Improving Video Streaming over Cellular Networks with DASH-based Device-to-Device Streaming

Ala'a Al-Habashna[1], Gabriel Wainer[1], Stenio Fernandes[2,1]
[1]Dept. of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada
[2]Center of Informatics, Federal University of Pernambuco, Recife, Brazil
alaaalhabashna@sce.carleton.ca, gwainer@sce.carleton.ca, sflf@cin.ufpe.br

***Abstract*—With the continuously increasing popularity of video streaming applications, providing video streaming services with high Quality of Experience (QoE) is becoming a main concern for cellular network operators. This paper presents and investigates the performance of an architecture for improving the QoE of video streaming in cellular networks. The architecture provides Dynamic Adaptive Streaming over HTTP (DASH) - based Peer-to-Peer (P2P) video streaming in cellular networks. The architecture employs Base-Station (BS) assisted Device-to-Device (D2D) video transmission in cellular networks for direct exchange of video contents among users. We evaluate the performance of the proposed architecture in terms of many video streaming QoE metrics. Furthermore, we investigate and provide a thorough analysis of the performance of the proposed architecture under various scenarios.**

Keywords—Cellular networks; D2D communication; Video streaming; QoE; DASH

I. INTRODUCTION

With the advancement of smart devices and the evolution of cellular networks, video streaming service has become very popular. Video content is the main contributor to data traffic over cellular networks nowadays. According to [1], video traffic accounted for 60% of total mobile data traffic in 2016. Furthermore, over three-fourths (78%) of the world's mobile data traffic is expected to be video by 2021. Another reason for this increasing popularity of video streaming is the emergence of new platforms for video streaming such as YouTube and Netflix. As such, providing high Quality of Experience (QoE) video streaming services has become a main concern for cellular networks operators.

Due to the limited capacity of cellular networks, it is difficult to provide the users with the data rates needed to achieve high QoE video streaming, especially in the case of high user density. The continuous increase in resource-demanding video applications is outpacing the improvements on the cellular network capacity. As per [2], video traffic is the main reason for congestion in mobile networks. The facts above made it necessary to develop new approaches that help serving the increasing video traffic over cellular networks, and improve the QoE of video streaming.

Device-to-Device (D2D) communication, introduced by the LTE-Advanced (LTE-A) standard, allows direct communication between devices in cellular networks [3]. In our previous work [4], we proposed an architecture that improves the QoE of video streaming in cellular networks. The proposed architecture employs Cached and Segmented Video Download (CSVD) [5-6]; an algorithm for Base-Station (BS) assisted D2D video transmission in cellular networks. CSVD manages video content caching in selected User Equipments (UEs) in the cell, referred to as Storage Members (SMs). Furthermore, CSVD is employed by the BS to use D2D communication between UEs in the cell for Peer-to-Peer (P2P) video content distribution to requesting UEs. When we use the term P2P transmission here, we refer to direct transmission of video content between UEs in the cell over D2D links. The proposed architecture is called **DA**SH-based **BS** **A**ssisted P2P/D2D video **ST**reaming in cellular networks (DABAST).

The proposed architecture also supports 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. This increases the bandwidth utilization and improves the quality of video streaming as perceived by the end user. As such, DASH is employed by the proposed architecture due to its advantages and to allow support for DASH-based applications.

With DABAST, employing CSVD helps relaxing the Radio Access Network (RAN) bottleneck, which is the main bottleneck in cellular networks with limited frequency resources that are shared among a large number of users. In [4], we investigated the performance of DABAST in a single simulation setup with high user density cellular network. Results have shown that DABAST achieves performance improvements in terms of the studied QoE metrics.

Here, we provide more details on the operation of DABAST, and investigate the performance of DABAST in various scenarios, to analyze the impact of various factors on the performance improvements achieved by DABAST. We use the Discrete EVent System Specification (DEVS) formalism to build a model for the proposed architecture in an LTE-A network. Simulations based on this model are used to evaluate the performance of DABAST in terms of video streaming QoE metrics in various scenarios. We provide analysis of the results and present the new findings on the performance of DABAST.

The rest of this paper is organized as follows, Section 2 provides an overview on the background and the related work. Section 3 presents the different aspects of DABAST and its operation. Section 4 illustrates modeling DABAST with DEVS. The simulation scenarios and results are discussed in Section 5. Finally, Section 6 states the conclusions and future work.

SummerSim-SPECTS, 2017 July 9-12, Bellevue, Washington, USA;

II. Background and Related Work

A. Technical background

Video streaming was considered ever since the early stages of the Internet, and nowadays, it is the most popular application on the web. According to [7], during the peak hours, YouTube traffic only accounts for 27% of the mobile downlink (DL) video traffic in North America. With video streaming, a user can start playing the video before the entire video file is downloaded. Most videos on the web nowadays are accessed via streaming. Contents such as movies, video news clips, and YouTube videos are watched by millions of people every day.

HTTP video streaming is the most popular form of video streaming nowadays, and it has been adopted by major video streaming solutions such as YouTube, Netflix, and Hulu. This is due to the convenience of using HTTP [8], which eliminates the need to install and use a dedicated streaming application and helps to get the streaming traffic past firewalls. HTTP video streaming works by breaking the overall video stream into a sequence of small HTTP-based file downloads, referred to as video segments. Users progressively download these small segments, while the video is being played. Playout usually starts after receiving a certain "sufficient" number of video segments. The received segments are buffered in a video/application buffer. The application that plays the video is usually referred to as the client. The streaming client requests video segments from the server. Since each segment has a fixed duration, the size of the segment depends on its duration and the video bit rate. The client receives the pieces from the video buffer. The duration of video content available for playout 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.

Bad network conditions (insufficient bandwidth, delay, etc.) may cause the playout buffer to get empty, as the video bit rate is higher than the video streaming rate, which causes video playout interruptions. These interruptions are referred to as video stalling or rebufferings. When stalling occurs, playout stops until sufficient data is buffered again.

Although HTTP video streaming provided a convenient way for video streaming over the Internet, it was still challenging to stream video to wireless and mobile devices due to the high bandwidth variability of the wireless links. DASH provided a promising technique to improve video streaming over networks with varying bandwidth [9], as it allows changing the quality of video streaming to adapt to network conditions.

DASH provides two features that helped improving video streaming. First, it breaks down the video into small, easy to download segments (for example 5-seconds chunks). Second, each segment is encoded at multiple bit rates, providing multiple quality levels for each segment, which allows adaptive streaming. Clients will choose between various bit rates to adapt to the network conditions. As such, DASH helps improving the bandwidth utilization and reducing the interruptions of the video playback, which results in a higher streaming quality. Due to these advantages of DASH over classical HTTP video streaming, DASH has been employed by big video streaming platforms, such as YouTube and Netflix, and it is being adopted by an increasing number of video applications.

There are various adaptation strategies that can be used to determine how the client selects the streaming quality to adapt to the varying network conditions. 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.

With the increasing demand for video applications, providing high quality video service as perceived by the end user has become an important concern for cellular network operators. As such, quality measure has shifted from Quality of Service (QoS) to QoE. The ITU defines QoE as the overall acceptability of the service as perceived by the end user. Video streaming QoE is very important because users pay their operator and they expect to get video service with good QoE in return. If the user is not satisfied, they may look for other options and switch to another provider. As such, video streaming QoE must be considered in the network design and management in order to maintain user satisfaction. There are many factors that are used to measure video streaming QoE, here we present the most important ones [10].

- Video stalling (rebuffering): the stopping of video playback as the playout buffer gets empty. Increasing video stalling decreases the QoE. Many studies [10] have shown that video stalling has the biggest impact on QoE, and thus, should be avoided as much as possible.
- Video continuity index: a measure of the extent by which rebuffering pauses are avoided [11]. The continuity index is measured as follows,

$$\eta_c = 1 - \frac{\Delta T_{rb}}{\Delta T}, \tag{1}$$

  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).
- Initial (startup) delay: the delay from the request to stream the video until the playback starts. Initial delay is always present as certain number of video segments should be received before decoding and playback starts.
- Video bit rate: it is a measurement of the amount of data in one second of the video. Video bit rate is determined by many quality factors of the video such as video frame rate, resolution, and quantization parameters. As the video bit rate increases, the video quality increases, which increases the QoE.

B. Related work

Direct communication between nodes in the wireless networks has been studied in the context of wireless ad hoc networks [12]. However, it has not been considered in cellular networks until the emergence of D2D communications [13-16]. The introduction of D2D communications in the LTE-A standard has opened the door for direct P2P communications between UEs in cellular networks [3]. With D2D communication, two UEs that are within proximity of each other communicate directly without going through the BS or the core network.

D2D communication provides a promising solution to increase the capacity in cellular networks by allowing frequency reuse, and by increasing the data rates due to potentially improved transmission over shorter distance and fewer hops. As such, it has been investigated in the last few years [13-16]. Some of the work focuses on finding incentive mechanisms to motivate involvement of UEs in D2D communication as the success of D2D communication depends on the participation of users to share their content [17-19]. We do not consider incentive mechanisms here, as this is a different research area that is out of the scope of this paper.

There has been some work on P2P video streaming in cellular networks. A protocol for P2P video streaming on mobile phones, called RapidStream, was proposed in [20]. It is similar to many of the P2P streaming protocols on wired networks that involve the dissemination of buffer maps and video chunks between peers. While such protocols work well in wired networks, they involve too much signaling and transmission (dissemination of buffer maps) to be appropriate for UEs that has limited power, processing, and transmission resources (especially on a large scale). In [21], multi-source video streaming was proposed where mobile users can connect through WiFi direct to other users to get some of the video content. Such system requires the device to perform device discovery to find neighbors, and service discovery to find services offered by neighboring devices. These requirements along with the signaling needed to exchange content consume significant amount of resources.

In [22], a system, called MicroCast, was designed and evaluated using a testbed. MicroCast is used by a group of smart phone users who trust each other, are interested in watching the same video at the same time, and who are within proximity of each other. Users employ their cellular connection to download segments of the video, and use their WiFi connections to share among each other the downloaded content, to improve the streaming experience. While this could result in some improvement for a group of users, the scope of the system is limited. Furthermore, users usually do not use their cellular connection for downloading video segments when WiFi is available.

In [23] 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.

None of the research studies above on P2P video streaming in cellular networks considers how the video segments are actually cached. When evaluating the performance, they consider that requested segments are already available in helper UEs. With DABAST, we provide a framework that takes care of video content caching and distribution. Moreover, the employed CSVD algorithm provides an approach for inter-cluster as well as inter-cell interference mitigation. Furthermore, the work above considers small-scale networks, i.e., up to 10 UEs including the helpers. We show that using clustering and BS assistance, the potential of collaborative D2D communication between UEs is significant. In [4], we proposed DABAST and provided a preliminary investigation of its performance under a single transmission setup. Here, we extend our work by, first, providing further details on the components of DABAST and its operation. Second, we further investigate the performance of DABAST under various scenarios, to provide quantitative evaluation of the impact of many parameters on the performance of DABAST.

We used the DEVS formalism [24] to build a model for the DABAST architecture, and used that model to test and evaluate the performance of DABAST using various simulations. 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. A DEVS model is composed of structural (Coupled) and behavioral (Atomic) components, in which the coupled component maintains the hierarchical structure of the system, while each atomic component represents a behavior of a part of the system. The CD++ toolkit [25] was used to implement our model of DABAST. CD++ is an open-source simulation software written in C++ that implements the DEVS abstract simulation technique. The simulation engine tool of CD++ is built as a class hierarchy. C++ is used to develop the atomic components of the model. These components can be incorporated into the class hierarchy. Passive classes can be also used to model components of the system. Coupled models can be created using a language built in the simulation engine.

III. THE DABAST ARCHITECTURE

As previously mentioned, DABAST improves video streaming QoE of end users, by reducing the bottleneck of the RAN. This is achieved by employing the CSVD algorithm [5-6]. In this section, we briefly introduce the CSVD algorithm employed. Afterwards, we describe the DABAST architecture and how it is implemented in the cellular network.

A. The CSVD algorithm

CSVD employs BS assisted D2D communication for P2P video content distribution in cellular networks. As previously mentioned, when we use the term P2P transmission here, we refer to direct transmission of video content between UEs in the cell over D2D links. In CSVD, the BS divides the coverage area into non-overlapping subareas, each of which is 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. Only the UEs in the middle of each cluster are selected as SMs, in order to prevent inter-cluster as well as inter-cell interference when the SMs transmit to other UEs in the same cluster using D2D links. 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 file (or parts of it) is available in any of the SMs, the BS will ask the SMs to send the pieces to the requesting UE over D2D links.
- Send To an SM (STSM): if the requested file is not available in the distributed cache (or more copies need to be cached in the cluster) and the requesting UE is an SM, the BS will send the file to that UE over a cellular link, and it will ask the UE to cache the file. This case allows the SMs to cache video files. These files will be available for UEs in the cluster when requested later.
- Send To a UE (STUE): otherwise, the BS will send the file directly to the requesting UE over a cellular link.

As shown in [6], CSVD achieves significant improvements in terms of the cell's aggregate data rate as well as the average data rate.

B. *DABAST implementation*

Fig. 1 illustrates the main structure for DABAST. At the bottom, we have the LTE-A network that provides the infrastructure for communication between the BS and UEs over cellular links, and the communication between UEs over D2D links where the UEs exchange data directly without going through the BS.

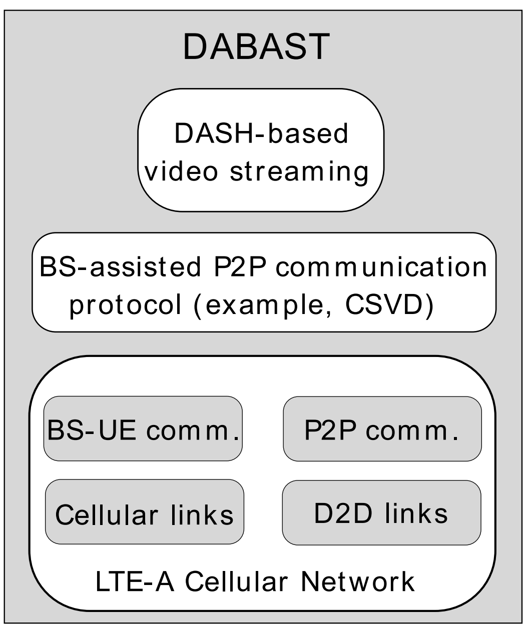

Fig. 1. The DABAST architecture.

A BS assisted P2P/D2D communication protocol is implemented on top of that (here, as an example, we use CSVD) which uses both the cellular and D2D communication. DASH-based video streaming takes place on top of these layers, as the transmission of video segments is implemented as per the communication protocol at the layer below.

Fig. 2 illustrates how DABAST can be implemented 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 the cached and segmented video download algorithm. The CSVD server can also be used to provide caching capabilities to store popular files at the BS.

The requests sent by the streaming client are processed by the CSVD server as per the used cached and segmented algorithm (here, we use the CSVD). If the requested content is to be delivered from the cluster's cache, assistance requests will be sent to SM(s). Otherwise, the request will be forwarded to the DASH server. Here, 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 cluster's cache and D2D channel.

Furthermore, since we assume continuous playout here, we consider a proactive DABAST architecture. In such architecture, the DASH controller in each UE will adjust the video rate according to the length of the playout buffer. When the video bit rate needs to be changed, a request is sent to the BS to change it. Meanwhile, video segments are downloaded on behalf of the client. This reduces the signaling and latency between the BS and the UE.

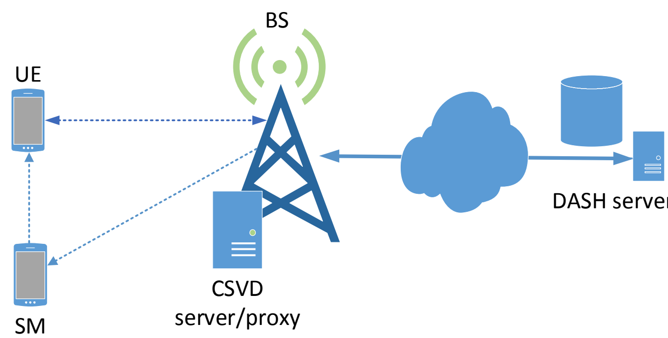

Fig. 2. Illustration of the implementation of DABAST.

IV. MODELING DABAST WITH DEVS

As discussed earlier, we used DEVS to build a model of DABAST in an LTE-A network. Fig. 3 shows the coupled DEVS model of the top-level architecture. As can be seen, we have defined a *Cell* coupled model that contains many *UE* coupled models, a *BS* coupled model, and a *Transmission Medium* coupled model. The Cell coupled model also contains *Cell Manager* and *Log Manager* atomic models.

A *UE* coupled model contains four atomic models: *UE Queue*, *UE Controller, Streaming Client, and DASH controller*. Messages received are buffered at the *UE Queue*. The *UE Controller* is where the UE part of the CSVD algorithm is implemented.

The DASH-based streaming client is implemented in the *Streaming Client* and *DASH controller* atomic models. The streaming client manages the video buffer. It adds video segments received to the video buffer and removes video segments that were played from the buffer. As the video buffer usually has a certain length that could be shorter than the video length, it is implemented as a sliding window. Video segments that were already played will be removed from the video buffer and the buffer slides to cover the next segments in the stream. The *DASH controller* implements the adaptation algorithm. It monitors the video playout buffer, and updates the video bit rate accordingly. When the video bit rate is to be updated, a request is sent to the BS with the new video bit rate.

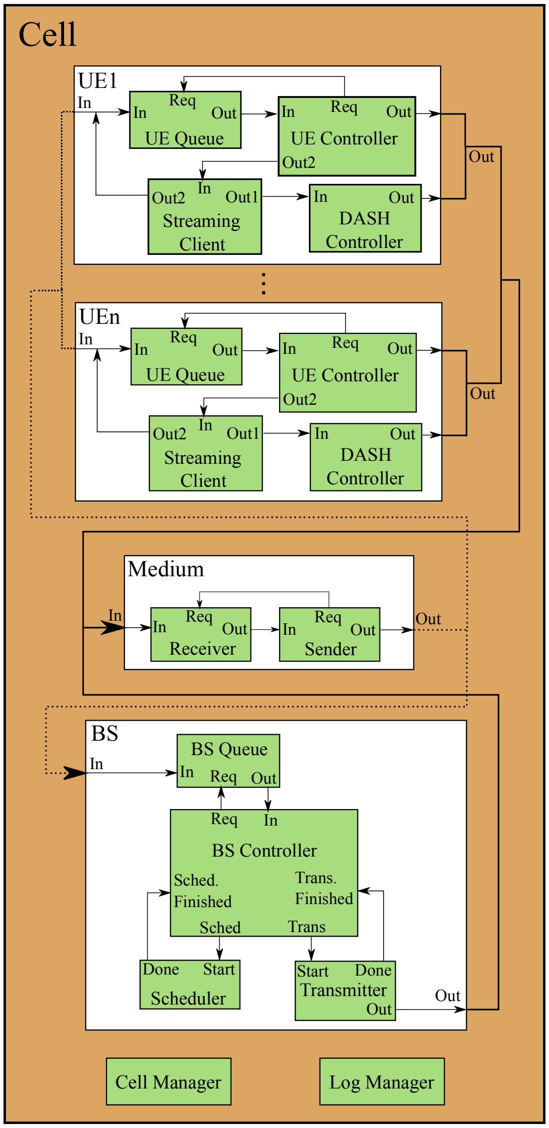

Fig. 3. Coupled DEVS model of DABAST.

The *BS* coupled model includes four atomic models: *BS Queue*, *BS Controller*, *Scheduler*, and *Transmitter*. Received messages are buffered at the *BS Queue*. The *BS Controller* is where the BS part of the CSVD algorithm is implemented. The *Scheduler* schedules the messages to be transmitted in the next Transmission Time Interval (TTI), which is 1 ms. Every TTI, the *BS Controller* also asks the *Transmitter* to send messages that were scheduled for transmission during this TTI.

The *Medium* model simulates the transmission medium and the *Cell Manager* atomic model initializes and sets the parameters of the cellular DLs and uplinks (ULs) between the BS and the UEs, as well as the D2D links between the UEs. For further details on the communication models used for simulation of the LTE-A cellular links and D2D links, the reader is referred to [6]. In addition to the atomic models above, many other passive classes where developed to model other components of the system such as classes to model the cellular and D2D links, download sessions the BS has with UEs, etc.

V. SIMULATION SCENARIOS AND RESULTS

Simulations were performed to evaluate the performance of DABAST in terms of the aforementioned QoE metrics. The simulation setup is shown in Table I. We ran various simulations to study the impact of many parameters on DABAST such as files' popularity, the number of UEs in the cell, etc.

The simulations consider a single LTE-A cell. The urban macro propagation model [26] 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 [27].

TABLE I. SIMULATION SETUP

Parameter	Value
Cellular Channel BW (MHz)	10
Cell Range (m)	500
Number of clusters	9
BS antenna gain (dB)	12
BS transmission power (dBm)	43
UE antenna gain (dB)	0
UE transmission power (dBm)	21
Noise spectral density (dBm)	-174
Antenna height (m)	15
Transmission model	UTRA-FDD
DL Carrier frequency (MHz)	900
Number of requests by a UE	2
Area configuration	Urban
D2D Channel BW (MHz)	60
D2D Carrier frequency (GHz)	24
D2D transmitter TX Power (dBm)	23
D2D Large-scale fading std deviation (dB)	4.3
D2D Receiver noise figure (dB)	9
D2D TX/RX Height from Ground (m)	1.5
Segment length (second)	10
Number of buffered segments to start playout	4
Video bit rate levels (kbps)	384, 768, 2000, 4000
Videos length (second)	441

In each iteration of the simulation, the UEs are uniformly distributed throughout the cell. Clustering takes place in the beginning in case of DABAST where the cell is divided into 9 clusters. The UEs then start requesting video streams. During each iteration of the simulation, each UE will request two video streams. A UE requests a video stream, and after finishing the playout, it will request 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 popularity of videos is generated according to a Zipf distribution to simulate the variable popularity of the videos, as it has been established this is a good model for this purpose [28]. Using this distribution, some videos are requested more often than others. The length of the videos is 441 seconds, which is the mean length of a YouTube video [29]. Four video bit rate levels where used as shown in Table I. These are adapted from the H.264/AVC video coding standard [30].

Regarding the DASH controller, we use the buffer-based approach in [31]. 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 to determine the video bit rate.

TABLE II. PLAYOUT BUFFER LENGTH-VIDEO RATE MAPPING

Playout buffer length (s)	Video bit rate (kbps)
$0 \leq f(B) \leq 130$	384
$130 < f(B) \leq 240$	768
$240 < f(B) \leq 350$	2000
$350 < f(B)$	4000

With this approach, the video bit rate will be set to the minimum level at the beginning, as the buffer is empty. When the playout buffer length reaches a certain value, the client will ask for the next higher video bit rate. The algorithm follows a simple rule, stay at rate R_i as long as the playout buffer length is between B_i and B_{i+1}. The use of hybrid approaches will be considered in future work. We assume the video buffer is long enough to accommodate all received segments. The playout buffer length to video bit rate mapping used is shown in Table II.

We measured the number of rebufferings, video continuity index, initial delay, and video bit rate levels of the received video segments. Table III shows the mean values for these measurements. The results in Table III 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 30 simulation runs. In addition to the mean, we show the Margin of Error (MoE) for 95% confidence interval.

Table III shows that DABAST achieves an improvement over conventional DASH in terms of all the measured metrics above. Regarding the average number of rebufferings, DABAST achieved 49% decrease in the average number of rebufferings, which is a significant improvement. The continuity index is also improved with DABAST due to decreasing the average number of rebufferings as well as the rebuffering time. The average initial delay for conventional DASH is relatively high because in this scenario, there are 500 UEs in the cell requesting video streams, and sharing limited cellular frequency resources (10 MHz).

Table III shows that DABAST also achieves a 47% decrease in the average initial delay, which is also a significant improvement. In addition to the improvements above, DABAST also achieved an improvement in terms of the average video bit rate. Due to the increase in the transmission rates achieved by DABAST, video segments are delivered to UEs much faster than in the case of conventional DASH. This is because in the case of DABAST, the CSVD algorithm is employed, where video segments are sent to many UEs from both the BS (over cellular links) and SMs (over D2D links) as opposed to only from the BS. This reduces the initial delay to receive the first 4 segments needed to start playing, and consequently, it reduces the initial delay. This also reduces the events of video buffer stalling, and consequently reduces the number of rebufferings. In the following, we investigate the impact of different parameters on the performance of DABAST.

TABLE III. SIMULATION RESULTS

	Conventional DASH		**DABAST**	
	Mean	MoE	Mean	MoE
Rebufferings	3.2894	0.0157	1.6800	0.0299
Cont. index	0.75545	0.0007	0.8737	0.0014
Initial delay (s)	51.294	0.1938	27.062	0.5267
Video bit rate (kbps)	389.81	0.2369	411.22	0.9297

Fig. 4 shows the average number of rebufferings for both conventional DASH and DABAST versus the number of UEs in the cell. Fig. 4 shows that at 300 UEs, the average number of rebufferings is 0 for both conventional DASH and DABAST. This means that the cellular resources are enough with conventional DASH to avoid rebufferings for all the video streams. As the number of UEs increases, the cellular resources will be shared by more UEs, which reduces the average data rate and increases the transmission delay of video segments. This increases the possibility of video buffer depletion and increases the average number of rebufferings. Even with DABAST, the average number of rebufferings increases with increasing the number of UEs. This is because we study the case of progressive caching, where in the beginning there are no videos cached in the clusters, and videos are cached by SMs as requested.

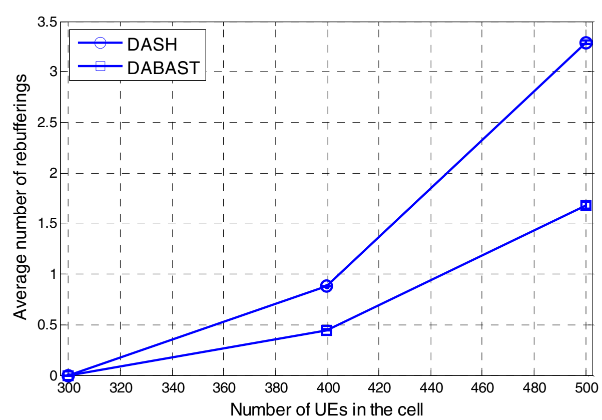

Fig. 4. Average number of rebufferings versus number of UEs in the cell. Zipf exponent = 1.5 and 500 videos.

Fig. 4 shows that the improvement achieved by DABAST increases by increasing the number of UEs in the cell.

Increasing the number of UEs increases the number of requests and also increases the number of SMs in each cluster. This increases the number of cached videos in a cluster and the number of requests that would be satisfied from the cluster cache, increasing the improvement achieved by DABAST over conventional DASH.

Fig. 5 shows the average initial delay for both conventional DASH and DABAST versus the number of UEs in the cell. Fig. 5 shows that although the average number of rebufferings is zero at 300 UEs for both (as in Fig. 4) DABAST achieves improvement in terms of the average initial delay at 300 UEs. Furthermore, it can be noticed that the gain achieved by DABAST over conventional DASH increases with the number of UEs. This is for the same reason explained above for the average number of rebufferings.

As mentioned above, the Zipf distribution was used to model video files popularity. The Zipf distribution has one parameter, namely the Zipf exponent. This exponent controls the relative popularity of files. When the Zipf exponent increases, the content reuse increases. This is because higher exponent means that the popularity of the first files in the list will increase, and they will be requested more often.

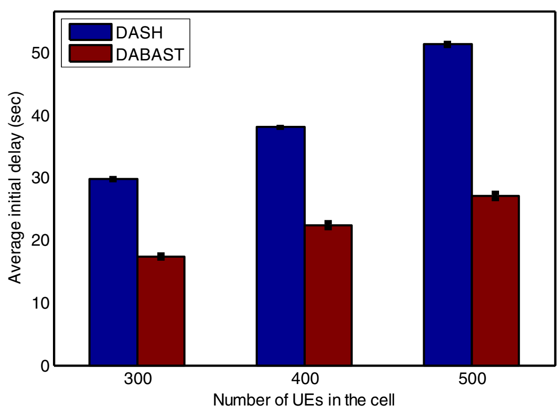

Fig. 5. Average initial delay versus number of UEs in the cell. Zipf exponent = 1.5 and 500 videos.

Fig. 6 shows the average number of rebufferings versus the Zipf exponent. As can be seen, the average number of rebufferings decreases as the Zipf exponent increases. This is because as the number of popular files increases, the content reuse increases because more files will be cached and delivered later from the distributed cache rather than from the BS. This speeds up the transmission process and decreases the possibility of playout buffer depletion, which decreases the average number of rebufferings.

Fig. 6 shows that the improvement achieved with increasing the Zipf exponent slows down. This due to the fact that in our scenario, each UE requests only two videos. Increasing the number of requests made by each UE further increases content reuse and increases the improvement achieved by DABAST. This is because cached video files will be further utilized by the later requests.

Fig. 7 shows the average number of rebufferings for DABAST versus the number of videos available to request from. As can be seen, there is no considerable effect for the number of files on the average number of rebufferings. As per the Zipf distribution, having more videos will not cause a noticeable impact on the probability of requesting the popular files. This means that for a certain Zipf exponent, although the number of videos increases, there will still be content reuse as long as there are popular files.

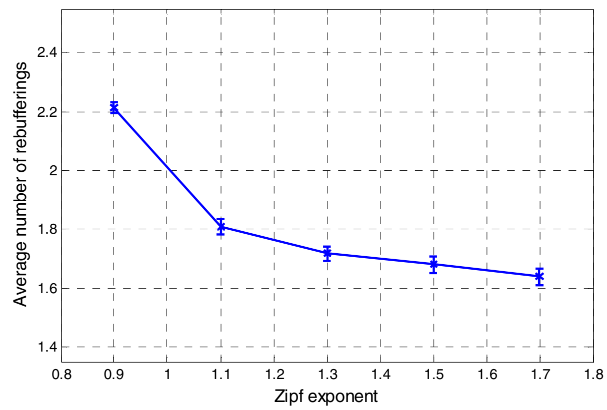

Fig. 6. Average number of rebufferings versus the Zipf exponent. 500 UEs and 500 videos.

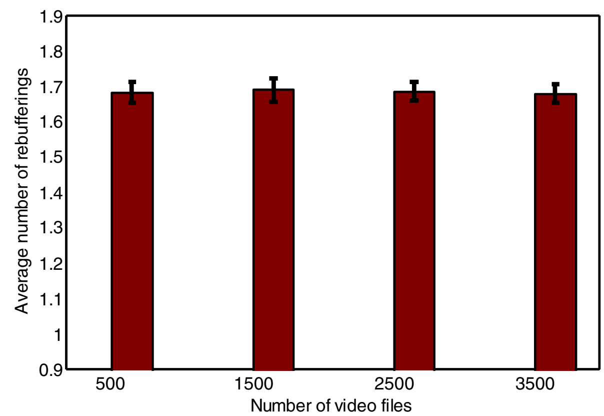

Fig. 7. Average number of rebufferings versus the number of videos. 500 UEs and Zipf exponent = 1.5.

VI. Conclusion

The continuous increase in resource-demanding video applications is outpacing the improvements on the cellular network capacity. As such, providing good Quality of Experience (QoE) video streaming service is becoming a main concern for cellular network operators. This made it necessary to develop new approaches that help serving the increasing video traffic over cellular networks, and improve the QoE of video streaming. Here, we present our proposed architecture, namely, **D**ynamic **A**daptive Streaming over HTTP (DASH) -based **BS** **A**ssisted P2P/D2D video **ST**reaming in cellular networks (DABAST).

DABAST employs Cached and Segmented Video Download (CSVD) for Base-Station (BS) assisted Device-to-Device video transmission in cellular networks. We use the Discrete EVent System Specification (DEVS) formalism to build a model for DABAST in an LTE-A network. We used the developed model to study the performance improvement achieved by DABAST. Furthermore, we study the impact of different parameters (videos' popularity, number of UEs in the cell, etc.) on the performance of DABAST. We study the performance of DABAST in terms of many video streaming QoE metrics. Results have shown that the performance gain achieved by DABAST increases by increasing the number of UEs participating and the popularity of the videos available. This is because these factors increase content reuse, which increases the transmission rate of video contents. Results also show that the number of available videos does not significantly affect the performance of DABAST as long as there are popular videos.

In future work, we will investigate further improvements to DABAST, by improving, for instance, video segment distribution and video rate adaptation.

REFERENCES

[1] Cisco, "Cisco visual networking index: global mobile data traffic forecast update." Feb. 2017 [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[2] Sandvine, "Global internet phenomena report," technical report, 2016.

[3] S. Parkvall and D. Astely, “The evolution of LTE towards IMT-Advanced,” *Journal of Communications*, vol. 4, No. 3, pp. 146-154, Apr. 2009.

[4] A. Al-Habashna, S. Fernandes, and G. Wainer, "DASH-based peer-to-peer video streaming in cellular networks," *in Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, Montreal, Canada, 2016, pp. 481-488.

[5] A. Al-Habashna, G. Wainer, G. Boudreau, and R. Casselman, "Improving wireless video transmission in cellular networks using D2D communication," Canada. Provisional patent P47111. May, 2015.

[6] A. Al-Habashna, G. Wainer, G. Boudreau, and R. Casselman, "Distributed cached and segmented video download for video transmission in cellular networks," *in Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, Montreal, Canada, 2016, pp. 473-480.

[7] 3GPP, "Policy and charging control signaling flows and quality of service (QoS) parameter mapping," technical report, Jan. 2016.

[8] B. Li, Z. Wang, J. Liu, and W. Zhu, "Two decades of internet video streaming: a retrospective view, " *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, pp. 1-20, Oct. 2013.

[9] T. Stockhammer, "Dynamic adaptive streaming over HTTP: standards and design principles," *in Proc. ACM MMSys,* New York, USA, 2011, pp. 133-144.

[10] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A servey on quality of experience of HTTP adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469-492, Mar. 2015.

[11] L. Cicco and S. Mascolo, "An adaptive video streaming control system: modeling, validation, and performance evaluation," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 526-539, Apr. 2014.

[12] J. Zhao, P. Zhang, G. Cao, and C. R. Das, "Cooperative caching in wireless P2P networks: design, implementation, and evaluation," *IEEE Trans. on parallel and distributed systems*, vol. 21, no. 2, pp.229-241, Feb. 2010.

[13] A. Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1801-1819, Nov. 2014.

[14] B. Kaufman and B. Aazhang, “Cellular networks with an overlaid device to device network,” *in Proc. of Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, USA, 2008, pp. 1537–1541.

[15] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, “Device-to-device communication as an underlay to LTE-advanced networks,” *IEEE Communications Magazine*, vol. 47, no. 12, pp. 42–49, Dec. 2009.

[16] K. Doppler, M. P. Rinne, P. Janis, C. Ribeiro, and K. Hugl, “Device-to-device communications: functional prospects for LTE-Advanced networks,” *in Proc. of IEEE ICC Workshops*, Dresden, Germany, 2009, pp. 1–6.

[17] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Contract-based incentive mechanisms for device-to-device communications in cellular networks," *IEEE JSAC,* vol. 33, no. 10, pp. 1-12, Oct. 2015.

[18] L. Gao, J. Huang, Y. Chen, and B. Shou, “Cooperative spectrum sharing: a contract-based approach,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 1, pp. 174–187, Jan. 2014.

[19] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, “Motivating smartphone collaboration in data acquisition and distributed computing,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2320–2333, Oct. 2014.

[20] P. Eittenberger, M. Herbst, and U. Krieger, "RapidStream: P2P streaming on android," *in IEEE International Packet Video Workshop*, Munich, Germany, 2012, pp. 125-130.

[21] V. Siris and D. Dimopoulos, "Multi-source mobile video Streaming with proactive caching and D2D communication," *in IEEE WoWMoM*, Boston, USA, 2015, pp. 1-6.

[22] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "MicroCast: cooperative video streaming on smartphones," *in Proc. MobiSys*, New York, USA, 2012, pp. 57-70.

[23] T. Q. Duong, N-S. Vo, T-H. Nguyen, M. Guizani, and L. Shu, "Energy-aware rate and description allocation optimized video streaming for mobile D2D communications," *in Proc. IEEE ICC*, London, UK, 2015, pp. 6791 - 6796.

[24] B. Zeigler, H. Praehofer, and T. Kim, Theory of modeling and simulation. San Diego: Academic Press, 2000.

[25] G. Wainer, Discrete-event modeling and simulation: a practitioner's approach. Boca Raton: CRC/Taylor & Francis Group, 2009.

[26] 3GPP, "Evolved universal terrestrial radio access; RF system scenarios," technical report, Dec. 2015.

[27] A. Al-Hourani, S. Chandrasekharan, and S. Kandeepan, “Path loss study for millimeter wave device-to-device communications in urban environment,” *in Proc. ICC*, Sydney, Australia, 2014, pp. 102-107.

[28] M. Cha, H. Kwak, P. Rodriguez, Y-Y. Ahn, and S. Moon, “I tube, you tube, everybody tubes: analyzing the world’s largest user generated content video system,” in *Proc. ACM SIGCOMM,* New York, USA, 2007, pp. 1–14.

[29] S. Ahsan, V. Singh, and J. Ott, "Characterizing internet video for large-scale active measurements," *arXiv preprint arXiv:1408.5777v1*, Aug. 2014.

[30] ITU-T Recommendation H.264, “Infrastructure of audiovisual services-coding of moving video,” Jan. 2012.

[31] T-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: evidence from a large video streaming service," *in Proc. ACM SIGCOM*, New York, USA, 2014, pp. 187-198.