

DASH-BASED DEVICE-TO-DEVICE VIDEO STREAMING FOR CELLULAR NETWORKS WITH HIGH USER DENSITY

Ala'a Al-Habashna
Dept. of Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada
alaaalhabashna@sce.carleton.ca

Gabriel Wainer
Dept. of Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada
gwainer@sce.carleton.ca

ABSTRACT

Supporting video streaming services and providing high Quality of Experience (QoE) to end users have become main concerns for cellular network operators. In this work, we present an architecture for improving the QoE of video streaming in cellular networks with high user density. The architecture employs progressive caching of video contents, Dynamic Adaptive Streaming over HTTP (DASH), and Device-to-Device (D2D) communication. The Base-Station (BS) controls the progressive caching process of video contents and the Peer-to-Peer (P2P) transmission of video segments among User Equipments (UEs). We present two different implementations of the architecture. The implementations differ in the employed approach for video contents caching and distribution over the UEs. We use the Discrete Event System Specification (DEVS) formalism to build a model for the proposed architecture in an LTE-A network and use the model to study the performance achieved by the proposed architecture in terms of many video streaming QoE metrics.

Keywords: Cellular networks, Video streaming, QoE, DASH, D2D communication.

1 INTRODUCTION

Due to the increasing popularity of video streaming, the emergence of many video streaming platforms such as YouTube and Netflix, and due to the improvement of mobile devices, the demand for video streaming over mobile and cellular networks has increased exponentially during the last couple of years. A traffic forecast report has shown that video accounted for 60% of the total mobile data traffic in 2016 and by 2021, 78% of the world's mobile data traffic will be video (Cisco 2018).

This increasing demand for video streaming presents a challenge for cellular network operators, as the increase in video traffic is outpacing the improvement on the cellular network capacity. According to (Sandvine 2016), video traffic is the main reason for congestion in mobile networks. Another problem is the delay-sensitive nature of video traffic. Video contents should be delivered to users within a certain delay or users would experience high startup delay and video rebufferings, which ruin the streaming experience. Quality of Experience (QoE) is a metric that is used to measure the quality of the video streaming service as perceived by the end user, as opposed to network metrics like the download rate. There are many factors that are used to measure video streaming QoE such as the startup delay and number of video rebufferings. Due to the challenges above, providing video streaming service with high QoE to end users has become a main concern for cellular network operators. The facts above made it necessary to develop new approaches that help serving this enormous video traffic over cellular networks and improve video streaming QoE.

In (Al-Habashna et al. 2016; Al-Habashna and Wainer 2017), we proposed two algorithms for BS-controlled progressive caching of video contents and D2D video transmission in cellular networks. Device-to-Device (D2D) communication, introduced by the LTE-Advanced (LTE-A) standard, allows direct communication between devices in cellular networks (Parkvall and Astely 2009). The algorithms are called

Cached and Segmented Video Download (CSVD) and *DIStributed Cached and Segmented video download (DISCS)*. The algorithms are employed by the BS to control progressive video content caching in selected User Equipment (UEs) in the cell, referred to as Storage Members (SMs). Furthermore, the algorithms are employed by the BS to control D2D communication between UEs in the cell for Peer-to-Peer (P2P) video content distribution to requesting UEs.

In (Al-Habashna, Wainer, and Fernandes 2017), we proposed an architecture to improve the QoE of video streaming in cellular networks. The proposed architecture employs the aforementioned cached and segmented video download algorithms. The architecture also employs Dynamic Adaptive Streaming over HTTP (DASH). DASH is an adaptive video streaming technique which allows changing the video bit rate during video streaming to adapt to the available throughput (IOS 2012). The proposed architecture is called **DASH-based BS-Assisted P2P/D2D video STreaming** in cellular networks (DABAST).

In (Al-Habashna and Wainer 2017), we investigated the performance of CSVD and DISCS in terms of aggregate and average data rates. Results have shown that DISCS significantly improves the data rates over CSVD. In (Al-Habashna, Wainer, and Fernandes 2017), we studied the performance of DABAST with CSVD (DABAST-CSVD) in various scenarios. Results have shown that DABAST-CSVD achieves significant gains and improves all the measured QoE metrics. Here, we implement DABAST with DISCS (DABAST-DISCS) and investigate its performance in terms of video streaming QoE metrics. We compare the results of both DABAST-CSVD and DABAST-DISCS to see if the improvement in the data rates achieved by DISCS would translate into significant improvements in terms of video streaming QoE. We provide analysis of the results and present the new findings on DABAST with both algorithms.

We use the Discrete Event System Specification (DEVS) formalism (Wainer 2009) to build a model for DABAST-CSVD and DABAST-DISCS. DEVS provides a formal framework for modeling generic dynamic systems. It has formal specifications for defining the structure and behavior of a discrete event model. We implement our DEVS model with the CD++ toolkit (Wainer 2009) and use this implementation for performance evaluation.

The rest of this paper is organized as follows, in Section 2 we review the background and related work. In Section 3, we present CSVD, DISCS, and DABAST. We discuss the simulation scenarios and results in Section 4. Finally, we conclude the paper in Section 5.

2 BACKGROUND AND RELATED WORK

2.1 Technical background

Video streaming provides a convenient way to watch online video contents. With video streaming, a user can start playing the video before the entire video file is downloaded. Nowadays, people watch TV shows, movies, educational videos, and news clips online via video streaming. At the same time, due to the improvement on the storage, screen size, and battery of mobile phones, an increasing number of users watch such videos on their phones. According to (Sandvine 2016), during the peak hours, YouTube traffic only accounts for 27% of the mobile downlink video traffic in North America.

Over the last two decades, video streaming has been a hot topic that received much interest from academia and industry, due to its increasing popularity and bandwidth-demanding nature (Li et al. 2013). Nowadays, HTTP video streaming is the most popular form of online video streaming and it has been adopted by major video streaming solutions such as YouTube and Netflix. This is due to the convenience of using the HTTP protocol, which eliminates the need to install and use a dedicated streaming application and helps to get video traffic past firewalls (Li et al. 2013). With HTTP video streaming, the video is divided into a sequence of small segments. Each segment has a fixed duration, and the size of the segment depends on its duration and video bit rate. Video segments are progressively downloaded by the user as the video is being watched. The received segments are buffered in a video/application buffer. Video playout starts after some video content is buffered. The application that plays the video is usually referred to as the client. The streaming client requests video segments from the video server. The duration of video contents available for playout at the buffer is called the playout buffer length, measured in seconds of video. Furthermore, every second, one second of video is removed from the buffer and played to the user. If the data rate at which video

segments are transmitted is lower than the video bit rate (rate at which video is played), the playout buffer length will keep decreasing. This is because the rate at which the video is consumed is higher than the rate at which video content is transmitted (supplied). In such case, playout buffer will eventually be depleted which causes video playout interruption. This interruption is referred to as video stalling or rebuffering, as the video playout stops to buffer video contents again. This made it challenging to implement HTTP video streaming over networks with limited capacity and variable network conditions. DASH provides a promising solution to these challenges, as it allows changing the quality of video streaming to adapt to network conditions.

As an HTTP-based video streaming technique, DASH breaks down the video into small, easy to download segments. With DASH however, each one of these segments is encoded at multiple video bit rates. This feature is where the adaptability of DASH comes from. By having the segments of a video available at the server with multiple quality levels, clients can choose among the various bit rates to adapt to network conditions. This improves video streaming QoE for users as such adaptation helps avoiding video interruptions when the network conditions deteriorate. Furthermore, this improves bandwidth utilization as higher video quality can be used when the network conditions improve. As such, DASH has been employed by big video streaming platforms, such as YouTube and Netflix, due to these advantages it has over classical HTTP video streaming.

As video streaming is becoming the most popular application over cellular networks, providing reliable video streaming service is a key to differentiated competitiveness. As such, it is important to evaluate the quality of the video streaming service as perceived by the end user. Hence, quality measure has shifted from Quality of Service (QoS) to QoE. There are many factors that are used to measure video streaming QoE. The number of video rebufferings is an important factor. Increasing the number of rebufferings significantly reduces video streaming QoE, and hence, it should be avoided as much as possible. Initial delay is another factor that also reduces video streaming QoE. Video bit rate is a measurement of the amount of data in one second of the video. As the video bit rate increases, the video quality increases, which increases the QoE.

2.2 Related work

D2D communication allows two UEs that are within proximity of each other to communicate without routing the traffic through the BS and the network infrastructure. This direct communication between UEs can be utilized to improve the capacity of cellular networks and improve the achieved data rates. D2D communication can also be utilized to extend the cell coverage and improve the service for users at the cell-edge. As such, D2D communication is one of the main technologies in the Fifth Generation (5G) cellular networks (Agiwal, Roy, and Saxena 2016). Much research has been conducted in recent years on D2D communication and developing applications for it in cellular networks (Asadi, Wang, and Mancuso 2014).

In (Zhang et al. 2015; L. Duan, Gao, and Huang 2014), the authors propose many incentive mechanisms to motivate involvement of UEs in D2D communication. This is important because the success of D2D communication depends on the participation of users to share the content they have. We do not consider incentive mechanisms here, as this is a different research area that is out of the scope of this paper.

Some of the work in the literature is focused on caching data contents in smart devices for D2D communication. In (Kang et al. 2014), mobile content delivery networks are proposed where special mobile devices are placed in the network as caching servers. Popular contents are cached in such servers and can be provided to users in the vicinity over D2D communication. While this technology could help improving the data rates in cellular networks, it is costly as these designated devices need to be placed throughout the network, configured, and maintained. In (Golrezaei, Dimakis, and Molisch 2012; Golrezaei et al. 2014), an architecture is proposed to improve the throughput of video transmission and overcome the problem of rapidly increasing wireless video traffic. In this architecture, UEs in the cell can save video files. When a video file is requested by a UE, it will be transmitted from the UE that has the file over a D2D link if the file is cached. The work above either focus on caching polices for video/data contents assuming that caching of such contents takes place during low traffic load or focus on the impact of D2D communication assuming that data is already cached. Under high traffic load, it would be highly inefficient to transmit unrequested

video contents for caching under the assumption it might be requested in the future, as this would waste precious cellular resources. Furthermore, the popularity of contents might not be known for the network. Moreover, mobile devices are usually not stationary for extended periods of time to be treated as caching servers (content is solely transmitted for future utilization). As such, in (Al-Habashna et al. 2016; Al-Habashna and Wainer 2017), we proposed cached and segmented video download algorithms for progressive caching and distribution of video segments among UEs in the cell. These algorithms operate under high traffic load and high user density. Hence, video files are cached as requested. The algorithms define how video files are cached and exchanged among UEs and protocols have been defined for this purpose. Only selected UEs are used for video caching to reduce interference when transmitting video content among users.

Research has also been conducted on utilizing D2D communication to provide P2P video streaming in cellular networks. In (Siris and Dimopoulos 2015), multi-source video streaming was proposed. In this system users can connect through WiFi direct to exchange video contents. Such system requires each device to perform device discovery to find neighbors, and service discovery to find services offered by neighboring device. These requirements along with the signaling needed to exchange contents consume significant amount of resources.

RapidStream is a protocol for P2P video streaming on mobile phones that was proposed in (Eittenberger, Herbst, and Krieger 2012). RapidStream emulates P2P streaming protocols on wired networks that involve the dissemination of buffer maps and request messages between peers. Although such protocols are suitable for wired networks, they are not as scalable in cellular networks as they involve too much signaling (dissemination of buffer maps) for UEs that has limited power, processing, and transmission resources.

In (Duong et al. 2015) the authors proposed a D2D communication system where multiple helpers collaborate to send a video segment to the requesting UE. The video, which is assumed to be in scalable video coding standard, is encoded by applying multiple description coding by each helper, and each helper sends a different description to the requesting UE. The authors analytically studied the problem of optimizing the number of transmitted descriptions to the requesting UE to maximize the video quality and efficiently consume the helpers' energy. However, the work only considers the energy consumed by the helpers to send the segments without considering the processing power and energy needed to encode the video segments. Encoding the video segments is a big favor to ask for, considering the limited energy and processing power of UEs. Furthermore, the work does not consider the most important QoE metrics such as rebufferings.

In (Le et al. 2016), a system, called MicroCast, was designed and evaluated using a testbed. MicroCast is designed for a small group of smart phone users (a group of friends) who are within proximity of each other and interested in watching the same video at the same time. In this system, the cellular connection is employed to download segments of the video and the WiFi connections is employed to share among users the downloaded contents to improve users' experience. However, the scope of this system is limited as it is designed for a small group of users watching a video simultaneously.

Most of the work above on P2P video streaming in cellular networks evaluate the improvement achieved by employing D2D communication considering that popular contents are already cached. With DABAST, we provide a framework that takes care of video content caching and distribution. Furthermore, all the work above on P2P video streaming consider small-scale networks, i.e., up to 10 UEs including helpers. DABAST is designed for high group of users (high user density). Results show that using clustering and BS assistance, the potential of collaborative D2D communication between UEs is significant. In this paper, we study the performance of DABAST-DISCS and investigate its performance in terms of video streaming QoE metrics. We compare the results of both DABAST-CSVD and DABAST-DISCS to see if the improvement in the data rates achieved by DISCS would translate into significant improvements in terms of video streaming QoE.

3 THE DABAST ARCHITECTURE

In this section, we discuss DABAST and its operation. Thereafter, we present the employed cached and segmented video download algorithms.

3.1 DABAST

By employing the cached and segmented video download algorithms (Al-Habashna and Wainer 2017), DABAST relaxes the bottleneck of the Radio Access Network (RAN), and hence, improves video streaming QoE for end users. DABAST is designed for LTE-A networks that provide the infrastructure for communication between the BS and UEs over cellular links and direct communication between UEs over D2D links. Over such infrastructure, CSVD and DISCS are used to provide BS-assisted content caching and D2D communication. This allows P2P video content exchange between users in the cell. Video segments can be cached at various video bit rates and DASH-based video streaming is used to provide users with video segments over the cellular channel or the D2D channel when the requested segment is cached.

Figure 1 depicts the implementation of DABAST in cellular networks. A CSVD server/proxy is used in the RAN at the BS. This provides the processing and networking capabilities needed to implement CSVD or DISCS. The CSVD server can also be used to provide caching capabilities to store popular files at the BS. Streaming clients at the UEs send their requests asking for video segments. These requests are processed by the CSVD server. Based on the employed algorithm (CSVD or DISCS), the server decides whether to send the segment from the distributed cache over the D2D channel or get it from the content server and send it over the cellular channel. If the video segment is to be delivered from the distributed cache, an assistance request will be sent to an SM. Otherwise, the request will be forwarded to the DASH server. Under high traffic load, the BS sends a video segment from the distributed cache (when found) even if the segment found in the distributed cache does not match the video bit rate requested by the UE. This is to maximize the exploitation of the distributed cache and D2D channel. Another feature of DABAST is that it can operate in proactive mode. In this mode, DABAST can send up to a certain number of video segments to the user when found in the distributed cache before the segments are requested. This reduces the signaling and latency between the BS and the UE and speeds up transmission of video segments. However, if the video segment is not available in the distributed cache, the request will be awaited.

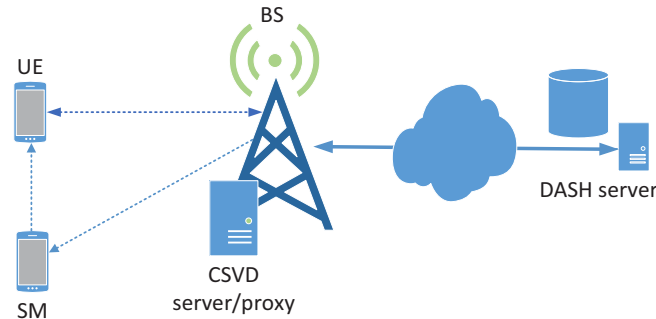


Figure 1: Illustration of the implementation of DABAST.

A video bit rate adaptation strategy is used by DASH to determine how the client selects the streaming quality to adapt to the varying network conditions. Many video bit rate adaptation strategies are proposed for DASH (De Cicco et al. 2013; Huang et al. 2014). These strategies usually try to balance between two factors. They try to maximize the video quality by selecting the highest video rate the network can support, and at the same time minimize rebufferings. We refer to the component in the client that runs the adaptation strategy as the DASH controller. We adopt the buffer-based approach in (Huang et al. 2014). This is because in our architecture, the UE could receive a video segment from the BS or from any SM in the cluster. As such, it would be difficult to estimate the throughput at which the next segment will be received. The adaptation algorithm used is a piecewise function, $f(B)$, that uses the length of the playout buffer, B , to determine the video bit rate (Huang et al. 2014).

3.2 The CSVD and DISCS algorithms

In (Al-Habashna, Fernandes, and Wainer 2018), we show that high traffic load and resource shortage at the RAN could significantly degrade video streaming QoE. CSVD and DISCS relax the RAN bottleneck by

providing BS-assisted progressive video content caching and P2P video segment distribution in cellular networks. When we use the term P2P here, we refer to direct transmission of video segments between UEs in the cell over D2D links.

With both algorithms, the cell is divided into clusters. To do that, the coverage area of the cell is divided into non-overlapping subareas by the BS. Each one of these subareas will be a cluster. The BS assigns UEs to clusters based on their locations, and it selects the UEs in the central area of each cluster as SMs of that cluster. SMs are UEs that are used as helpers in the cluster. To prevent inter-cluster as well as inter-cell interference, only the UEs in the central area of each cluster are selected as SMs.

After clustering, when a UE requests a video file from the BS, the BS processes the request and responds as follows:

- **Send With Assistance (SWA):** if the video (or parts of it) is available in any of the SMs of the cluster, the BS will ask the SMs to send the video segments to the requesting UE over D2D links.
- **Send To an SM (STSM):** if the requested video is not available in the distributed cache of the cluster, and the requesting UE is an SM, the BS will send the video to that UE over a cellular link, and it will ask the UE to cache it. This case allows the SMs to cache video files. These files will be available for UEs in the cluster when requested later.
- **Distribute to SMs (DTSMs):** this case is only used in DISCS. In this case, if a requested video is popular and it is not available in the distributed cache of the cluster, the BS will distribute the segments of the video among the SMs. The BS asks the SMs to cache the pieces (as the video is popular) and forward the received pieces to the requesting UE.
- **Send To a UE (STUE):** otherwise, the BS will send the video directly to the requesting UE over a cellular link.

The difference between CSVD and DISCS is the DTSMs case. The DTSMs case is only used in DISCS. The other 3 cases are used by both CSVD and DISCS. For further detail on the operation of CSVD and DISCS, the reader is referred to (Al-Habashna and Wainer 2017).

4 SIMULATION SCENARIOS AND RESULTS

We used the DEVS formalism (Wainer 2009) to build a model for DABAST-CSVD and DABAST-DISCS. We implemented our DEVS model with the CD++ toolkit (Wainer 2009) and used this implementation for performance evaluation. Due to lack of space, we do not show our DEVS model here. For detailed description of the developed DEVS model and the used simulation setup, the reader is referred to (Al-Habashna, Wainer, and Fernandes 2017). We executed simulations to evaluate the performance of both DABAST-CSVD and DABAST-DISCS in terms of many video streaming QoE metrics. The urban macro propagation model (3GPP 2015) was used for cellular links with a DL operating carrier frequency of 900 MHz, and a transmission bandwidth of 10 MHz. The D2D channel model at 24 GHz is used for D2D transmission (Al-Hourani, Chandrasekharan, and Kandeepan 2014).

A single LTE-A cell with 500 UEs is considered in the simulations. The UEs are randomly distributed throughout the cell with a uniform distribution. In each iteration of the simulations, the UEs are first distributed through the cell and clustering is performed dividing the cell into 9 clusters. After clustering, the UEs request video streams. During each iteration of the simulation, each UE requests two video streams. A UE requests a video stream, and after finishing video playout, it requests a second video. Before generating each request, a UE waits for a random period according to a Poisson distribution with mean of 10 seconds. The Zipf distribution is used to model the relative popularity of videos (Cha et al. 2007). Using this distribution, some videos are requested more often than others. Four video bit rate levels were used as shown in Table 1. These are adapted from the H.264/AVC video coding standard (ITU-T 2012). The length of the videos is 441 seconds, which is the mean length of a YouTube video (Ahsan, Singh, and Ott 2014).

As mentioned in the previous section, we use the buffer-based video bit rate adaption method in (Huang et al. 2014). The playout buffer length and video bit rate mapping used by the DASH controller is shown in Table 1.

We measured the number of rebufferings, initial delay, and video bit rate levels of the received video segments. In addition to these metrics, we show the results for the video continuity index, which is a measure of the extent by which rebuffering pauses are avoided (De Cicco and Mascolo 2014). The continuity index is calculated as follows,

$$\eta_c = 1 - \frac{\Delta T_{rb}}{\Delta T}, \quad (1)$$

where ΔT_{rb} is the total time the client remains paused due to rebuffering events and ΔT is the duration of the experiment (playing time and rebuffering time). Table 2 shows the mean values for these measurements.

The results in Table 2 are for 500 UEs in the cell, Zipf exponent of 1.5, and 500 videos. The average value for each simulation run was calculated. The values below show the mean of all the average values from 50 simulation runs. In addition to the mean, we show the Margin of Error (MoE) for 95% confidence interval.

Table 1: Playout buffer length-video rate mapping.

Playout buffer length (s)	Video bit rate (kbps)
$0 \leq B \leq 90$	384
$90 < B \leq 150$	768
$150 < B \leq 200$	2000
$200 < B$	4000

Before discussing the results in Table 2 for the measured QoE metrics, it is worth mentioning that the average data rates achieved with DABAST-CSVD and DABAST-DISCS are 3.89 and 8.32 Mbps, respectively. As discussed in (Al-Habashna and Wainer 2017), DISCS significantly improves the achieved average data rate because it speeds up video caching which increases the percentage of requests that are satisfied from the distributed cache which speeds up the transmission. This is also because in the case of DISCS, many files will be sent in parallel from multiple SMs (as opposed to one SM), thanks to the DTSMs case which distributes segments of a cached video file among multiple SMs. This causes further parallelism in sending video files and better load balancing among SMs, which speeds up the transmission of video files and increases the average data rate. From Table 2, we can see that as expected, DABAST-DISCS provides improvement over DABAST-CSVD in terms of the measured QoE metrics. Table 2 shows that DABAST-DISCS reduced the average number of rebufferings from 1.73 to 1.63, which increased the continuity index. With DABAST-DISCS, the initial delay is also reduced from 28.7 seconds to 21.2 seconds, which is a significant improvement. Furthermore, the average video bit rate increased by 18 kbps with DABAST-DISCS.

Table 2: Simulation results.

	DABAST-CSVD		DABAST-DISCS	
	Mean	MoE	Mean	MoE
Rebufferings	1.7272	0.0164	1.6294	0.0169
Cont. index	0.8699	0.0001	0.8763	0.0011
Initial delay(sec)	28.654	0.4821	21.192	0.4163
Video bit rate (kbps)	430.16	1.3694	448.57	0.9607

Although the results above show that DABAST-DISCS improves the QoE metrics, one can see that the only significant improvement achieved by DABAST-DISCS is in terms of the initial delay. Only slight improvement is achieved in terms of the average number of rebuffering and continuity index. One would expect higher gains by DABAST-DISCS over DABAST-CSVD given the that DISCS achieves more than double the average data rate obtained with CSVD. In the following, we present further analysis of the above results to understand this behavior.

Figure 2 depicts the relative frequency histogram for the number of rebufferings of DABAST-CSVD and DABAST-DISCS. From the figure, one can see that the main difference is that with DABAST-DISCS, higher number of video streams have 3 rebufferings and fewer number of video streams have 4 rebufferings. This explains why the average number of rebufferings with DABAST-DISCS is less than that with DABAST-CSVD. The figure shows that with both DABAST-CSVD and DABAST-DISCS, 50% of the video streams have 0 rebufferings. With DABAST-CSVD, 27.3% of the video streams have 3 rebufferings, and 22.7% of the streams have 4 rebufferings. With DABAST-DISCS, 36.0% of the video streams have 3 rebufferings, and 13.3% of the streams have 4 rebufferings. This improvement in the number of rebufferings is expected, as DABAST speeds up video caching, and improves the rate at which video segments are delivered to requesting UEs. Figure 2 also shows that with DABAST-DISCS, a very small percentage of the video streams (0.1%) have 5 rebufferings. This increase in the number of rebufferings experienced by a slight percentage of the streams is a result of the DTSMs case, where video segments are sent to the requesting UEs in two steps, i.e., the segment is sent to the SM first, and then sent to the UE by the SM. Despite of this increase to a very small percentage of the UEs, DABAST-DISCS still achieves lower initial delay and number of rebufferings, on average. This means that the DTSMs case is beneficial to the cell, as expected.

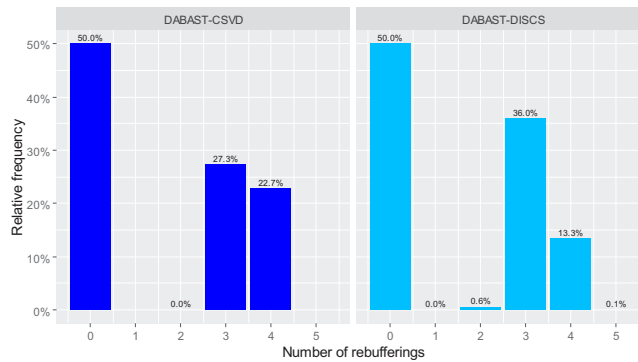


Figure 2: Relative frequency histogram of the number of rebufferings for DABAST-CSVD and DABAST-DISCS.

Although the results show that DABAST-DISCS decreases the number of rebufferings, one can argue that DABAST-DISCS is expected to achieve higher gains in terms of the number of rebufferings, as it significantly improves the average data rate. To further investigate the results and explain this behavior, we separate the results for each request. As previously mentioned, each UE in the simulations makes 2 video streaming requests. After playout of the first video, a UE would stay idle for a random period of time, and then generates another request for a video stream.

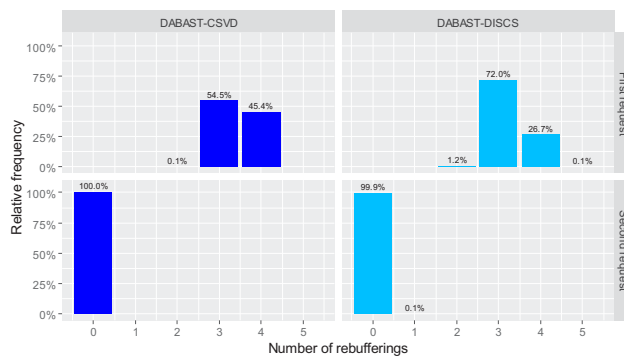


Figure 3: Relative frequency histogram of the number of rebufferings in each request for DABAST-CSVD and DABAST-DISCS.

Figure 3 shows the relative frequency histogram for the number of rebufferings of each request for both DABAST-CSVD and DABAST-DISCS. As with the previous figure, the histograms on the right side are for DABAST-CSVD, and the ones on the left are for DABAST-DISCS. In this figure, however, the histograms on the top are for the first requests, while the ones at the bottom are for the second requests. Figure 3 shows that for DABAST-CSVD, all the rebufferings take place during the first set of video streams. All the second video streams have 0 rebufferings. This is because by the time most of the video streams start, there are no video segments cached in the distributed caches. Hence, most of the video segments will be delivered over the cellular channel. As such, the limited cellular channel will be shared by the large number of users, which means the average data rate at which these segments are delivered is low and explains the high number of rebufferings. The figure shows that with DABAST-CSVD, all the second set of streams have 0 rebufferings. By the time the second set of streams starts, there will be many video segments cached in the clusters. Hence, many of the segments will be delivered over D2D links, which eliminates rebufferings for those streams. This also relaxes the bottleneck of the RAN because, at this time, only a portion of the video segments will be sent over the cellular channel. Because the cellular channel is now shared by a much lower number of UEs, the data rate will increase, and rebufferings will be avoided for these video streams as well.

With DABAST-DISCS, almost all rebufferings occur in the first set of video streams, as 99.9% of the second set of video streams have 0 rebufferings, and only 0.1% of the second set of video streams have 1 rebufferings. This is for the same reason all the rebufferings with DABAST-CSVD occur during the first set of video streams. Initially, there are no video segments available in the distributed cache, and hence, all video streams will experience multiple rebufferings. By the time the second set of streams starts, there will be many video segments cached in the clusters. Hence, many of the segments will be delivered over D2D links, which eliminates rebufferings for these streams, and relaxes the RAN bottleneck for segments delivered over cellular resources. The one difference here is that there is a very small portion of the second set of video streams that still get its segments with the DTSMs case, which causes 1 rebuffering to 0.1% for the second set of video streams.

The results in Figure 3 also explain why DABAST-DISCS does not achieve very significant improvement in terms of the average number of rebufferings over DABAST-CSVD. As discussed above, DISCS achieves significant improvement in terms of the average data rate over CSVD. This is because in the case of DISCS, many files will be sent in parallel from multiple SMs (as opposed to one SM). This causes further parallelism in sending video segments and better load balancing between SMs, which speeds up the transmission of video files and increases the average data rate. However, we have seen from Figure 2 that video streams with segments delivered over D2D links already have 0 rebufferings, even in the case of DABAST-CSVD. As such, the increase in the average data rate achieved by DABAST-DISCS will not translate into reduction in the average number of rebufferings, as all rebufferings take place in the first set of streams when video segments are not cached. This means that the improvement in the average number of rebufferings gained by DABAST-DISCS over DABAST-CSVD is due to the fact that DISCS speeds up video caching and achieves better hit ratio, which increases the percentage of requests that are satisfied from the cluster's cache and speeds up the relaxation of the RAN bottleneck.

Figure 4 shows the histogram of the continuity-index for both DABAST-CSVD and DABAST-DISCS. As expected, 50% of the streams have a continuity index of 1. With DABAST-DISCS there is more concentration of the values around 0.76 and less concentration of values between 0.71 and 0.75. This agrees with the results for the number of rebufferings. However, with DABAST-DISCS, there are few continuity index values less than 0.7. These values correspond to streams with 5 rebufferings.

Regarding the average video bit rate, we can also see that DABAST-DISCS did not achieve a significant improvement over DABAST-CSVD (only 4.3% improvement). This can be explained as follows. As most of the cached video segments are downloaded during high traffic load. These segments are usually downloaded with low video bit rate. By the time the second set of video streams starts, there will be many segments available in the clusters' caches. However, most of these segments have low video bit rate. As DABAST sends available segments from the clusters' caches over D2D links (in the case an SM is available), most of the segments transmitted over the D2D channel will be sent from the distributed cache with low video bit rate. Although these segments are transmitted with higher data rates in the case of

DABAST-DISCS, this does not increase the video bit rate for these segments. DABAST operates in this fashion to save the valuable cellular resources and exploit them for sending video segments that are not available in the clusters' caches to avoid rebufferings as much as possible.

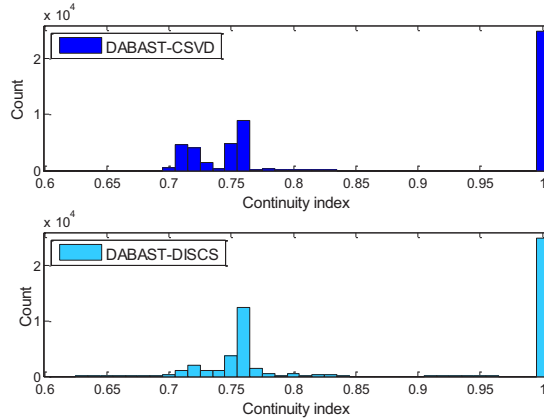


Figure 4: Histogram of the continuity-index (DABAST-CSVD and DABAST-DISCS).

A cached video segment is sent over the cellular channel only when there are no SMs available to send the segment. These video segments that are sent over cellular resources (despite being cached) will usually be sent with high video bit rate as such segments are usually requested with high video bit rate. This is because streams with cached video segments usually have long playout buffer length, as most of their segments are sent over the D2D channel with higher data rates. Furthermore, such video streams have even longer playout buffer length in the case of DABAST-DISCS, as video segments are sent with higher data rate than these with DABAST-CSVD. Because of that, video segments that are sent over cellular resources (in spite of being cached) will be sent with higher video bit rate in the case of DABAST-DISCS when compared to DABAST-CSVD. This explains the small improvement achieved by DABAST-DISCS over DABAST-CSVD in terms of the average video bit rate.

This behavior, explained above, can be seen in Table 3, which shows the number of video segments received with each video bit rate, for DABAST-CSVD and DABAST-DISCS. Table 3 shows that in the case of DABAST-DISCS, fewer segments are received with video bit rate of 768 kbps, and more segments (about the double) are received with 2Mbps and 4Mbps video bit rates, when compared to DABAST-CSVD. While this is beneficial for the streams that receive video segments with high video bit rates, it increases the RAN bottleneck and decreases the average data rate for other UEs receiving video segments exclusively over cellular resources. In future work, we will consider approaches for resource management in DABAST that help avoiding such unfairness in resource allocation.

Table 3: Count of the received segments with each video bit rate.

Video bit rate (kbps)	Count	
	DABAST-CSVD	DABAST-DISCS
384	2077111	2089649
768	149365	105874
2000	19273	46179
4000	4251	8298

5 CONCLUSION

In this paper, we present an architecture to improve video streaming Quality of Experience (QoE) in cellular networks with high user density. The proposed architecture employs progressive caching of video contents,

Dynamic Adaptive Streaming over HTTP (DASH), and Device-to-Device (D2D) communication to provide Peer-to-Peer (P2P) video content distribution. The architecture is called **DASH-based BS-Assisted P2P/D2D video STreaming in cellular networks (DABAST)**. The Base-Station (BS) controls the progressive caching process of video contents and the Peer-to-Peer (P2P) transmission of video segments among User Equipments (UEs). We present two different implementations of the architecture. The implementations differ in the employed algorithm for video contents caching and distribution over the UEs. One implementation employs the Cached and Segmented Video Download (CSVD) algorithm while the other employs DIStributed Cached and Segmented video download (DISCS). We use the Discrete Event System Specification (DEVS) formalism to build a model for both implementations of DABAST in an LTE-A network. We use the model to study the performance achieved by both DABAST-CSVD and DABAST-DISCS in terms of many video streaming QoE metrics. Results show that DABAST-DISCS achieves improvements over DABAST-CSVD in terms of the measured video streaming QoE metrics. We provide analysis of the results and present the new findings on DABAST with both algorithms.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Stenio Fernandes from the Federal University of Pernambuco, Brazil, for his valuable assistance during this work.

REFERENCES

- 3GPP. 2015. *Evolved Universal Terrestrial Radio Access; RF System Scenarios*. Technical report TR36.942.
- Agiwal, M., A. Roy, and N. Saxena. 2016. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 18, no. 3 (February): 1617–1655.
- Ahsan, S., V. Singh, and J. Ott. 2014. Characterizing Internet Video for Large-Scale Active Measurements. *arXiv Preprint arXiv:1408.5777v1* (August 7).
- Al-Habashna, A., S. Fernandes, and G. Wainer. 2018. Analyzing the Effect of LTE-A Transmission Parameters on Video Streaming Quality of Experience. In *Proceedings of the 21st Communications and Networking Symposium*, 236–247. Baltimore, USA: Society for Computer Simulation International.
- Al-Habashna, A., and G. Wainer. 2017. Improving Video Transmission in Cellular Networks with Cached and Segmented Video Download Algorithms. *Mobile Networks and Applications* (September 13): 1–17.
- Al-Habashna, A., G. Wainer, G. Boudreau, and R. Casselman. 2016. Distributed Cached and Segmented Video Download for Video Transmission in Cellular Networks. In *2016 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, 473–480. Montreal, Canada: IEEE.
- Al-Habashna, A., G. Wainer, and S. Fernandes. 2017. Improving Video Streaming over Cellular Networks with DASH-Based Device-to-Device Streaming. In *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, 468–475. Seattle, USA: IEEE.
- Al-Hourani, A., S. Chandrasekharan, and S. Kandeepan. 2014. Path Loss Study for Millimeter Wave Device-to-Device Communications in Urban Environment. In *2014 IEEE International Conference on Communications Workshops*, 102–107. Sydney, Australia: IEEE.
- Asadi, A., Q. Wang, and V. Mancuso. 2014. A Survey on Device-to-Device Communication in Cellular Networks. *IEEE Communications Surveys & Tutorials* 16, no. 4 (November): 1801–1819.
- Cha, M., H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. 2007. I Tube, You Tube, Everybody Tubes. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, 1–14. New York, USA: ACM Press.
- De Cicco, L., V. Caldaralo, V. Palmisano, and S. Mascolo. 2013. ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In *20th International Packet Video Workshop*, 1–8. San Jose, USA: IEEE.
- De Cicco, L., and S. Mascolo. 2014. An Adaptive Video Streaming Control System: Modeling, Validation, and Performance Evaluation. *IEEE/ACM Transactions on Networking* 22, no. 2 (April): 526–539.
- Cisco. 2018. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update. Accessed February 28. <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>.

- Duong, T.Q., N.-S. Vo, T.-H. Nguyen, M. Guizani, and L. Shu. 2015. Energy-Aware Rate and Description Allocation Optimized Video Streaming for Mobile D2D Communications. In *2015 IEEE International Conference on Communications*, 6791–6796. London, UK: IEEE.
- Eittenberger, P.M., M. Herbst, and U.R. Krieger. 2012. RapidStream: P2P Streaming on Android. In *2012 19th International Packet Video Workshop*, 125–130. Munich, Germany: IEEE.
- Golrezaei, N., A.G. Dimakis, and A.F. Molisch. 2012. Wireless Device-to-Device Communications with Distributed Caching. *arXiv Preprint arXiv:1205.7044* (May 31).
- Golrezaei, N., P. Mansourifard, A.F. Molisch, and A.G. Dimakis. 2014. Base-Station Assisted Device-to-Device Communications for High-Throughput Wireless Video Networks. *IEEE Transactions on Wireless Communications* 13, no. 7 (July): 3665–3676.
- Huang, T.-Y., R. Johari, N. McKeown, M. Trunnell, M. Watson, T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. 2014. A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. In *Proceedings of the 2014 ACM SIGCOMM*, 187–198. New York, USA: ACM.
- IOS. 2012. *Information technology—Dynamic Adaptive Streaming over HTTP (DASH)—Part 1: Media Presentation Description and Segment Formats*. ISO/IEC 23009-1:2012.
- ITU-T. 2012. *Infrastructure of Audiovisual Services coding of Moving Video*. ITU-T Recommendation H.264.
- Kang, H.J., K.Y. Park, K. Cho, and C.G. Kang. 2014. Mobile Caching Policies for Device-to-Device (D2D) Content Delivery Networking. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, 299–304. Toronto, Canada: IEEE.
- L. Duan, L. Gao, and J. Huang. 2014. Cooperative Spectrum Sharing: A Contract-Based Approach. *IEEE Transactions on Mobile Computing* 13, no. 1 (January): 174–187.
- Le, A., L. Keller, H. Seferoglu, B. Cici, C. Fragouli, and A. Markopoulou. 2016. MicroCast: Cooperative Video Streaming Using Cellular and Local Connections. *IEEE/ACM Transactions on Networking* 24, no. 5 (October): 2983–2999.
- Li, B., Z. Wang, J. Liu, and W. Zhu. 2013. Two Decades of Internet Video Streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications* 9, no. 1 (October 1): 1–20.
- Parkvall, S., and D. Astely. 2009. The Evolution of LTE towards IMT-Advanced. *Journal of Communications* 4, no. 3 (April): 146–154.
- Sandvine. 2016. *2016 Global Internet Phenomena: Latin America and North America*. Technical report.
- Siris, V.A., and D. Dimopoulos. 2015. Multi-Source Mobile Video Streaming with Proactive Caching and D2D Communication. In *IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks*, 1–6. Boston, USA: IEEE.
- Wainer, G.A. 2009. *Discrete-Event Modeling and Simulation: A Practitioner's Approach*. Boca Raton: CRC Press.
- Zhang, Y., L. Song, W. Saad, Z. Dawy, and Z. Han. 2015. Contract-Based Incentive Mechanisms for Device-to-Device Communications in Cellular Networks. *IEEE Journal on Selected Areas in Communications* 33, no. 10 (October): 2144–2155.

AUTHOR BIOGRAPHIES

ALA'A AL-HABASHNA is a PhD candidate in the Department of Systems and Computer Engineering at Carleton University, Ottawa, ON, Canada. His email address is alaaalhabashna@sce.carleton.ca.

GABRIEL WAINER is a Professor at the Department of Systems and Computer Engineering at Carleton University. He is a Fellow of SCS. His email address is gwainer@sce.carleton.ca.