen. 3	66.4	33.6	38.7	53.51	46.49	7.75

6.2 Results in the Wild

We now present the result of a 3-day evaluation for the same channel as in Sect. 3 where the CLR mitigation algorithm is deployed. Figure 13 represents the evolution of aggregated traffic over the three days. We used a conservative approach and used a threshold $th = 80\%$ for this experiment, as we test on an operational channel.

The three days picked for the initial analysis in Sect. 3 were in fact chosen so as to offer a similar profile (with at least one major event) as the period where the algorithm was deployed. This enables to compare the two sets of days, even if we can not guarantee reproducibility due to the nature of the experiment.

We first focus on the V2V efficiency which is the most important factor for the broadcaster. We want the algorithm to reduce the CLR but not the

V2V efficiency as far as possible. The aggregated V2V efficiency for the days without the CLR mitigation algorithm is 28.98% whereas it is 30.61% when it is turned on. The scale of the events does affect the V2V % for both the algorithms. For a small (resp. large) scale event where the total data download remains less than 1.5 TB (over 6TB), the V2V protocol without mitigation algorithm has 32.5% (resp. 27.47%) of V2V efficiency whereas the V2V protocol with algorithm features an efficiency of 28.9% (resp. 32.55%). This suggests that when the protocol has enough viewers with good download capacity, there is no big performance impact on V2V efficiency. Even in the case of less viewers, the V2V efficiency percentage is reduced by only 4%.

The second metric we consider is the CLR. The overall (over the three days) CLR without the algorithm was 24.7% whereas it fell to 13.0% when the algorithm was turned on. Thus overall, the algorithm reduced the CLR by almost a factor of 2.

We further compared the distributions of the CLR for good viewers and bad viewers, using the same definition as in Sect. 3, for the two periods of 3 days in Figs. 14 and 15 respectively. We clearly observe the positive impact of the CLR mitigation algorithm on both the good and bad peers with more mass on the smaller CLR values, e.g. almost 22% of the good viewers do not lose any data at all.

As explained in the introduction, the library operations are transparent to the video player. One can however question if our CLR mitigation algorithm can adversely impact the video player by indirectly influencing the video quality level it picks. As a preliminary assessment of the interplay between the library and the player, we report in Table 2 the fraction of sessions at each quality level observed, per day, for the two periods of interest for the top 1% of viewers. We observe no noticeable difference in the distributions of client sessions at each quality level for the two periods, which suggests that the CLR mitigation algorithm has no collateral effect.

Table 2. Video quality levels distribution (top 1% viewers)

	Period 1 (no algo.)			Period 2 (algo.)		
	Low Q. %	Medium Q. %	High Q. %	Low Q. %	Medium Q. %	High Q. %
Day 1	27	30	44	33	29	38
Day 2	33	25	42	30	25	45
Day 3	30	35	35	28	34	38

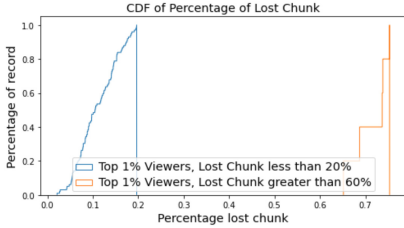


Fig. 14. CLR **No** mitigation alg.

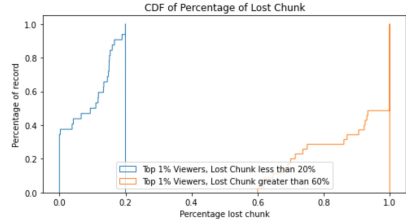


Fig. 15. CLR **With** mitigation alg.

7 Conclusion and Future Work

In this work, we have presented an in-depth study of a live video channel operated over the Internet using a hybrid CDN-V2V architecture. For such an architecture, the main KPI is the fraction of chunks delivered in V2V mode. The chunk loss rate (CLR) metric is another key factor. It indicates, when it reaches high values, that some inefficiencies exist in the system design since some chunks are sent but not delivered (before the deadline) to the viewers that requested them.

We have followed a data driven approach to profile the clients and relate the observed CLRs to other parameters related to the neighborhood characteristics, the type of clients (mobile or fixed) or the access link characteristics. The latter is inferred indirectly using the throughput samples obtained when downloading from the CDN or uploading to other peers. We demonstrated that, in a number of cases, the blame was to put on the access links of some of the viewers. We devised a mitigation algorithm that requires no cooperation between clients as each client individually assesses its uplink capacity and decides if it acts as server for the other peers or simply downloads in V2V mode. We demonstrated the effectiveness of the approach in a controlled testbed and then in the wild, with observed gains close to 50% with a negligible impact on the V2V efficiency. As our library is independent from the actual viewer, and simply acts as a proxy between the CDN server and the video player by re-routing requests for the content to other viewers if possible, our study provides a way to optimise any similar hybrid V2V architecture.

The next steps for us will be to devise an adaptive version of our CLR mitigation algorithm and test at a larger scale on the set of channels operated by our hybrid CDN-P2P live delivery system. We also want to study in more detail the relation between our QoS metrics at the library level and the classical QoE metrics used at the video player level.

References

1. Cohen, B.: Incentives build robustness in BitTorrent. In: Workshop on Economics of Peer-to-Peer systems, vol. 6 (2003)
2. Alex Bybyk **2**(5) (2020). <https://restream.io/blog/live-streaming-statistics/>. Accessed 18 Oct 2020

3. Bergkvist, A., Burnett, D.C., Jennings, C., Narayanan, A., Aboba, B.: Webrtc 1.0: Real-time communication between browsers. Working draft, W3C (2012)
4. Sarkar, I., Rouibia, S., Pacheco, D.L., Urvoy-Keller, G.: Proactive Information Dissemination in WebRTC-based Live Video Distribution. In: IWCMC, pp. 304–309 (2020)
5. Bruneau-Queyreix, J., Lacaud, M., Négru, D.: Increasing End-User’s QoE with a Hybrid P2P/Multi-Server streaming solution based on dash.js and webRTC (2017). fhal-01585219f
6. Rhinow, F., Veloso, P.P., Puyelo, C., Barrett, S., Nuallain, E.O.: P2P live video streaming in WebRTC. In: World Congress on Computer Applications and Information Systems (WCCAIS), Hammamet, vol. 2014, pp. 1–6 (2014). <https://doi.org/10.1109/WCCAIS.2014.6916588>
7. Hei, X., Liang, C., Liang, J., Liu, Y., Ross, K.: A measurement study of a large-scale P2P IPTV system. *IEEE Trans. Multimedia* **9**, 1672–1687 (2008). <https://doi.org/10.1109/TMM.2007.907451>
8. Silverston, T., Jakab, L., Cabellos-Aparicio, A., Fourmaux, O., Salamatian, K., et al.: Large-scale Measurement Experiments of P2P-TV Systems Insights on Fairness and Locality. *Signal Process. Image Commun.* **26**(7), 327–338 (2011). f10.1016/j.image.2011.01.007f. fhal-00648019f
9. Bolze, R., et al.: Grid’5000: a large scale and highly reconfigurable experimental grid testbed. *Int. J. High Perform. Comput. Appl.* **20**(4), 481–494 (2006)