

MEC Proxy for efficient cache and reliable multi-CDN video distribution

Roberto Viola, Angel Martin and Mikel Zorrilla
Digital Media Department
Vicomtech
San Sebastián, 20009 Spain
Email: {rviola, amartin, mzorilla}@vicomtech.org

Jon Montalbán
Electronic Technology Department
University of the Basque Country
San Sebastián, 20018 Spain
Email: jon.montalban@ehu.es

Abstract—The massive consumption of media contents needs of network accelerators, which boost the media delivery and optimize the traffic volume crossing the network from servers to media players. Content Delivery Network (CDN) is the common network function to distribute in a cloud-manner the contents, enhancing media availability and distribution performance. However, high concurrency rates of media sessions can produce CDN performance degradations and outages that impact negatively the Quality of Experience (QoE). 5G Multi-access Edge Computing (MEC) architecture envisions a QoE-aware system at the network edge which performs analytics to enhance or boost media services. This paper provides a novel MEC proxy to expand and enforce caching infrastructures for efficient and reliable content distribution. First, the proxy caches contents on the network edge to reduce the Capital Expenditure (CAPEX) of the CDN for the OTT service provider. To this end, the proxy is able to identify recurrent requests. Second, the proxy shields from identified or predicted CDN malfunction. Here, the proxy switches the download sessions to an alternative CDN in order to ensure QoE rates, enabling a CDN dynamic selection based on live connectivity statistics. The proposed solution is evaluated by delivering Dynamic Adaptive Streaming over HTTP (MPEG-DASH) streams in a dense client cell while applying different caching strategies.

Index Terms—5G, CDN, distributed cache, MEC, QoE, reliability & streaming analytics.

I. INTRODUCTION

The increase of video streaming users and the service requirements are driving the evolution of media services over the Internet. Mobility and quality improvements are the key catalysers in this evolution. Cisco estimates that video traffic will cover the 82% of all Internet traffic by 2021 [1]. Moreover, mobile traffic growth is estimated to be twice as fast as fixed IP traffic from 2016 to 2021 [2].

The delivery performance from the network has a big impact on the perceived QoE of the end user of media services. However, video streaming services work on top of unmanaged networks, where the traffic is managed and forwarded on a best-effort basis [3]. This operational context is targeted by the video streaming solution MPEG-DASH. MPEG-DASH inherits many benefits from Hypertext Transmission Protocol (HTTP). First, it enables a pull-based streaming [4] to easily traverses network functions such as firewalls and NAT devices. Second, MPEG-DASH streams can be played anywhere as any

connected device supports HTTP. Third, MPEG-DASH has a Content Delivery Network (CDN)-ready design enabling the exploitation of existing HTTP caching infrastructures, which enhance the availability and the responsiveness of the content distribution.

Beyond the current network functions to boost and enhance the traffic delivery, the capacity and performance promised by 5G networks will make a significant leap towards higher data rates, heavier user densities and ultra-reliable and low-latency communications. The European Telecommunications Standards Institute (ETSI) specifies a set of component technologies which will be essential part of 5G systems, such as, Network Functions Virtualization (NFV), Millimetre Wave Transmission (mWT), Next Generation Protocols (NGP) and Multi-access Edge Computing (MEC) [5]. MEC architecture allows Mobile Network Operators (MNOs) to supply video delivery analytics-based intelligence at the network edge. Thus, MEC plays a significant role to achieve specific media traffic goals with zero-latency in a distributed manner. Furthermore, MEC enables coordinated operations at the network edge such to be transparent to the media server and players.

This work proposes a novel network proxy, called MEC4CDN, which complies with MEC architecture. MEC4CDN provides two features. First, it performs a local cache of identified trending contents to minimize the traffic between the CDN and the network edge. Second, it applies a dynamic selection of the CDN based on live measurements in the context of a multi-CDN delivery. The proposed solution allows the client to start playing from a CDN, that is favourable in terms of Quality of Service (QoS), and transparently and dynamically switch to another one which provides better performance in case of CDN degradation or outage. To this end, the proxy collects and process L3 connectivity metrics, L7 MPEG-DASH Media Presentation Description (MPD) files and L7 QoE scores. So, the proxy handles multiple CDN endpoints, deciding to locally cache a response for near future requests, or to conduct the request to a healthy cloud CDN in case of detected issues.

The paper is structured as follows. First, section II reviews the related work to improve the network delivery. Then, section

III describes the proposed MEC4CDN network proxy. Section IV presents the implementation of the solution. To validate the solution, section V compiles the testbed and the results. Finally, section VI gathers the conclusions of this work.

II. RELATED WORK

Apart from the content catalogue, the Quality of Experience (QoE) is a key aspect for user satisfaction and retention when rating streaming services. Hence, any system trying to enhance media delivery needs to consider QoE metrics.

QoS assesses the network delivery performance and it has a direct impact on human perception QoE. In this sense, stability of media playback is related to efficient utilization and fair resource sharing. Nevertheless, these key aspects are not guaranteed in unmanaged networks. This situation may lead to suboptimal results in terms of video playback, link utilization, and fairness among the clients [6].

Specific metrics for QoE of HTTP-based adaptive media streaming services, such as initial delay, stalling time, number of quality switches and inter-switching times, are fundamental parameters to get the estimated Mean Opinion Score (eMOS) [7]. This model quantifies the quality of video streaming services without a demographic perception study. Recently, the work [8] investigates a new model for MOS, called Ubiquitous-Mean Opinion Score for Video (U-vMOS), which makes initial buffering more dominant than [7] model.

Caching is a common technique to get enhanced QoE for massive content consumption services. The CDN is the traditional network function provisioning cache features as a cloud service. Fueled by the CDN vendor proliferation, media services employ multi-CDN strategies to get a more reliable and cost-effective content delivery. Netflix case, as an example of multi-CDN solution, is studied in [9]. More recent work [10] also includes Hulu analysis for 3 CDN vendors. Here, with alternative CDNs available, the employed CDN is set by the streaming service for each session, where the choice is done by the server. This centralized architecture is difficult to scale hampering the orchestration of common policies or decisions to all users in an area.

When the employed CDN is not pre-set, transferring to clients the possibility to dynamically choose, each client should analyze repeatedly the network performance of each CDN. It means the introduction of a network overhead proportional to the number of clients and the number of CDNs. Hence, a client-side CDN selection is not an optimal solution.

Consequently, the European Broadcasting Union (EBU) aims to avoid both pre-set CDN and client's CDN selection by proposing the EBU Flow Multi-CDN [11]. It consists in a CDN switching service which selects the optimal CDN at any given moment in time. A similar solution is also provided by Cedexis Multi-CDN [12]. Both EBU and Cedexis proposals monitor network analytics at core network, then they are not aware of the state of connected clients at network edge.

On the contrary, if we focus at the network edge, the access point has knowledge of both the connected clients' activity in its cell and the CDN performance from its location. A

proxy located at the network edge, retrieving access point information, can evaluate the performance of the delivery network just once and exploit it for each client (independently from the number of clients in the cell). This perfectly suits the telecommunication industry which proposes MEC as a new functional architecture to be integrated on the mobile network infrastructure. MEC allows Internet service provider (ISP) to provide Information Technology (IT) and cloud computing services at the edge of the network, closer to the clients. Hence, MEC opens the door for authorized Content Providers (CPs) to develop their own applications hosted on the MEC server. Therefore, CPs are able to tune the behavior of content delivery to end users in 4G and 5G contexts. Furthermore, next generation networks and 5G MEC architecture enable the deployment of distributed local caches at the network edge to efficiently minimize the volume of traffic passing through the network core and backhaul.

III. MEC PROXY FOR MULTI-CDN DELIVERY

The proposed MEC4CDN proxy server, located on a MEC component, can exploit the knowledge of network analytics of different delivery paths. It aims to improve the overall throughput while minimizing the latency, in an environment where multiple clients are competing for network resources.

Figure 1 shows a scenario where a content is provided by means of two CDN vendors. For some reasons, such as sudden massive connections or geo-based cycles of human activity, the performance of *CDN A* starts to degrade. At the same time, the capacity of *CDN B* is improved because the *CDN B* starts to provision more resources for a specific area. The MEC4CDN system gets awareness of this unbalanced situation, based on L3 stats, and orchestrates all the media players subscribed to the managed cell to dynamically employ a CDN which ensures better QoS.

Therefore, for services delivered over multiple CDN providers, MEC4CDN can select an appropriate CDN for a RAN geo-position in real-time, according to L3 metrics. To this end, MEC4CDN gets alternative CDNs set from the MPD file, provided that it contains multiple definitions of *base URL* storing the path to the CDNs, or from the media service. Then, MEC4CDN employs the set of CDNs to dynamically switch the base URL of media sessions in the same RAN when necessary. So, in case of detected performance degradation, the MEC4CDN system replaces the base URL field of all the managed sessions to another known CDN endpoint, migrating all the managed clients at once to avoid outages.

The sequence diagram with the exchanged messages to allow the CDN switching is depicted in Figure 2. First, the MEC proxy server running MEC4CDN captures the HTTP GET requests from the User Equipments (UEs) to download the MPD file from the media server. Then, MEC4CDN retrieves the MPD file from the media server and appropriately parses it in order to retrieve the *base URLs* set before sending it to the UE. Then, the UE selects a representation bitrate (R_j) from the available ones according to display resolution, user preferences and connection capacity stats. The UE requests,

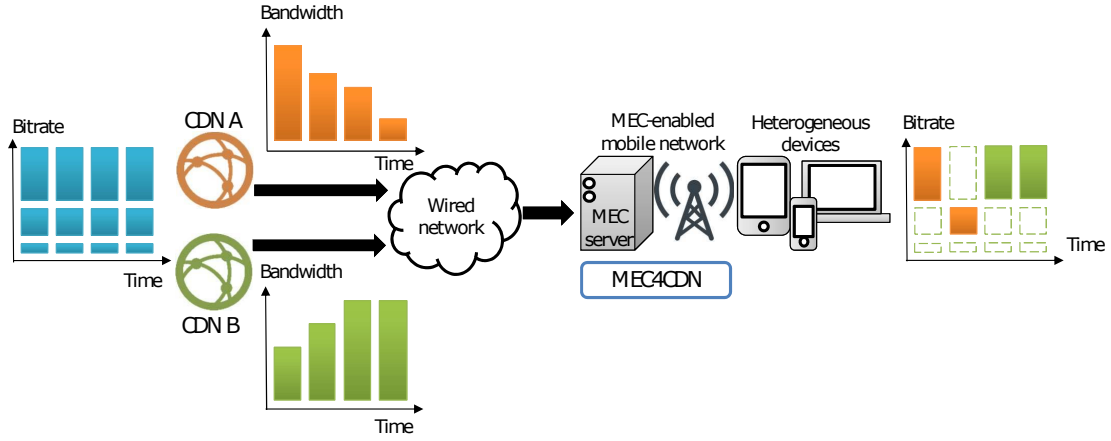


Fig. 1: MEC4CDN multi-CDN selection.

through the MEC proxy, a specific segment file to the CDN accordingly. Once the MEC proxy server detects the HTTP GET requests from the UEs for downloading a segment file, it retrieves wired path stats. When network performance is not enough, the MEC4CDN switches the *base URL* field of the MPD file applicable for the next segment requests. Such operations are executed at the stream start and each time that the UEs asks for a new segment.

manner. Triggered by the first content request, MEC4CDN downloads all the next segment representations from the CDN. Then, all the following requests are already downloaded and available in the MEC4CDN local cache for any representation bitrate chosen by the media players. Then, the greater the number of clients consuming a live content in a cell, the higher efficiency of this solution.

The utilization of a CDN means a significant operational cost for media services. Moreover, when massive connections come concurrently to a CDN, the CDN infrastructure can start to collapse and the QoS could be negatively impacted. To prevent media delivery from this situation, it is necessary to turn broadcast of live content into an efficient distribution. To this end, MEC systems located at the network edge can cache recurrent contents, such as live sports events or concerts, in a proactive manner. Cache at network edge can improve the streaming experience caching all available representations. Thus, the selected representation will be locally available for any number of subscribers in a cell, enabling the video to start faster and reducing the buffering time, while minimizing CDN traffic. This approach preserves media players ability to decide the bitrate that fits with their contexts independently.

The sequence diagram with the exchanged messages to allow the local cache of a popular or live content is depicted in Figure 4. The proxy waits for a media content request and proceeds to download all the available representations. Then, the media players in the cell will select a specific representation (R_j) according to the device features, the assessed available bandwidth, the user preferences and subscription plan. Once the players start to request a representation for a specific time (t), the proxy requests the next segments ($Segments_{t+1} \forall R_j$) to be ready for the following requests.

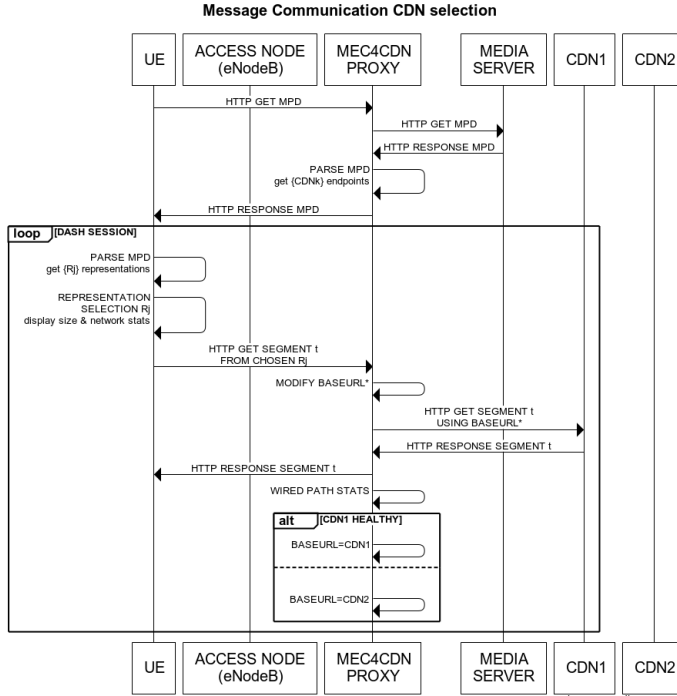


Fig. 2: MEC4CDN multi-CDN sequence diagram.

Furthermore, MEC4CDN ships the ability to significantly reduce the CDN traffic for a live content distribution in dense client cells, as depicted in Figure 3b. In a live media consumption scenario, each UE makes a request to get the video from a CDN. When MEC4CDN comes into play, it performs a local cache at the network edge in a proactive

It is also important to remark the zero-latency and distributed performance features by design coming from the MEC architecture fundamentals. The MEC systems are distributed and autonomously empower specific services for the co-located cell.

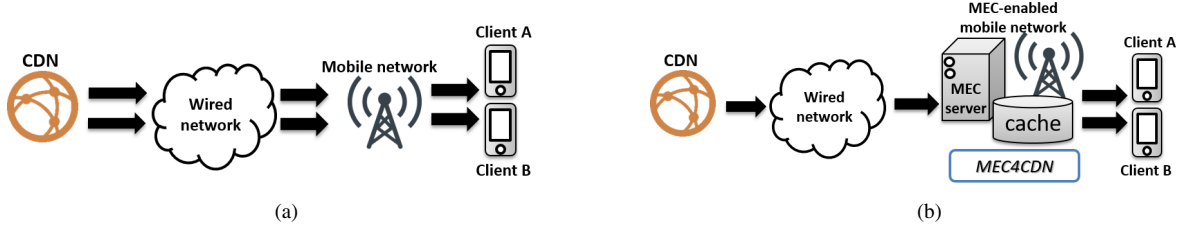


Fig. 3: Legacy content delivery **a** and MEC4CDN cache-powered delivery **b**.

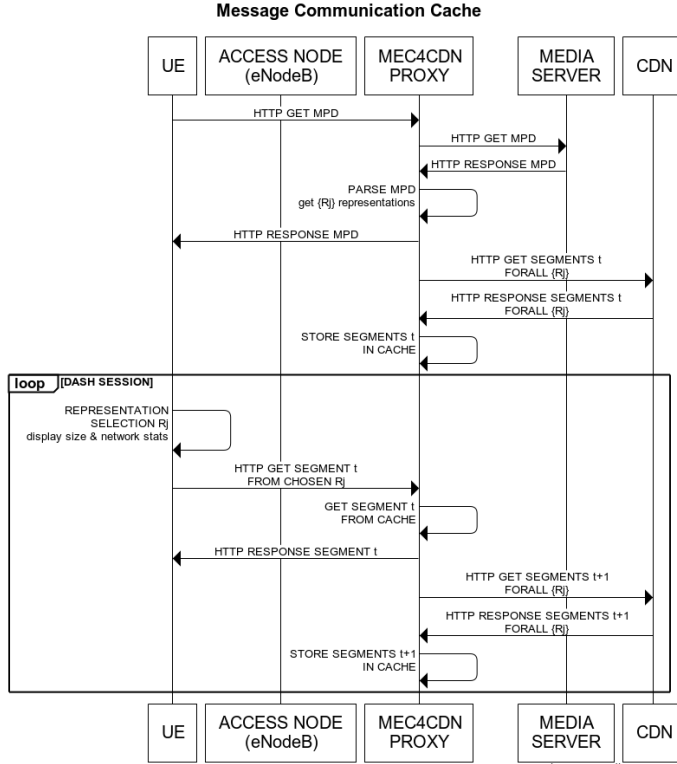


Fig. 4: MEC4CDN cache sequence diagram.

IV. IMPLEMENTATION

The program of MEC4CDN for CDN selection is described in Algorithm 1. The outcome of MEC4CDN is to identify violations of bandwidth or latency performance in order to perform reactive switching to an alternative CDN provider. The inputs of the algorithm are the RTT and the bandwidth from the MEC4CDN proxy to the CDN and the current number of media playing sessions to this CDN. The output is the base URL field of the updated MPD file to be used by the media player.

Moreover, the program of MEC4CDN for local cache is described in Algorithm 2. The outcome of MEC4CDN is to proactively download what the users will play in order to locally cache next segments before the player needs to play them, enabling the video to start faster, decreasing stalls, reducing CDN traffic and enhancing the quality of experience.

Algorithm 1 CDN health check to switch all sessions at eNodeB

```

procedure CDNCHECK( )
     $rtt_{max}$   $\triangleright$  listen to requests & switch CDN
     $CDN_{list}$   $\triangleright$  setup maximum latency
     $n_p$   $\triangleright$  set of alternative CDNs
     $n_p$   $\triangleright$  number of players
    for all segment request do
        SegmentRequest()
         $s_{CDN_k} \leftarrow$  SegmentResponse()  $\triangleright$  from the UE
         $rtt_{CDN_k} \leftarrow rtt(s_{CDN_k})$   $\triangleright$  to the  $CDN_k$ 
         $rtt_{CDN_k} \leftarrow rtt(s_{CDN_k})$   $\triangleright$  from the  $CDN_k$ 
         $rtt_{CDN_k} \leftarrow rtt(s_{CDN_k})$   $\triangleright$  L3 latency
         $bw_{CDN_k} \leftarrow 2 \frac{size(s_{CDN_k})}{rtt(s_{CDN_k})}$   $\triangleright$  L3 BW
         $R_{max}^{bitrate} \leftarrow$  parse(MPD)  $\triangleright$  maximum representation bitrate
        if ( $rtt_{CDN_k} > rtt_{max}$ ) || ( $bw_{CDN_k} < (n_p \times R_{max}^{bitrate})$ ) then
             $\triangleright$  insufficient performance
             $baseURL \leftarrow alternative(CDN_{list}, CDN_k)$ 
             $\triangleright$  change CDN

```

Thus, MEC4CDN reduces resource usage coming from the CDN provider by caching requested contents and exploiting their recurrent likelihood. The inputs of the algorithm are the current segment index employed by the media player and the available representations. The result is the temporal storage of next segments to be requested beyond the current one played by the user.

Algorithm 2 Cache proxy at eNodeB

```

procedure LOCALCACHE( )
     $i_t$   $\triangleright$  listen to requests & cache segments
     $\{R_j\}$   $\triangleright$  current segment index
     $\{R_j\}$   $\triangleright$  set of alternative representations
    for all  $R_j \in \{R_j\}$  do
         $s_{i_t+1} \leftarrow$  Download( $i_t + 1, R_j$ )  $\triangleright$  download next segment
        Cache( $s_{i_t+1}, R_j$ )  $\triangleright$  from the CDN
         $\triangleright$  at eNodeB proxy

```

V. RESULTS

In order to test MEC4CDN solution, we deploy an experimental setup by exploiting a MPEG-DASH distributed dataset described by Lederer et al. [13] and provided for public experimentation of CDN-like infrastructures. This dataset consists in Red Bull Playstreet sequence stored at several geographically distributed mirror servers. The content is provided in 17 video representations encoded in H.264 Advanced Video Coding (H.264/AVC) and 4 dual channel audio representations

encoded in Advanced Audio Coding (AAC). Both audio and video are segmented with different segment lengths of 2, 4, 6, 10, and 15 seconds and multiplexed in ISO MPEG4 files (ISO/IEC 14496-12 - MPEG-4 Part 12).

Then, the overall experimental setup comprises both public network nodes and internal ones belonging to our network infrastructure:

- Three mirror servers: network nodes provided by Lederer et al. [13] which we use as CDN-like nodes for storing the media segments.
- A media server: a server in our infrastructure storing the MPD file which is requested by the client to play the content. The MPD file provides information for the player to retrieve the segments from the mirror servers. Moreover, the service employs segments with duration of 6 seconds and the video content is limited to six different representations, widely used by market services. Each representation is characterized by a specific bitrate as shown in Table I.
- A proxy server running MEC4CDN: a server in our infrastructure provided by public Internet connection and acting as a network gateway for the wireless network edge deployed inside our infrastructure. Then, all the requests from the players are processed by the proxy server before being transmitted, if necessary, to the media infrastructures on Internet. It executes the proposed MEC4CDN solution.
- A wireless access point: in order to provide wireless capabilities, an access point is used, which provides a wireless local area network (WLAN) using 2.4Ghz band. The access point is directly connected to the proxy server to provide a MEC-like architecture. The only role of the access point consists in forwarding all the incoming traffic on both directions (download and upload).
- A UE running MPEG-DASH players: a wireless network node connected to the access point, which is running GStreamer MPEG-DASH players.

The experimental setup is shown in Figure 5.

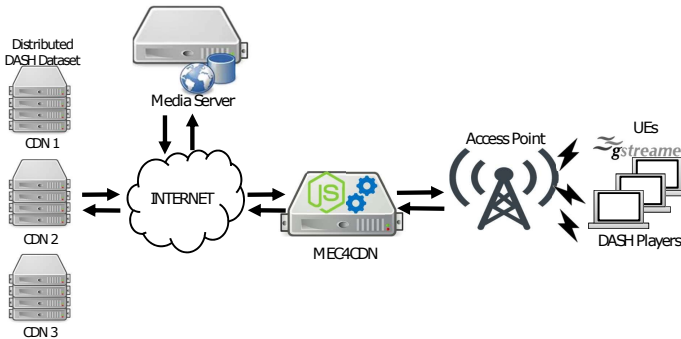


Fig. 5: Testbed.

The tests include two different experiments aiming to verify separately the two main features of MEC4CDN:

- Multi-CDN experiment: the proxy server running MEC4CDN employs a multi-CDN infrastructure which

TABLE I: Set of MPEG-DASH representations employed in the experiments.

index	bitrate	resolution	framerate
1	400kbps	480x360	30fps
2	900kbps	854x480	30fps
3	1500kbps	1280x720	30fps
4	2000kbps	1280x720	30fps
5	2500kbps	1280x720	30fps
6	3000kbps	1920x1080	30fps

allows CDN malfunction detection and switches the content download to an alternative CDN when necessary. In order to simulate CDN malfunctions, we introduce a random latency between 0 and 500 milliseconds on the wired path between the MEC4CDN proxy and the three mirror servers storing the segments.

- Local cache experiment: the proxy server running MEC4CDN empowers the content delivery by caching segments at the network edge. This enables the reduction of CDN transactions, while improving the delivery with lower experienced latency.

In both experiments, 20 players are sharing both core and network edge resources, competing for the shared wireless access point. The duration of each experiment is fixed to 10 minutes.

Moreover, to get evidence of the benefits of MEC4CDN solution, we carry out the two mentioned experiments where MEC4CDN comes into play and a baseline one without MEC4CDN. This basic setup provides a common delivery infrastructure where the network just forwards the requests to the first CDN. Here, GStreamer players are not able to identify CDN malfunctions, then they use always the same pre-set CDN even when it is suffering severe latency issues. Moreover, no cache is done at the edge, which means that CDN bandwidth is shared among the concurrent clients.

Concerning the first experiment, Figure 6 shows the representation switches along the multi-CDN experiment (10 minutes) where principal CDN suffers performance degradation. In Figure 6a the actors are not able to take decisions while in Figure 6b MEC4CDN apply strategies to switch to a healthy CDN. It is clear from the graphs that a legacy solution (6a) does not let the players to continue playing at the same representation level when a malfunction occurs while downloading the content. The players reduce the chosen representation bitrate in order to continue playing. In the case of a multi-CDN strategy (6b), some players access to higher representation bitrates since the MEC4CDN is able to enforce the delivery by switching to another CDN which better performs.

The improvements are also evident by the eMOS evaluation. The eMOS mean value among all the clients is 3.09 while downloading from a single CDN and 3.47 in case of multi-CDN delivery. This means a eMOS enhancement of +12.3%.

Regarding the second experiment, Figure 7 shows the mean value and the deviation of the measured bitrate and the

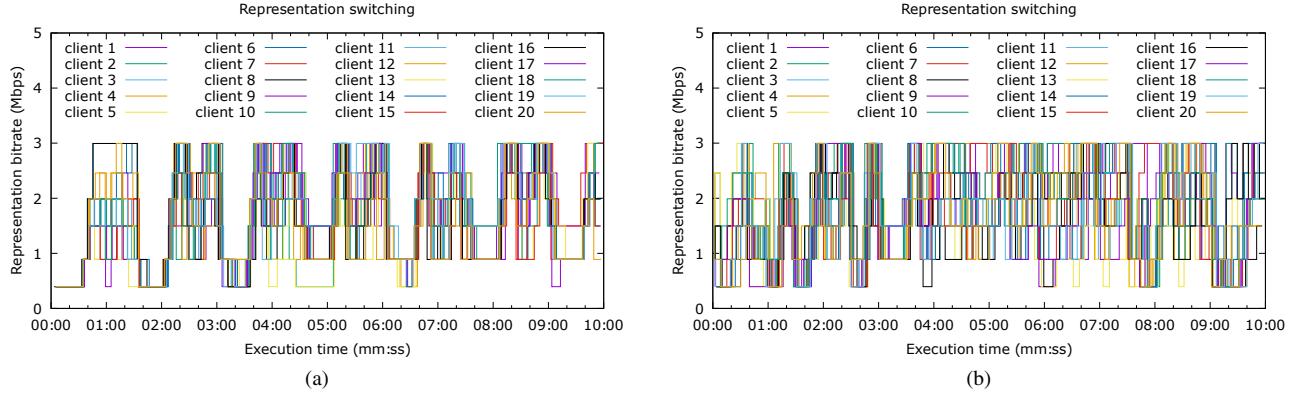


Fig. 6: Representation selection over time: legacy CDN on **a** and MEC4CDN multi-CDN on **b**.

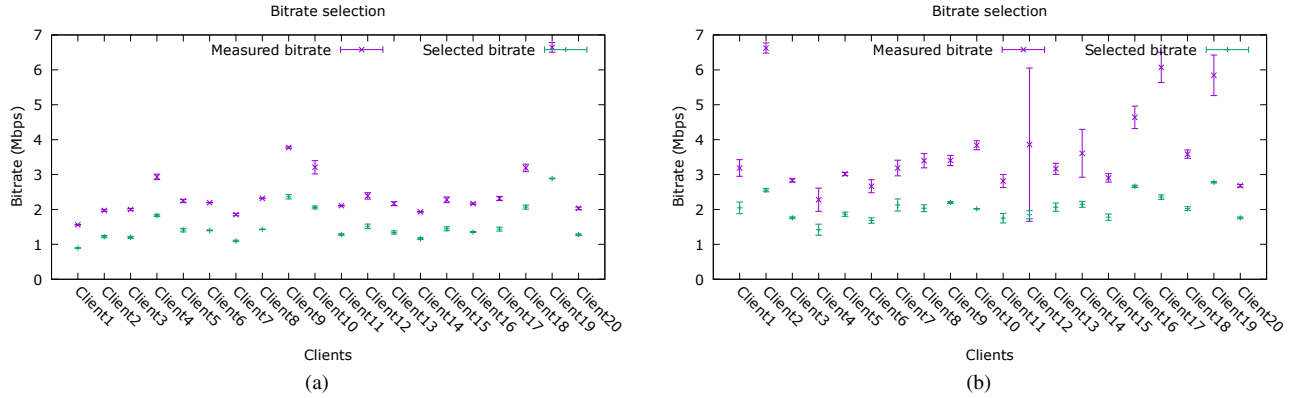


Fig. 7: Mean value and deviation of measured bitrate and selected representation bitrate: legacy content delivery on **a** and MEC4CDN cache-powered content delivery on **b**.

representation bitrate. The results for a legacy content delivery are shown in Figure 7a and, when MEC4CDN with local cache delivery is added, the results improve as depicted in Figure 7b. This is clear since the cache lets the player to experience lower latency since the segments are closer to the clients, then the throughput is higher. Therefore, the clients tend to request higher representation bitrate which improves the user's QoE.

In this case, the eMOS mean value among all the clients is 3.08 in case of legacy delivery and 3.92 in case of cache-powered delivery, which means an eMOS increase of +27.3%.

VI. CONCLUSION

This paper proposes a network proxy which enables multi-CDN video distribution and local cache at the network edge by exploiting the MEC architecture proposed by ETSI for future 5G networks.

The proposed proxy has two main outcomes. First, the proxy reduces the CAPEX of the CDN since it makes distributed cache at the network edge, then all the clients can play the contents received from the cache. Second, the proxy shields from CDN malfunction by switching the content download session to another available CDN such to keep QoE rates.

Finally, this proposal has been tested by performing two experiments on a real testbed. The first one exploiting a multi-CDN delivery and the second one employing local cache at network edge. In both the experiments, the results show that the MPEG-DASH players experience higher QoE compared to a legacy content delivery.

REFERENCES

- [1] Cisco Inc, "Visual Networking Index: Forecast and Methodology, 2016-2021," 2017. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>
- [2] Cisco Inc, "Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014-2019," 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html
- [3] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," in *IEEE MultiMedia*, vol. 18, no.4, pp. 62-67, 2011. doi:10.1109/MMUL.2011.71
- [4] A. Begen, T. Akgul and M. Baugher, "Watching Video over the Web: Part 1: Streaming Protocols," in *IEEE Internet Computing*, vol. 15, no. 2, pp. 54-63, 2011. doi:10.1109/MIC.2010.155
- [5] European Telecommunications Standards Institute (ETSI), "Developing Software for Multi-Access Edge Computing," 2017. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp20_MEC_SoftwareDevelopment_FINAL.pdf

- [6] J. Chen, M. Ammar, M. Fayed and R. Fonseca, "Client-Driven Network-level QoE fairness for Encrypted 'DASH-S'," in *Proc. of 2016 workshop on QoE-based Analysis and Management of Data Communication Networks (Internet-QoE '16)*, pp. 55-60, 2016. doi:2940136.2940144
- [7] M. Claeys, S. Latre, J. Famaey and F. De Turck, "Design and evaluation of a self-learning HTTP adaptive video streaming client," in *IEEE communications letters*, vol. 18, no. 4, pp. 716-719, 2014. doi:10.1109/LCOMM.2014.020414.132649
- [8] C.M. Lentisco, L. Bellido and E. Pastor, "QoE-Based Analysis of DASH Streaming Parameters Over Mobile Broadcast Networks," in *IEEE Access*, vol. 5, pp. 20684-20694, 2017. doi:10.1109/ACCESS.2017.2755438
- [9] V.K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner and Z.L. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *Proceedings IEEE INFOCOM 2012*, pp. 1620-1628, 2012. doi:10.1109/INFCOM.2012.6195531
- [10] V.K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z.L. Zhang, M. Varvello and M. Steiner, "Measurement Study of Netflix, Hulu, and a Tale of Three CDNs," in *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1984-1997, 2015. doi: 10.1109/TNET.2014.2354262
- [11] European Broadcasting Union (EBU), "EBU Flow Multi-CDN," 2016. [Online]. Available: https://tech.ebu.ch/docs/events/IBC16/IBC\%20Demo\%20Fact\%20Sheet_EBUFlow.pdf
- [12] Cedexis, "Multi-CDN Architecture." [Online]. Available: <https://www.cedexis.com/solutions/multi-cdn/>
- [13] S. Lederer, C. Mueller, C. Timmerer, C. Concolato, J. Le Feuvre and K. Fliegel, "Distributed DASH dataset," in *Proceedings of the 4th ACM Multimedia Systems Conference*, pp. 131-135, 2013. doi:10.1145/2483977.2483994