Wireless Networks (2020) 26:2051–2073
https://doi.org/10.1007/s11276-019-02055-x

QoE awareness in progressive caching and DASH-based D2D video streaming in cellular networks

Ala'a Al-Habashna[1] · Gabriel Wainer[1]

Published online: 13 June 2019

Abstract
In this paper, we present an architecture to improve video streaming quality of experience (QoE) in cellular networks with high user equipment (UE) density. In the proposed architecture, video segments are progressively cached, as requested, in selected UEs called storage members (SMs). Video segments are strategically cached to be available to requesting users in the cell. Furthermore, the base-station controls the device-to-device communication between the UEs to provide collaborative peer-to-peer transmission of video segments. Dynamic adaptive streaming over HTTP is also employed to adapt the quality of video segments to network conditions. We study the improvements achieved by the proposed architecture in terms of many video streaming QoE metrics. Thereafter, we improve the operation of the proposed architecture by introducing QoE awareness to both caching and distribution of video segments. We employ QoE awareness in three aspects of the proposed architecture; cellular resource allocation, caching of video segments, and SM-assignment optimization. We analyze the improvements achieved by the prospered QoE-awareness techniques in terms of video streaming QoE metrics.

Keywords 5G · Collaborative D2D communication · QoE awareness · Adaptive video streaming

1 Introduction

Nowadays, more than half of the global online video viewing take place over mobile devices [1]. This increasing adoption of mobile devices for video viewing and the rise of many platforms for video streaming and Over-The-Top (OTT) media services have caused video traffic to account for the majority of data traffic over mobile networks. According to [2], YouTube traffic only forms 21% of the total mobile downlink traffic in north America during peak hours.

This increasing popularity of video streaming is escalating the growth of data traffic to be transmitted over cellular networks and raising the challenge for cellular network operators. Consequently, supporting video streaming services has become a main concern for cellular network operators, and new techniques are much needed to help serving video traffic that is becoming the majority of data traffic over cellular networks. Moreover, achieving user satisfaction about the video service has become another concern for cellular network operators. This issue is becoming more important considering that not only the popularity of online video streaming is increasing, but also the quality of videos available via online streaming. For instance, YouTube and Vimeo nowadays provide 4 K video support (a very high-resolution format), while having lower video qualities (e.g., 240p). This means that different end users can have different data rate requirements. The video playout dynamics at the clients such as video playout, pausing, and rebuffering further complicate the problem. As such, quality measure has shifted from quality of service (QoS) to quality of experience (QoE) [3, 4], which is the overall acceptability of the service as perceived by the end user [5]. Hence, it is necessary not only to develop new techniques to improve the delivery of video streaming traffic, but also for the new techniques to consider the complex, dynamic, and delay-sensitive nature of video streaming traffic to provide end users with good QoE video streaming.

✉ Ala'a Al-Habashna
alaaalhabashna@sce.carleton.ca

Gabriel Wainer
gwainer@sce.carleton.ca

[1] Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada

Much research in the literature has been conducted on improving video streaming over cellular networks. Most of the work in the literature either focus on developing methods for cellular resource allocation over traditional cellular networks (e.g., [6, 7]), or on developing new adaptation strategies for the Dynamic Adaptive Streaming over HTTP (DASH) (e.g., [8, 9]); an adaptive video bit rate streaming technique. There has also been some work on utilizing Device-to-Device (D2D) communication to improve video streaming over cellular networks (e.g., [10, 11]). D2D communication is a technique that allows direct communication between nearby devices in cellular networks. We discuss the related work in the literature, in more detail, in Sect. 2.2.

In [12], we propose an architecture to improve video streaming QoE in cellular networks under high traffic load. The architecture employs the cached and segmented download algorithms that we propose in [13–15]. The cached and segmented download algorithms provide base-station (BS)-controlled progressive caching of video segments in selected user equipments (UEs) in the cell. These UEs are called storage members (SMs) [15]. The algorithms are also implemented by the BS to improve delivery of video contents by providing collaborative D2D transmission of video segments among UEs in the cell. Furthermore, the architecture supports DASH. The proposed architecture is called **DA**SH-based **B**S-**A**ssisted D2D video **ST**reaming in cellular networks (DABAST). In [12], we evaluate the performance improvements achieved by DABAST in terms of many video streaming QoE metrics. Results show that DABAST significantly improves video streaming QoE for many users in the cell by enhancing all the measured QoE metrics.

We discuss DABAST in Sect. 3, and present some of our previous performance evaluation results in Sect. 6.1. As these results show, despite the clear improvements achieved by DABAST for a significant portion of the video streams in the cell, a high number of video streams did not experience considerable improvement, and hence, did not receive video streaming service with good QoE, due to the repetitive video rebuffering. As such, making DABAST aware of video streaming QoE for users in the cell can further increase the QoE gains achieved by DABAST. QoE awareness can be used in DABAST to minimize QoE degradation that might be inevitable for some users under high traffic load. This can be achieved by targeting users who are experiencing poor QoE or by improving a certain QoE metric as needed. QoE awareness can be also employed to maximize the QoE achieved over the network, or to achieve a balance between the objectives above.

Here, we extend our previous work in [12] by employing QoE awareness in three aspects of DABAST, namely, cellular resource allocation, caching of video segments, and SM-assignment optimization. We analyze the improvements achieved by each QoE-awareness technique in terms of video streaming QoE metrics. Results show that all the QoE-awareness techniques above improve the performance gains achieved by DABAST. Here, we summarize the main contributions of this paper over our previous work:

- Employment of QoE-aware cellular resource allocation in DABAST and analyzing the improvement achieved over state-of-the-art cellular resource allocation algorithms.
- An approach for High Rate Caching (HRC) of video segments in DABAST to further improve video bit rate under relatively lower traffic load.
- An optimization model for the SM-assignment problem in DABAST to simultaneously maximize the aggregate video bit rate in the cell and minimize the number of rebufferings.
- Performance evaluation of the improvements achieved by HRC and SM-assignment optimization in terms of many video streaming QoE metrics.

The remaining of this paper is structured as follows, in Sect. 2, we discuss the background and review the related work in the literature. In Sect. 3, we present DABAST and its implementation in cellular networks. In Sect. 4, we discuss cellular resource allocation and the proposed QoE-aware scheduling metric for DABAST. In Sect. 5, we present the other two QoE-awareness approaches, namely HRC and SM-assignment optimization. In Sect. 6, we start by evaluating the performance improvements achieved by DABAST. Afterwards, we analyze the performance improvements achieved by all the proposed QoE-awareness techniques. Finally, in Sect. 7, we present the conclusion of this work.

2 Background and related work

Here, we discuss the background for the topics involved in this work. Thereafter, we review the related work in the literature.

2.1 Technical background

2.1.1 HTTP video streaming

HTTP video streaming has an important role in the ubiquitousness of online video streaming, as it allows users to request and watch online videos using web browsers and avoid the NAT and firewall traversal problems [16]. Nowadays, HTTP video streaming is the most popular way of online video steaming. Moreover, it is employed by the

biggest video sharing and OTT media services such as YouTube and Netflix.

With HTTP video streaming, the video file is divided into a stream of small HTTP files called segments. When a user requests a video stream, video segments are downloaded progressively. Received segments are stored in a video buffer. The client starts video playback after few segments are received, and the remaining segments are downloaded during playback. Playout buffer length refers to the duration of video contents available in the video buffer for playback. Video playout continues as long as video contents are available in the video buffer. If the rate at which video contents are received (data rate) is lower than the rate at which video contents are consumed (playback or playout rate), video buffer will be depleted until it is consumed. In such case, playout stops so that video contents are rebuffered.

2.1.2 DASH

In the case of limited network capacity or variable throughput, the data rate could repeatedly decrease below the playout rate. This could cause multiple playback interruptions (also called video rebufferings). Video rebufferings are annoying and degrade the quality of the service as perceived by the end user. DASH is a dynamic video streaming technique that allows changing the video quality to adapt to the available data rate [17, 18]. With DASH, the segments of a video are encoded at different compression levels. As such, each segment will be available at the server side at multiple video bit rates. A media presentation description (MPD), that contains information about the video segments and the video rates at which they are available, is sent to the client. During progressive video download, the client switches between the different video bit rates to adapt to the varying data rate. This reduces the possibility of video playout buffer depletion when the data rate degrades, which reduces the number of rebufferings and improves the streaming quality. Furthermore, this allows increasing video quality when the data rate increases which improves the channel utilization. Many video streaming platforms, such as YouTube and Netflix, have adopted DASH due to the performance gains it achieves.

In DASH, switching between different video bit rates takes place at the end of video segments. An adaptation strategy is used by the DASH controller at the client to select the video bit rate of the next segment to adapt to varying data rates. A good adaptation strategy achieves a trade-off between two factors; maximizing the quality of the video by selecting the highest video bit rate the network can support, and at the same time, avoid rebufferings. Much research has been carried out on developing video bit rate adaptation strategies [8, 19, 20]. Some of these are based on the estimated instantaneous data rate at the receiver. In such algorithms, the DASH controller selects the highest available video bit rate that is lower than the current throughput. The drawback of these approaches is that it is difficult to obtain an accurate estimation of the data rate in environments with highly variable throughput. Other algorithms are based on the length of the video playout buffer length for adaptation, as it is the main variable the controller is trying to manage. Here, we employ the buffer-based approach in [8], as it is robust to high data rate variability. Furthermore, in DABAST, the UE could receive a video segment from the BS or from an SM. Consequently, it would be difficult to estimate the throughput at which the next segment will be received. The adaptation algorithm, $f(L)$, uses the playout buffer length, L, to select the video bit rate of the next segment [8].

2.1.3 Video streaming QoE

With the increasing demand for video applications, supporting high quality video services while achieving user satisfaction about the service has become an important concern for cellular network operators. Users pay their operator and they expect reliable support for video services in return. If the user is not satisfied, they may switch to another provider. At the same time, different users can have different data rate requirements to achieve their satisfaction, due to the wide quality range of online videos and due to the complex, dynamic, and delay-sensitive nature of video streaming traffic. As per [21], major content providers have lost $2.16 billion due to low video streaming experience, and the loss in revenue due to poor QoE is expected to increase. This has made it necessary to use QoE when evaluating video streaming services and to implement QoE-aware video delivery techniques.

Many subjective QoE studies have been conducted to determine the metrics that influence users' opinion and decide video streaming QoE [4, 22, 23]. These metrics provide objective way to study, evaluate, and improve video streaming QoE. Regarding HTTP video streaming, it has been shown by many studies that video rebuffering and initial delay are very important factors on the user's QoE [4, 23]. The quality of the video, measured by the video bit rate, is also an important factor on the user's QoE of HTTP video streaming.

Obviously, video stalling decreases video streaming QoE. The authors in [24] have shown that users usually prefer one long rebuffering duration over multiple shorter rebufferings. Many studies [23, 24] have shown that video

stalling has the biggest impact on QoE, and even few short rebufferings could have severe impact on the QoE. Consequently, avoiding rebufferings should be a priority. Video continuity index is a metric that considers the ratio of the rebuffering time to the total video viewing time. The continuity index is given by [9],

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

where $\eta_c \in [0, 1]$, ΔT_{rb} is the total rebuffering time, and ΔT is the duration of the experiment (playing time and rebuffering time). When 0 rebufferings are experienced, the continuity index value will be 1, which is the best-case scenario.

Initial delay is another factor that degrades video streaming QoE. It refers to the delay from the time a video stream is requested by the user until the time video playout starts. As video playout usually starts after few segments are received and available for playout, initial delay depends on the amount of video data needed to start playout and on the data rate. Although initial delay has an impact on video streaming QoE, studies have shown that it has less effect than video stalling [23, 25] because initial delay is usually expected, especially for relatively longer videos. Video bit rate is also a metric that affects QoE. A higher video bit rate accommodates higher image resolution and frame rate in the video, which increases video streaming QoE.

2.1.4 Resource allocation in LTE-A networks

The downlink scheduler is a crucial component in LTE-A systems due to its importance in efficient radio resource utilization. The downlink scheduler allocates radio resources to the UEs in the cell according to some objective (e.g., maximizing the cell's aggregate data rate). In LTE-A, the radio spectrum is accessed using orthogonal frequency-division multiple access (OFDMA) in the DL [26]. With OFDMA, the radio channel is divided in both the time and frequency domains. In time, the channel is divided into sub-frames, and each sub-frame is 1 ms. Each sub-frame is divided into 2 slots of 0.5 ms each. In frequency, the channel is divided into sub-channels of 180 kHz each. A unit that is composed of one slot in time (0.5 ms) and one sub-channel in frequency is referred to as one Resource Block (RB). A RB is the smallest schedulable radio resource unit. The number of RBs in the channel depends on the channel bandwidth. For example, a 10 MHz channel contains 50 RBs. In LTE-A, the scheduler is implemented at the BS, and it is responsible for allocating RBs to active UEs. Scheduling takes place every transmission time interval (TTI), which equals to 1 ms. Scheduling involves deciding which UEs will be allocated the RBs in the next 1 ms interval, based on a certain scheduling metric.

2.1.5 D2D communication in cellular networks

D2D communication is one of the main technologies in the fifth generation (5G) cellular networks [27] due to the improvements it provides. In traditional cellular networks, all communications from and to the UEs have to be relayed over the BS. In such communication paradigm, the radio access network (RAN) becomes the main bottleneck in cellular networks with limited cellular frequency resources that are shared among large number of users. With D2D communication, on the other hand, two UEs within proximity of each other can exchange data over direct links, without the need to relay the traffic over the BS. This can improve the data rate between the two UEs due to transmission over one hop and shorter distance. Moreover, the capacity of the cellular network can be increased by coordination of multiple short distance transmissions to achieve spatial frequency reuse. D2D communication can also extend the coverage area of the cell and improve the received signal for users at the cell edge. As such, much work has been conducted to develop applications for D2D communications in cellular networks and improve its performance [28–30]. D2D communication allows collaboration of users in cellular networks to share contents they have. However, approaches are needed to motivate participation of users in D2D communication. Incentivizing users to participate in D2D communication is a topic that has received much interest in the last couple of years. The interested reader in this area is referred to [31, 32].

2.2 Related work

Some work in the literature have been conducted on D2D video streaming in cellular networks. In [33], a system called MicroCast was proposed to improve the quality of video streaming. MicroCast is designed for a small group of smart phone users who are within proximity of direct communication from each other and would like to watch the same video at the same time. A similar P2P application was developed in [11] for live video streaming. The application is designed for a small set of devices that have both cellular and WiFi connections and interested in watching the same live stream. In both systems, users employ their cellular connections to download chunks of the video and use their WiFi connections to share among each other the downloaded chunks to improve the quality of the video stream. While the proposals in [11, 33] result in improvement for a small group of users, their scope is limited as they are designed for a small group of users. Furthermore, these systems are designed for live video streaming where users have synchronous playout of the video.

In [34], a protocol, called RapidStream, was proposed for P2P video streaming on mobile phones. The protocol is analogous to some P2P streaming protocols on wired networks where peers disseminate their buffer maps to announce availability of video segments and request video chunks from each other based on such information. Although such protocol is suitable for P2P video streaming over wired networks, it requires too much signaling and transmission (dissemination of buffer maps) to be scalable for UEs that has limited power, processing, and transmission resources.

In [35], a system that employs D2D communication was proposed to improve the delivery of video segments to requesting UEs. In that system, multiple helpers can be used to deliver video segments to the requesting UE. The video is encoded by applying multiple description coding by each helper. Each helper sends a different description to the requesting UE. The authors propose an optimization model to maximize video quality and efficiently consume the helpers' energy. However, the work in [35] does not consider the most important metrics of video streaming QoE such as the number of rebufferings and initial delay. Improving video bit rate/video quality increases the end user QoE. Never the less, increasing video bit rate could increase transmission delays which increases the number of rebufferings and initial delays. Such factors are very important to consider as degradation of these metrics would significantly worsen video streaming QoE.

All the work above either consider the case of live streaming or do not consider how video segments are cached in helpers when evaluating performance (requested segments are assumed to be pre-cached). Moreover, all the work above consider small-scale networks (up to 10 UEs including helpers). In this work, we present and evaluate DABAST; an architecture to improve the QoE of video streaming in cellular networks with high user density. DABAST employs the cached and segmented download algorithms for BS-assisted progressive caching and D2D video transmission between UEs [15]. As such, DABAST does not only consider the transmission of video segments over both cellular and D2D links, but also caching of video segments. Video segments are strategically cached as per the algorithms to be available to all users in the cell. Furthermore, segments are progressively cached as requested (not to waist valuable cellular resources). Although these characteristics are very beneficial in any video streaming scenario with high number of users, they are crucial for high traffic load scenarios where various unsynchronized users are streaming recently published contents (e.g., a YouTube live video). However, with DABAST, not all requested segments are cached in the cell and the cached segments are not available in all video bit rates. Moreover, there are different frequency resources available, i.e., a cellular channel with limited bandwidth and out-of-band channel with higher bandwidth. As we will see, many decisions need to be taken dynamically regarding caching and distribution of video contents, as well as allocation of resources and SMs to improve video streaming QoE for users in the cell. Additionally, DABAST operates in proactive mode under high traffic load, where a series of successive video segments are transmitted to the user from multiple sources. From all the above, it is necessary to study resource management in DABAST and how it impacts its performance.

In [36], the authors proposed two scheduling algorithms for the delivery phase in D2D video streaming. The algorithms allocate frequency resources to several pre-matched D2D links. A similar algorithm has been also proposed in [37], where a single channel is shared by multiple D2D links of pre-matched helper-receiver pairs. As with the previous work above, the work in [36, 37] assume that all videos are pre-cached in the user devices before video streams start, and that each caching device already have the video available at multiple video qualities. Our proposed architecture considers not only the delivery phase, but also video caching in the UEs in a high traffic load scenario. Furthermore, the work in [36, 37] only consider the D2D communication between the UEs in the network over a single channel, considering that all requested videos are cached. DABAST, on the other hand, provides a framework that considers not only the D2D communication between UEs (on a D2D channel), but also the cellular communication between the UEs and the BS. This is crucial because in a real scenario, many of the requested videos might not be available in the distributed cache in the cell, and hence, should be downloaded over the cellular channel through the BS. Moreover, the work in [36, 37] assumes that a UE is only provided the segments of a video by one helper. In DABAST, SMs are assigned dynamically to requesting UEs on a segment-by-segment basis, depending on SMs availability and video bit rates of available segments to maximize video streaming QoE.

Many video streaming QoE-aware approaches for resource allocation in conventional cellular networks (without D2D communication) have been proposed [6, 7, 38, 39]. In [6], an adaptive video steaming QoE maximization approach was proposed for resource allocation in LTE networks, where the playout buffer levels are signaled from the UEs to the QoE optimizer at the BS. The optimizer considers the playout buffer levels at the UEs to perform a multi-user resource allocation. A similar approach that employs the playout buffer length is proposed in [7]. Other approaches use a utility function that maps technical parameters such as transmission delay to QoE scores [38, 39]. Based on this mapping, resource allocation is performed to guarantee certain QoE score to

users [38, 39]. The problem with the methods that depend on utility functions for mapping is that such models do not provide an accurate and real time evaluation of users' QoE, especially considering the complex and variable characteristics of video contents as well as the video playout dynamics at the clients such as video playout and rebuffering. The work in [36, 37], adopt QoE-aware approaches. They utilize the queue backlog size at the senders or estimate of the playout buffer as the metric to avoid causing long delays. However, the queue backlog size at the senders or the duration of transmitted video contents do not provide an accurate and real time way of tracking the playout buffer at the clients considering the video playout dynamics at the clients and that a user might get video contents from different sources.

In this work, we employ QoE awareness in three aspects of DABAST, namely, cellular resource allocation, caching of video segments, and SM-assignment optimization. For cellular resource allocation, we propose a scheduling metric that employs the playout buffer length which is similar to the one in [6] for traditional cellular networks (without D2D communication). Our scheduling algorithm is proposed to operate in DABAST and considers not only the reported playout buffer length, but also an estimation of the transmitted video contents over both the cellular and the D2D channel. Furthermore, we propose an approach for high video bit rate caching of segments in the UEs that also utilizes the metric above. Finally, we propose an approach for SM-assignment optimization to maximize the achieved QoE for users in the cell. Results show that all the QoE-awareness techniques above do indeed improve the performance gains achieved by DABAST.

3 The DABAST architecture

In this section, we present DABAST and discuss its implementation in cellular networks. DABAST employs the cached and segmented download algorithms at the BS to provide progressive caching of video segments in SMs and to improve the delivery of video contents by providing D2D transmission of video segments from SMs to requesting UEs. The implementation of DABAST in cellular networks is depicted in Fig. 1. As the figure shows, a CSVD proxy is deployed at the BS to implement the caching and distribution algorithm. In the following, we present the Cached and Segmented Video Download (CSVD) algorithm [14, 15], which is employed here to study the performance improvements achieved by DABAST.

With CSVD, the cell is divided by the BS into clusters. UEs in the cell are assigned to clusters based on their geographical location in the cell. Furthermore, UEs in the central area of each cluster are selected as the SMs of that cluster. SMs of a cluster are helper UEs that are used as the distributed cache of that cluster. SMs are selected in this way to prevent inter-cluster as well as inter-cell interference when the SMs transmit to other UEs in the same cluster over the D2D channel. When a UE requests a video file, the BS processes the request and responds as follows:

- Send To an SM (STSM): This case is employed to progressively cache video segments in the SMs of the clusters as requested. If the requested video file is not available in the distributed cache of a cluster (or more copies need to be cached in the cluster), the BS sends the file to an SM in the cluster and asks the SM to cache the file. These files will be available for UEs in the cluster when requested later.
- Send With Assistance (SWA): if requested segments of a video file are available in any of the SMs, the BS will ask the SMs to send the segments to the requesting UE over D2D links.
- Send To a UE (STUE): otherwise, the video file is sent through the BS over the cellular channel.

In [14, 15], we define a protocol for CSVD including a variety of messages necessary for the interaction between the BS, requesting UEs, and SMs. A complete definition of the protocol is described. Furthermore, results show that CSVD significantly improves the cell's aggregate data rate as well as the average data rate. It is worth mentioning that D2D communication can take place over the cellular spectrum (in-band) or over unlicensed spectrum (out-of-band). Out-of-band D2D communication has the advantage of further increasing the network capacity and eliminating interference between cellular and D2D communication. As such, we consider the case of out-of-band D2D communication in our research. The channel models employed are listed in Sect. 6.1.

In DABAST, DASH-based video streaming is implemented on top of CSVD, i.e., video segments can be requested by the clients at various video bit rates with DASH, and the caching and delivery of video segments are implemented as per the CSVD algorithm. Video segments can be cached at multiple video bit rates (when possible) and delivered to users over the cellular channel or the D2D channel (when the requested segment is cached).

The CSVD proxy intercepts clients' requests for video segments and decides where to send the video segment from (e.g., as per the CSVD algorithm). If the segment is to be transmitted from an SM, an *Assistance Request* message will be sent to that SM to send the video segment to the requesting UE [15] over the D2D channel. Otherwise, the client's request will be forwarded to the DASH server.

Under high traffic load, avoiding playout rebufferings should be a priority. As such, video segments are

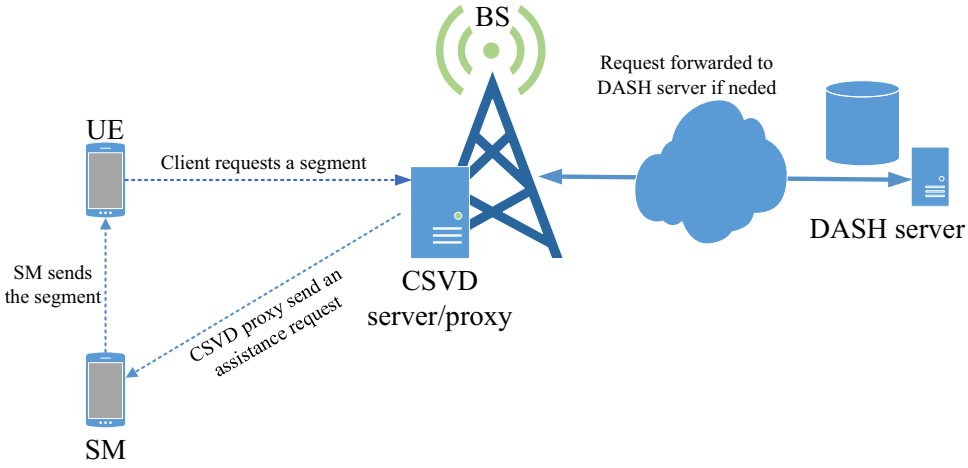


Fig. 1 Implementation of DABAST

progressively cached with low video bit rate, to minimize the number of rebufferings experienced in the cell. Furthermore, in such cases, requested video segments are transmitted to the UEs from the distributed cache (when found) even if the requested video bit rate does not match that of cached segments. This maximizes the utilization of the cached segments and the D2D channel in order to relax the RAN bottleneck and minimize the number of rebufferings. In Sect. 5, we show more sophisticated approaches for caching and distribution of video segments under relatively lower traffic load scenarios.

Under high traffic load, DABAST can also operate in proactive mode. In proactive mode, up to a maximum number of video segments can be transmitted to the requesting UE when found in the distributed cache, without waiting the user's request for each segment. This speeds up transmission of video segments and reduces the signaling and latency between the BS and the UEs. Never the less, proactive mode is activated under high traffic load and when a series of successive video segments are available in the distributed cache. If a video segment is not available in the distributed cache, user's request will be awaited.

4 DABAST with QoE-aware cellular resource allocation

As we will see in Sect. 6.1, results show that despite the clear improvements achieved by DABAST for a significant portion of the video streams in the cell (about half of the streams), the remaining streams do not experience considerable improvement, and hence, do not receive video streaming service with good QoE, due to the repetitive video rebuffering. As such, the operation of DABAST can be further improved by making it aware of video streaming QoE for users in the cell. This can further increase the performance gains achieved with DABAST by targeting users who are experiencing poor QoE or by improving a certain QoE metric as needed. We employ QoE awareness in three aspects of DABAST, namely, cellular resource allocation, caching of video segments, and SM-assignment optimization. In this section, we focus on the first aspect, i.e., cellular resource allocation.

4.1 Cellular resource allocation

Many scheduling algorithms have been proposed for LTE-A. RR is one approach that can be used to allocate frequency resources to UEs. For instance, all the RBs in a TTI can be allocated to a UE each time. Despite its simplicity and fairness, it does not consider the quality of the channel between the UE and the BS, and hence, it does not maximize the system throughput. Furthermore, although RR might be fair in a traditional cellular network, it will not be fair in the case of DABAST because some UEs get video segments over the D2D channel. If this data transmitted over the D2D channel is not taken into consideration, users who get some video segments over the D2D channel and others who get all their video segments over the cellular channel will be treated equally. One can tell that this unfairness might increase the delay to transmit segments to users who get video segments exclusively over the cellular channel, and consequently increase the possibility of rebuffering for such users.

Another algorithm that is widely used in LTE-A systems is Proportional Fair (PF) scheduling [40]. PF scheduling tries to achieve a balance between maximizing the system throughput and achieving fairness among the UEs competing for the cellular resources. This can be achieved by considering for each UE both the current instantaneous rate for that UE, as well as the recent average throughput of that UE.

The scheduling algorithms above try to optimize certain QoS objectives (e.g., fairness or maximizing the aggregate data rate). User satisfaction is very important for network operators and need to be considered in network operation. The algorithms above do not consider the QoE for users and do not try to improve the situation for users who are having low video streaming QoE. Due to the various metrics involved in video streaming QoE and due to the complex and dynamic attributes of video contents, scheduling resources for video streaming users is more complicated. Traditional scheduling algorithms that are oblivious to the end user QoE and to the characteristics of video contents might result in low QoE for high number of users.

In addition to the above, the nature of DABAST, where some video segments can be transmitted over the D2D channel from various sources, increase the complexity of the problem. The cellular resources should be allocated taking into consideration that some UEs will get their segments over the D2D channel, and hence, the cellular resources need to be utilized to help other users who are not fortunate to receive segments over the D2D channel (as their segments are not cached). In this section, we propose a scheduling algorithm for DABAST that takes into account the playout buffer length in the UEs. The metric is similar to the one proposed in [6] for traditional LTE-A systems. However, our algorithm is updated to consider the reported playout buffer length along with estimation of video contents transmitted over both the cellular and D2D channels. It takes into account all the information above to allocate the cellular resources in a way that improves the QoE for the users in the cell. Results show that our algorithm improves the achieved video streaming QoE for users in the cell when compared to RR and state-of-the-art PF scheduling. In the following subsections, we provide a detailed description of PF scheduling, followed by our Buffer-Based scheduling (BB).

4.2 PF scheduling

PF scheduling tries to achieve a trade-off between maximizing the system throughput and achieving fairness among the UEs in the cell [40]. In LTE-A, the BS regularly receives Channel Quality Indicator (CQI) reports from the UEs in the cell [40]. CQI conveys information to the BS on how good the communication channel quality is between the BS and the sending UE. From the information in the CQI, the BS estimates the supported instantaneous data rate, $\hat{r}_k[n]$, for each user k. The PF scheduler then selects the user k' for transmission that has the maximum scheduling metric [40], as follows,

$$k' = \arg\max_k \left\{ M_k^{PF}[n] \right\} = \arg\max_k \left\{ \frac{\hat{r}_k[n]}{T_k[n]} \right\}, \quad (2)$$

where $M_k^{PF}[n]$ is the PF scheduling metric for user k, $T_k[n]$ is the recent average throughput for user k over the past window of N transmission intervals, and n denotes the current scheduling interval. The recent average throughput for user k is calculated as follows,

$$T_k[n] = \left(1 - \frac{1}{N}\right) T_k[n-1] + \frac{\lambda_k}{N} r_k[n], \quad (3)$$

where N is the length of the window over which the average throughput is calculated, $r_k[n]$ is the instantaneous data rate at scheduling interval n, and $\lambda_k[n]$ is the activity factor, and it equals 1 if user k is scheduled for transmission in the *nth* TTI and 0 otherwise. From the above, we can see that the scheduling favors UEs with relatively higher channel quality (to maximize the aggregate data rate) by having the estimated instantaneous data rate in the nominator, and at the same time, try to maintain fairness by having the recent average throughput in the denominator.

Considering the channel instantaneous rate is very beneficial when employing PF scheduling. However, in the context of DABAST, the real advantage is using the recent average throughput of the users in addition to the instantaneous rate. As previously discussed, in DABAST, some video segments are transmitted over the cellular channel, while other video segments are transmitted over the D2D channel. By maintaining the recent average throughput of each UE, we consider the video data that is transmitted over both the cellular and the D2D channel. Hence, for UEs that received segments over the D2D channel, the recent average throughput will be relatively high, which consequently leads to favoring UEs with low recent average throughput when allocating cellular resources. Usually, these are UEs that receive their video segments exclusively over the cellular channel. This should reduce the number of rebufferings experienced by such users and consequently improve their QoE.

4.3 BB scheduling

Although PF scheduling provides advantages over RR scheduling, it still does not take into consideration the end user QoE. In this section, we propose a scheduling algorithm, to use with DABAST, that takes into consideration the reported length of the playout buffer at the client side. Similar metrics have been proposed for traditional LTE-A system. However, our BB scheduling algorithm is proposed to operate in DABAST and considers not only the reported playout buffer length, but also an estimation of the transmitted video duration over both the cellular and the D2D channel (from all SMs).

In BB scheduling, the UEs signal their playout buffer length to the BS. This update can be sent in certain cases; when the change in playout buffer length from the last reported value exceeds a certain threshold (e.g., 5 s), or after a certain period passes from the last update. The update can also be sent when the client status changes (e.g., playing to rebuffering). The main idea is that the BS will allocate resources to active UEs giving more priority to UEs with low playout buffer length to avoid rebufferings.

In our BB algorithm, the BS updates the scheduling metric for each UE every TTI. The scheduling metric considers the following,

- The last reported value of the playout buffer length
- An estimation of the video content length transmitted over the cellular channel since the last update
- A third part that takes into account the segments that are being transmitted by SMs over the D2D channel

To implement the above, the BB scheduler selects the user k' for transmission that has the maximum scheduling metric, as follows,

$$k' = \arg\max_k \{M_k^{BB}[n]\}, \tag{4}$$

where the BB scheduling metric, $M_k^{BB}[n]$, is calculated as follows,

$$M_k^{BB}[n] = V_{\max}[n] - \big(b_k[n] + p_{k,c}[n] + p_{k,d}[n]\big), \tag{5}$$

where $V_{\max}[n]$ is the current maximum video length, which is found as follows,

$$V_{\max}[n] = \max_k \{v_k[n]\}, \tag{6}$$

where $v_k[n]$ is the length of the current video played by user k. $b_k[n]$ is the playout buffer length in seconds for user k, as reported in the last update, and $p_{k,c}[n]$ is the recently transmitted playout time over cellular resources since the last update, calculated as follows,

$$p_{k,c}[n] = \frac{d_k[n]}{\zeta_k[n]}, \tag{7}$$

where $d_k[n]$ is the amount of video data transmitted to user k since the last update, and $\zeta_k[n]$ is the current video bit rate of the segment transmitted to user k over the cellular channel. $p_{k,d}[n]$ is used to take into account the segments that are being transmitted to user k via D2D communication since the last update from the UE, and it is calculated as follows,

$$p_{k,d}[n] = N_d \times w \times L \times \sigma_k, \tag{8}$$

where N_d is the number of segments currently transmitted over the D2D channel and w is the weight given for these segments, i.e., the ones the BS sent *Assistance Request* message for, but they are not received yet. L is the length of the segment in seconds. Since such segments are being transmitted now, they should be taken into account when allocating cellular resources. This allows the scheduler to early distinguish UEs that are being sent video segments over the D2D channel. However, this factor is only considered here when user k does not currently have a video segment scheduled over the cellular channel. Hence, the factor σ_k. If user k is currently not being sent a video segment over the cellular channel, σ_k will be set to 1 (σ_k is set to 0, Otherwise). This is because $p_{k,d}[n]$ significantly reduces the scheduling metric of user k. If the video segment that is currently scheduled over cellular resources precedes the one transmitted over the D2D channel, $p_{k,d}[n]$ might falsely indicate that user k has a long playout buffer, which results in long transmission delays.

From the above, we can see that the metric used for BB scheduling keeps the scheduler aware of the current playout buffer length at the UEs, and hence, avoid rebufferings by allocating more resources to users with imminent playout buffer stalling, and by saving the cellular resources for users who get their video segments through the BS only (over cellular resources). It is worth mentioning that as with PF scheduling, the BB algorithm can be implemented on RB-by-RB basis. In such case, $p_{k,c}[n]$ should be updated every time to keep track of the RBs assigned to each UE, as it is considered in the scheduling metric. However, it is computationally more efficient to implement the algorithm on a slot-by-slot basis (recall that a slot is half a TTI, i.e., 0.5 ms). In the latter case, if the RBs needed by the UE are fewer than the RBs in the slot (half TTI), the remaining RBs of the slot can be assigned to another UE to exploit all the RBs in the slot.

The work in this paper is focused on scheduling resources among video streaming users to improve video streaming QoE for such users in the cell, and avoid QoE degradation. However, in the case there are other types of traffic (i.e., background traffic) that share the same resources with video streaming users, the scheduler can employ BB scheduling to allocate resources among both video streaming traffic and other types of traffic. In such case, the scheduler can set a certain value for the playout buffer length of non-video streaming traffic. For example, a high playout buffer length value can be assigned to delay-tolerant data traffic so that it has lower priority than video streaming traffic. The used value of the playout buffer length assigned to delay-tolerant traffic can be used to adjust the priority of such traffic. The higher the assigned value for such traffic the lower priority it will have. This also guarantees that resources will be allocated to delay-tolerant traffic when video streaming users have a certain playout buffer length.

4.4 SM assignment

In DABAST, it is realistic to assume that SMs accept up to a certain number of concurrent assistance requests. As such, the BS should decide which SM should send video segments to which requesting UE. We refer to this as the SM-assignment problem. Here, we discuss how SMs are assigned the task of video segment transmission to requesting UEs in case of RR, PF, and BB cellular resource scheduling.

In the case of RR scheduling, every time SM assignment is performed, the BS goes through the active UEs in a RR fashion and looks for a UE that needs a video segment that is available in the distributed cache with an SM that is willing to assist. This means that SM assignment is performed in RR as well. In the case the BS finds more than one SM that is available to send a video segment to a requesting UE, the BS picks the SM that provided the least assistance so far, to achieve SM load-balancing.

In the case of PF scheduling, the recent average throughput of the requesting UEs and the average data rate between SMs and requesting UEs can be both used for allocating SMs to requesting UEs. This means that PF is used for SM assignment in a similar way cellular resources are allocated. Every time SM assignment is performed, the BS goes through the requesting UEs and selects the one with the highest ratio of the average data rate to an SM over the recent average throughput of that UE. This way, we assume that UEs periodically report to the BS an estimation of the average data rate that can be achieved if data is transmitted from a certain SM to this requesting UE as proposed in [41]. In the case the number of requesting UEs is high, this would cause much overhead. As such, the recent average throughput only can be utilized for SM assignment. This means that the BS goes through the requesting UEs in an ascending order based on the recent average throughput, and every time the BS finds a requesting UE that needs a cached segment with available SM, it will send assistance request to that SM. As with RR, in the case the average data rate between the UEs and SMs are not utilized, a load-balancing approach can be used to select among multiple available SMs.

In the case of BB scheduling, the UEs signal their playout buffer length to the BS. Since this valuable information is available at the BS, it can be further utilized for SM assignment. In this case, the BS assigns available SMs, giving more priority to UEs who have a low playout buffer. This further decreases the possibility of playout buffer depletion and consequently reduces the number of rebufferings for users in the cell, which potentially increases their QoE. When performing SM assignment, all the factors in (5) will be considered, i.e., $p_{k,d}[n]$ is always utilized, and hence is calculated as follows,

$$p_{k,d}[n] = N_d \times w \times L. \tag{9}$$

In the next section, we propose a more sophisticated SM-assignment approach that has more than one objective to further increase the QoE of users in the cell. In Sect. 6, we evaluate the improvements achieved by all the QoE-awareness techniques, including BB scheduling.

5 DABAST with HRC and SM-assignment optimization

In this section, we consider cases where the available resources are enough to avoid rebufferings. We aim to further increase the utilization of the D2D channel in such cases to improve video streaming QoE for users in the cell by improving their video bit rate. We present the first proposed technique; HRC. Afterwards, we present the second proposed technique, which is SM-assignment optimization.

5.1 HRC

As reported by many studies, video rebuffering has the highest impact on video streaming QoE, and hence, it should be avoided as much as possible. As such, when the traffic load is very high, reducing the number video rebufferings should be a priority. In these situations, video segments are downloaded with low video bit rate to avoid rebufferings. However, when the traffic load is lower, and users are not experiencing rebufferings, it would be very beneficial to send popular videos with high video bit rate. If segments of popular videos are cached with high video bit rate, they will be sent later to requesting users over the D2D channel, which will considerably increase video bit rate for these users, and hence, increase their video streaming QoE.

We propose DABAST with HRC, to further improve video bit rate for users in the cell. With HRC, segments of a video are sent in a video bit rate that is higher than the requested video bit rate. HRC is implemented by updating the operation of the CSVD algorithm, to consider one more case. We refer to this case as the HRC case. This case is implemented as follows,

1. The first condition that is needed to consider HRC is "low" traffic load. Otherwise, HRC will further increase the number of rebufferings in the cell and reduce video streaming QoE. This can be decided based on the number of rebufferings. For instance, the HRC mode can be activated in the case that none of the

users in the cell are experiencing video rebuffering, or if the number of rebufferings is below a certain threshold.
2. The second condition to employ HRC is if video segments are sent to an SM. In this case, segments of the video will be cached, and hence, will be sent later to requesting UEs in a high video bit rate. On the other hand, if the video is not going to be cached, only the video bit rate of this stream will increase, and this taken risk of sending a video with bit rate higher than the requested bit rate will not result in a significant reward.
3. HRC is only employed if the requested video is popular. This is explained as in the previous condition. If the video that is transmitted and cached with high video bit rate is not popular, the taken risk of sending a video with high video bit rate will not result in a significant reward. This is because if the file is not popular, it may not be requested often after this time, and consequently, will not be utilized by later requests. A video file can be considered popular enough for HRC if it is requested more than a certain number of times recently.
4. To avoid increasing the initial delay, the first few segments are always transmitted with a low video bit rate, even in HRC. The goal of HRC is to increase the video bit rate without causing considerable degradation to other video streaming QoE metrics.

HRC ensures that video segments of popular videos are cached with high video bit rate. This will diversify the content of the distributed cache in terms of video bit rate, i.e., some segments might be cached in more than one video bit rate. As such, the BS needs ensure that high rate segments are fully exploited, and at the same time, make sure that no degradation is induced to other video streaming QoE metrics. This can be achieved by carefully assigning SMs the tasks of sending video segments. This means that HRC introduces more complexity to the problem of SM assignment. One way to implement SM assignment in this case is by employing the same approach used with BB scheduling in the previous section. That approach assigns available SMs, giving more priority to UEs who have a low playout buffer, to minimize the number of rebufferings, and in the case more than one SM is available, the decision will be made based on load-balancing among SMs. However, this approach does not take into account the video bit rate of the cached segments. Such a video bit rate oblivious approach will not fully utilize the high video bit rate segments that are available in the distributed cache of the network.

In the following, we propose a better approach that takes into account both the playout buffer length at the clients and the video bit rate of the cached segments.

5.2 SM-assignment optimization

Here, we consider the problem of SM assignment in the case where there are many UEs requesting video segments that are cached with multiple video bit rates. An optimal SM-assignment approach, in this case, would maximize the achievable video bit rate in the cell, without causing a considerable increase in the number of rebufferings or in the initial delay. This should be achieved under the condition that an SM only accepts up to a maximum number of concurrent requests. We also assume that the BS sends up to a maximum number of assistance requests each time SM assignment is performed. SM assignment is performed by the SM-assignment module at the BS, which runs an optimization algorithm to find the best assignment. In the following, we formulate SM assignment as a Mixed Integer Linear Programming (MILP) problem.

Let us consider a cell with multiple clusters. C represents the set of clusters in the cell. Each cluster, $c \in C$, has a set of helping SMs, M_c. M_c contains SMs who have cached video segments that are currently requested by some UEs in the same cluster. A cluster also has a set of requesting UEs, Q_c, which contains the requesting UEs, for which, the currently requested segment can be provided by at least one SM in M_c. If G is the set of all video segments that can be requested, i.e., all the video segments of the current video streams, then each SM, $m \in M_c$, is caching a subset of the video segments, $G_m \subseteq G$. Moreover, each requesting UE, $q \in Q_c$, is currently requesting a segment $g_q \in G$. For each cluster, $c \in C$, S_c is the set of ordered pairs that represents the UE-SM combinations, in which, the SM has the segment requested by that UE. S_c can be represented as follows,

$$S_c = \{\langle q, m\rangle | q \in Q_c \wedge m \in M_c \wedge g_q \in G_m\}. \quad (10)$$

$S'_{c,z}$ is a subset of S_c, that contains only the ordered pairs that has z as the first item in the pair, where $z \in Q_c$. $S'_{c,z}$ can be expressed as,

$$S'_{c,z} = \{\langle q, m\rangle | q = z \wedge m \in M_c \wedge g_q \in G_m\}. \quad (11)$$

Similarly, $S''_{c,t}$ is a subset of S_c, that contains only the ordered pairs that has t as the second item in the pair, where $t \in M_c$. $S''_{c,t}$ can be expressed as,

$$S''_{c,t} = \{\langle q, m\rangle | q \in Q_c \wedge m = t \wedge g_q \in G_m\}. \quad (12)$$

The SM-assignment problem can be formulated as an MILP problem as follows,

$$\max \sum_{c \in C} \left(\sum_{s \in S_c} x_s \cdot R_s - \sum_{z \in Q_c} \left(\left(1 - \sum_{s \in S'_{c,z}} x_s \right) \cdot e^{\alpha / L_z} \right) \right), \tag{13}$$

s.t.

$$\sum_{s \in S'_{c,z}} x_s \leq 1, \quad \forall z \in Q_c, \forall c \in C, \tag{14}$$

$$\sum_{s \in S''_{c,t}} x_s \leq D_t, \quad \forall t \in M_c, \forall c \in C, \tag{15}$$

$$\sum_{c \in C} \left(\sum_{s \in S_c} x_s \right) \leq A, \tag{16}$$

$$x_s \in \{0, 1\}, \quad \forall s \in S_c, \forall c \in C, \tag{17}$$

where,

- x_s is the binary optimization variable for the *sth* UE–SM combination. It indicates whether this UE–SM combination will be selected, i.e., x_s is 1 if the SM in this combination is assigned to the requesting UE in this combination, and 0 otherwise.
- A is the maximum number of assistance requests that can be sent each time SM assignment is performed.
- R_s is the video bit rate of the segment for combination s, i.e., the video segment cached at the SM and requested by the UE in the combination.
- D_t is the maximum assistance requests that can be concurrently assigned to SM t.
- α is an optimization parameter.
- L_z is the current playout buffer length of requesting UE z.

Equation (13) shows the objective function to be maximized. The objective function is composed to two parts. The first part (before the minus sign) is to maximize the aggregate video bit rate. This can be achieved by assigning to each requesting UE the SM that is caching the requested segment with the highest video bit rate, which results in setting the binary variable of that combination, x_s, to 1. The second part of the objective function (after the minus sign) aims to minimizing the number of rebufferings, by ensuring that the playout buffer of the UEs is above a certain level. The parameter α decides how aggressive the optimizer is in favoring playout buffer depletion avoidance over maximizing the aggregate video bit rate. It controls the playout buffer level the optimizer tries to maintain before it starts assigning SMs to UEs solely to maximize the average video bit rate.

As can be seen from the equation, the first term (that maximizes the aggregate video bit rate) is usually much larger than $\sum_{z \in Q_c} \left(\left(1 - \sum_{s \in S'_{c,z}} x_s \right) \right)$, because the first term includes the video bit rate of each combination, R_s, which usually has high values (minimum value used for R_s is 384 kbps). As such, the exponential term, e^{α / L_z}, is used in the second part. This makes the values of the two parts of the objective function comparable and allows controlling the overall objective (playout buffer depletion avoidance vs. maximizing the aggregate video bit rate) via the parameter α. In the case there is a UE with a low playout buffer length, the small value of L_z will significantly increase the value of the exponential term. As such, the optimizer will give high priority to assigning an SM to that UE. However, when all the UEs have a relatively high playout buffer length, the values of the exponential term will be small, and hence, the optimizer will focus only on the first term of the objective function which maximizes the aggregate video bit rate.

From the above, we can see that this results in dynamically maximizing video bit rate. When there are UEs with low playout buffer that is below a certain threshold (decided by the value of α), increasing the playout buffer length of these UEs will have higher priority than maximizing video bit rate. However, if the playout buffer length of the UEs is above that threshold, the highest priority will be to maximize the video bit rate. In addition to maximizing the aggregate video bit rate dynamically while avoiding rebuffering, a considerable increase in the initial delay can also be avoided by controlling the value of α. This is because further increase to the value of α gives more priority to filling up the playout buffer of clients.

The first constraint which is imposed by Eq. (14) states that every time SM assignment is performed, a maximum of 1 SM can be assigned to any UE. The second constraint shown in Eq. (15) ensures that the number of requests sent to any SM does not exceed the maximum concurrent requests of that SM. The third constraint, which is imposed by Eq. (16) states that every time SM assignment is performed, the number of requests should not exceed a certain value. Equation (17) specifies the variable bounds of the optimization problem.

The above SM-assignment optimization problem is an MILP with $\sum |S_c|$ variables, where S_c is the set of UE–SM combinations in cluster c. S_c only contains the combinations where the SM can provide the current video segment requested by that UE. If S_c contains many combinations, solving the above problem might not be feasible in a TTI time. However, SM assignment does not have to be performed every TTI like cellular resource allocation. As SM assignment in DABAST is done on a segment-by-segment basis, the SM-assignment optimization problem can be performed on a scale of tens of milliseconds. Fortunately, there are many commercial solvers that can solve the above problem quickly. We employ Gurobi, a commercial

optimization solver [42], to solve the above optimization problem during simulations. Gurobi uses the Linear Programming (LP)-based branch-and-bound algorithm to solve MILP problems [42]. Furthermore, it employs many techniques to speed up execution time of the LP-based branch-and-bound algorithm, such as pre-solving, cutting planes, heuristics, and parallelism. We measured the execution times of the SM-assignment optimization problem with Gurobi. We present and discus the execution-time results in Sect. 6.3.

6 Performance evaluation of QoE-aware DABAST

6.1 Performance evaluation of DABAST

We used the DEVS formalism [43] to build a model for DABAST in an LTE-A network and implemented our DEVS model with the CD++ toolkit [43]. For detailed description of the developed DEVS model, the reader is referred to [12]. System-level simulations were performed using the developed simulator to evaluate the performance of DABAST in terms of many video streaming QoE metrics, and compare it to the performance of a conventional DASH system (without D2D communication). The simulation setup is shown in Table 1. The parameters in Table 1 are taken from the LTE-A standard [44]. An LTE-A cell with high user density is considered in the simulations. As previously mentioned, out-of-band D2D communication is employed here. As such, no interference between cellular and D2D communication is assumed, as each communication takes place on a separate band. The urban macro propagation model [44] 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 was used for D2D transmission [45]. A 60 MHz out-of-band channel is employed for D2D communication. The LTE-A protocol stack was considered for cellular and D2D communication [46], and all frequency resources are allocated by the BS. Furthermore, multi-radio access techNologies (RATs), dual connectivity, and flow splitting/aggregation [47, 48] are employed to send multiple video segments to a user (from more than one source) simultaneously, when possible.

In the beginning of each iteration of the simulations, the UEs are located throughout the cell with a uniform distribution. Then, the cell is divided into 9 clusters, where UEs are assigned to the clusters and SMs are selected. During each iteration, UEs randomly request and watch video streams from a list of 500 videos. The relative popularity of videos is modeled with the Zipf distribution. It is shown in [49] that this is a suitable model for this purpose. As such, some videos are more popular, and hence, have higher probability of being requested. 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. The arrival of requests is generated according a Poisson arrival process. The videos are available in four video bit rate levels as in Table 1. These video bit rates are adopted from the H.264/AVC video coding standard [50]. The length of the videos is 441 s, which is the mean length of a YouTube video [51]. Table 2 shows the mapping between the playout buffer length to the available video bit rates. This is used by the buffer-based video bit rate adaption approach that is employed by the DASH controller at the clients. Round Robin (RR) is used for cellular resource scheduling. The number of UEs in the cell is 500 and the Zipf exponent value is 1.5.

Table 1 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
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 (s)	10
Number of buffered segments to start playout	4
Video bit rate levels (kbps)	384, 768, 2000, 4000
Videos length (s)	441

We measure all the video streaming QoE metrics discussed in Sect. 2, i.e., the number of rebufferings, video continuity index, initial delay, and video bit rate. Table 3 shows the mean values of these measurements from 50 simulation runs and the Margin of Error (MoE) for 95% confidence interval.

As one can see in Table 3, clear improvement is achieved by DABAST over conventional DASH in terms

of all the measured QoE metrics. For instance, results show that DABAST achieved 50% reduction in the average number of rebufferings. This is a significant improvement in terms of the number of rebufferings, which significantly improves video streaming QoE for users in the cell. The reduction in the number of rebufferings and in the rebuffering time achieved by DABAST significantly improved the video continuity index.

The results also show that DABAST decreased the average initial delay by 50%, which is also a significant improvement. In this scenario, the average initial delay for conventional DASH is high because there are 500 UEs in the cell requesting video streams and sharing fixed cellular frequency resources (10 MHz). This is also because video playout starts after receiving 4 video segments. In addition to improving all the metrics above, DABAST also increased the average video bit rate. With DABAST, video segments are sent to requesting UEs from both the BS over the cellular channel and from the SMs over the out-of-band D2D channel, thanks to the CSVD algorithm. With conventional DASH, on the other hand, video segments are only delivered to requesting UEs through the BS over the cellular channel. As such, DABAST significantly relaxes the RAN bottleneck (reduces congestion) and improves the data rates at which video segments are transmitted to requesting UEs when compared to conventional DASH, which decreases the transmission delays of video segments. This reduces the likelihood of video buffer stalling, and consequently, reduces the number of rebufferings. This also improves the delay to transmit the first 4 segments needed to start playout, and hence, improves the initial delay.

We have also measured the number of video segments transmitted over the cellular channel and over the D2D channel. The results have shown that about half of the video segments (49.98%) were transmitted over the D2D channel. This shows how effective DABAST can be and explains the significant improvements achieved by DABAST.

Figure 2 depicts the relative frequency histogram of the number of rebufferings for both conventional DASH and DABAST. One can see that with DASH, over 96% of the video streams have either 3 or 4 rebufferings. The figure also shows that with DABAST, half of the video streams have 0 rebufferings, and slightly less than half of the streams have 3 or 4 rebufferings. This explains the significant improvement achieved by DABAST in terms of the average number of rebufferings (Table 3). Even in the case of DABAST, there are users who experience 3 and 4 rebufferings. This is because we consider the case where there are no video segments are cached in the beginning, and that video segments are cached progressively as requested. However, after video segments accumulate in the distributed caches in the cell, many video segments will be sent from the SMs over the D2D channel.

These segments will be transmitted faster to the requesting UEs which prevents rebufferings. Furthermore, as many of the segments are now sent over the D2D channel, more cellular resources will be available for video streams that receive segments exclusively through the BS. This relaxes the RAN bottleneck and reduces the transmission delays for such segments, which reduces the possibility of playout buffer depletion, decreasing the number of rebufferings by 50%.

The number of received video segments with each video bit rate is shown in Table 4 for conventional DASH and DABAST. One can see that in the case of DABAST, fewer video segments were received with the lowest video bit rate (384 kbps) and more video segments were received with higher video bit rates (768, 2000, and 4000 kbps). As such, the average video bit rate of DABAST is higher than that of conventional DASH, as shown in Table 3. In the case of DABAST, some clients receive video segments from both the BS and the SMs. These clients usually have longer video playout buffer length because they receive video segments with higher transmission rates. As such, these clients request video segments with higher video bit rates, which explains why some video segments were received with higher video bit rates in the case of DABAST. However, one can see from the results in Tables 3 and 4 that only a small improvement is achieved by DABAST in terms of the video bit rate. As this is a high traffic load scenario, the average data rates for users in the cell are low,

Table 2 Playout buffer length-video rate mapping

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

Table 3 Simulation results

	Conventional DASH		DABAST	
	Mean	MoE	Mean	MoE
Rebufferings	3.4448	0.0179	1.7272	0.0164
Cont. index	0.7447	0.0009	0.8699	0.0001
Initial delay (s)	56.881	0.2200	28.654	0.4821
Video bit rate (kbps)	397.27	0.3267	430.16	1.3694

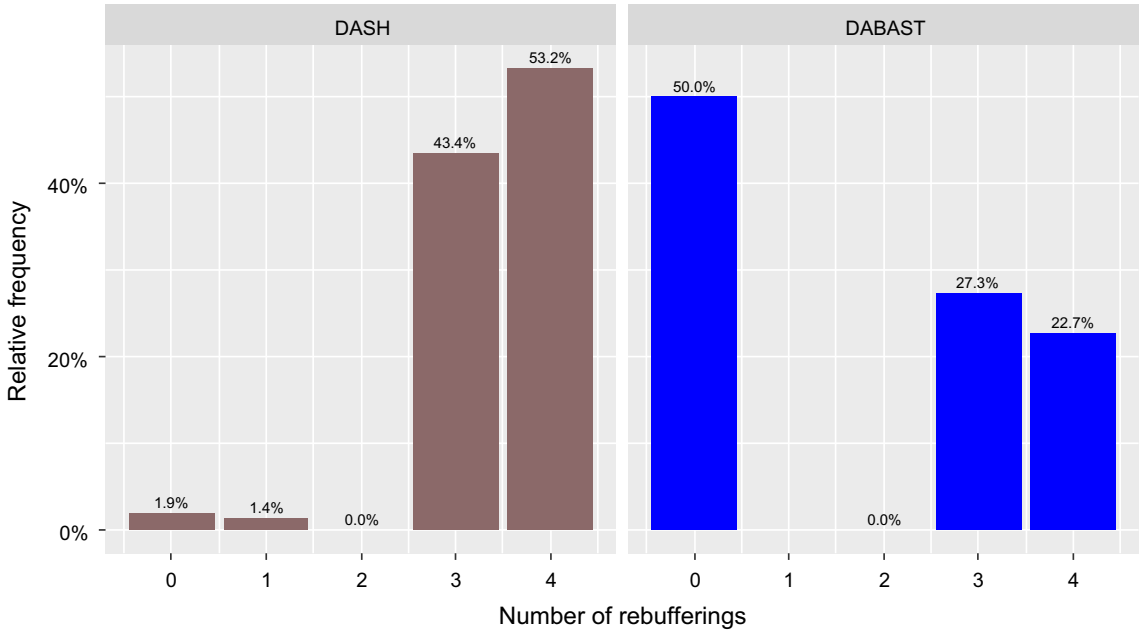


Fig. 2 Relatve frequency histogram of the number of rebufferings for conventional DASH and DABAST

and consequently, their playout buffers are low. Hence, video segments are usually received and cached in the SMs with the lowest video bit rate. Furthermore, under high traffic load, the BS sends a cached video segment from the distributed cache even if the video bit rate of the cached segment does not match that requested by the client. As previously mentioned, this is to increase the utilization of the D2D channel and reduce the RAN bottleneck, which reduces the number of rebufferings. Hence, despite the increase in the data rates and playout buffer lengths achieved by DABAST for some clients (which increases the requested video rate), these clients still receive most segments with low video bit rate (from the distributed cache). Only when caching SMs are not available for assistance, cached segments are received through the BS with higher video bit rates.

We would like to shed some light into the feasibility of using UEs as SMs in the cell. Each SM in the simulations above caches two videos. Given that the average video bit rate delivered to UEs is 430.16 Kbps (Table 4), the needed storage capacity by each SM is about 370 Megabytes (not even 1 Gigabyte). This is very affordable considering that smart devices nowadays have storage capacity of multiple Gigabytes (e.g., 32–128 GB).

Table 4 Count of the received segments with each video bit rate

Video bit rate (kbps)	Count	
	Conventional DASH	DABAST
384	2,207,508	2,077,111
768	31,494	149,365
2000	10,998	19,273
4000	0	4251

6.2 QoE-aware cellular resouce allocation

We ran simulations to evaluate the performance of DABAST in terms of video streaming QoE with all the scheduling algorithms discussed in Sect. 4 (RR, PF, and BB). We measure the same QoE metrics used in Sect. 6.1. The simulation scenario and the setup in Sect. 6.1 (Table 1 and Table 2) are also used here. Table 5 shows the mean values of the measured QoE metrics along with the MoE values for 95% confidence interval. The values in Table 5 show the mean of all the average values from 50 simulation runs. The results in Table 5 are for 500 UEs in the cell, Zipf exponent of 1.5, and 500 videos.

When comparing the results for DABAST with RR and PF scheduling, it can be seen that PF scheduling reduces the average number of rebufferings by 0.2848 (16.49% reduction), which is a considerable improvement. Employing PF scheduling also caused a slight improvement in the continuity index and in the average video bit rate. With PF scheduling, the scheduling metric considers the ratio of the instantaneous data rate of the UE to the recent average throughput of the UE. The recent average throughput considers the data transmitted over both the cellular and D2D resources. By maintaining the recent average throughput of each UE, we consider the video data

Table 5 Simulation results (RR vs. PF vs. BB)

	DABAST-RR		DABAST-PF		DABAST-BB	
	Mean	MoE	Mean	MoE	Mean	MoE
Rebufferings	1.7272	0.0164	1.4424	0.0127	0.8787	0.1195
Cont. index	0.8699	0.0001	0.8923	0.0009	0.9258	0.0079
Initial delay (s)	28.654	0.4821	31.588	0.3159	28.376	0.7506
Video bit rate (kbps)	430.16	1.3694	433.47	0.8354	428.72	1.6469

that is transmitted over both the cellular and the D2D channel. Hence, for UEs that received segments over the D2D channel, the recent average throughput will be relatively high, which consequently leads to favoring UEs with lower recent average throughput. Usually, these are UEs with low playout buffer and UEs that receive their video segments exclusively over the cellular channel. In this scenario, where many UEs are sharing the cellular channel, it would be beneficial to dedicate the limited cellular resources to UEs that can only get their video segments over the cellular channel. This will increase the cellular resources for these users, and further relax the bottleneck of the RAN, which reduces the number of rebufferings and increases the continuity index.

Although PF scheduling keeps track of the recent average throughput for each UE, which helps in reducing the average number of rebufferings, it is oblivious to the current playout buffer length of users. There are many cases where a high value of the recent average throughput might falsely indicate a high playout buffer length value. A general case is when a video segment with high video bit rate is transmitted to a UE. Although in such case the UE experienced high recent throughput, this was used to send relatively less video content (in s) due to the high video bit rate of the segment. There is a different and more complicated case that is specific to DABAST. With DABAST, a video segment might be sent over the D2D channel when found in the distributed cache, although the previous video segment is still being transmitted over the cellular channel. This speeds up transmission of video segments and keeps the playout buffer length consistently high. It is worth explaining that in such case, the previous segment was set for transmission over the cellular channel due to unavailability of SMs at the time this decision was made. In this case, the UE will have a high recent average throughput because a segment is transmitted over the D2D channel. Hence, the PF scheduling metric for this UE will be low, although the previous video segment is still being transmitted over the cellular channel. A low scheduling metric means that the PF scheduler will give low priority to this UE when allocating cellular resources. This usually does not result in rebuffering because such UEs (playing videos cached in the SMs) have relatively higher playout buffer as most segments are delivered over the D2D channel. Hence, the segment transmitted over the cellular channel arrives before consumption of the playout buffer. However, this might result in high initial delay when the UE is awaiting the initial video segments needed to start playout. This explains why the average initial delay is higher for DABAST with PF scheduling.

Table 5 shows that employing BB scheduling achieved significant improvement in terms of rebuffering. BB scheduling achieved 49.13% and 39.08% reduction in the average number of rebuffering over RR and PF, respectively. Table 5 shows that BB scheduling also significantly increased the continuity index. BB scheduling achieved these improvements in terms of rebuffering, while achieving a slight improvement in the average initial delay. With BB scheduling, the scheduler considers the reported playout buffer length as well as an estimation of the length of transmitted video contents. As such, the scheduler will be always aware of the current playout buffer length at the UEs, and hence, avoid rebufferings by allocating more resources to users with low playout buffer. This explains the improvement achieved by BB scheduling in terms of the number of rebufferings and the continuity index. Because BB scheduling keeps track of the actual playout buffer length, it will allocate resources to a UE if it currently has a relatively low playout buffer, even if the next segment is transmitted over the D2D channel. As such, BB scheduling does not increase the initial delay as with PF scheduling.

Table 5 shows that BB scheduling results in a very slight reduction in the average video bit rate. BB scheduling has 428.72 kbps average video bit rate, which is only 0.33% and 1.1% reduction when compared to RR and PF scheduling, respectively. This is a negligible reduction that is not even noticeable by the end user. This reduction is expected as BB scheduling utilizes more of the cellular resources to help users who get their video segments exclusively through the BS, and hence fewer segments will be sent with higher video bit rate to users with higher playout buffer, i.e., users who get some of their segments over the D2D channel.

Figure 3 depicts the relative frequency histogram of the number of rebufferings for DABAST with RR, PF, and BB resource allocation. When comparing RR and PF scheduling, one can notice that with PF scheduling, more

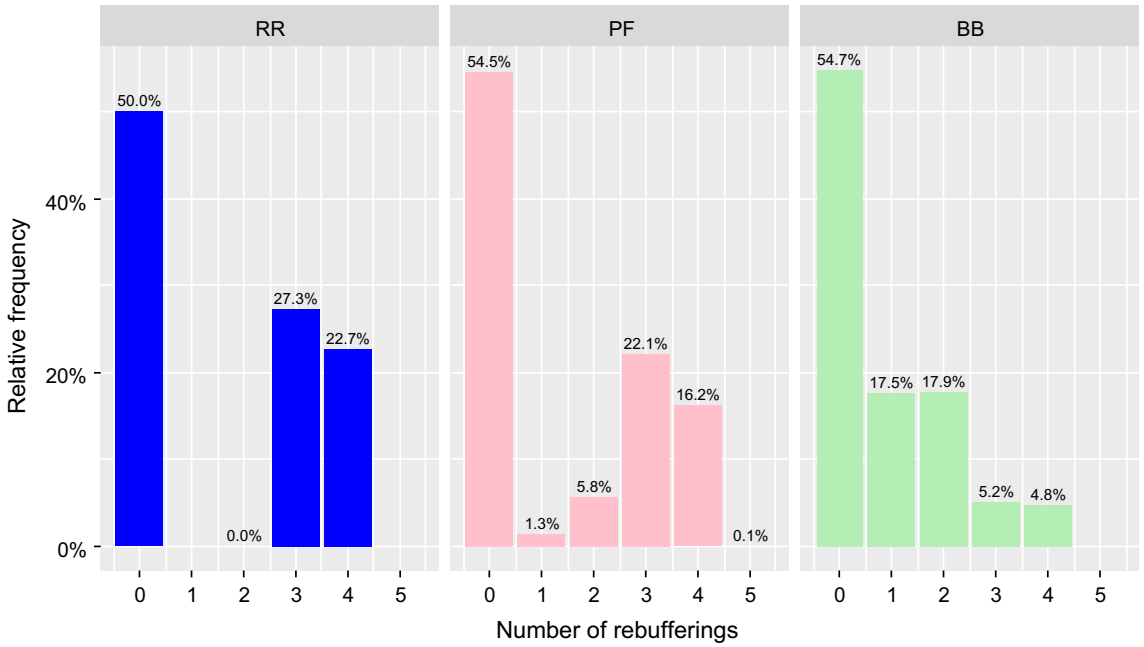


Fig. 3 Relative frequency histogram of the number of rebufferings for DABAST with RR, PF, and BB resource allocation

video streams have 0, 1, and 2 rebufferings, and fewer video streams have 3 and 4 rebufferings. This explains the improvement achieved with PF over RR scheduling in terms of the average number of rebufferings. However, one can see from Fig. 3 that with PF scheduling, there is a very small number of streams (0.1%) that have 5 rebufferings. As discussed above, this is because the PF scheduler is oblivious to the current playout buffer length of the users in the cell, and in some cases the higher value of the recent average throughput does not necessarily indicate a longer playout buffer. As such, the PF scheduler might ignore such users and cause them to experience 5 rebufferings.

Figure 3 shows that BB scheduling significantly reduced the number of streams with 3 and 4 rebufferings, and eliminates the case of 5 rebufferings. This explains the significant improvement achieved by BB scheduling in terms of the average number of rebufferings, and shows the importance of the playout buffer length awareness at the scheduler.

As mentioned in Sect. 6.1, in each simulation run, each UE requests and watches two videos consecutively. After playout of the first video, a UE stays idle for a random period of time, and then generates another request for a video stream. Figure 4 shows the relative frequency histogram for the number of rebufferings of each request for DABAST with RR, PF, and BB resource allocation. The figure shows that all the rebufferings take place during the first set of video streams. This is because at the time most of the first 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. This explains the high number of rebufferings experienced by user in the first set of video streams. All the streams in the second set 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 the D2D channel, which improves the data rates and eliminates rebufferings, as previously discussed.

Figure 5 shows the histogram of the continuity-index for DABAST with RR, PF, and BB resource allocation. As can be seen from the figure, although PF scheduling increased the continuity index value for many video streams when compared to RR scheduling, there is a small number of streams that have continuity index values less than 0.7, which is the lowest value with RR scheduling. These are the users who experience 5 rebufferings, as explained above. With BB scheduling on the other hand, the continuity index value for many streams have increased, and the minimum value did not decrease (still at 0.7).

6.3 HRC and SM-assignment optimization

Simulations were executed to evaluate the performance of DABAST with the techniques discussed in Sect. 5. We consider 3 versions of DABAST; DABAST with BB scheduling (BB), DABAST with BB scheduling and HRC (BB-HRC), and DABAST with BB scheduling, HRC, and SM-Assignment optimization (BB-HRC-SMA). The

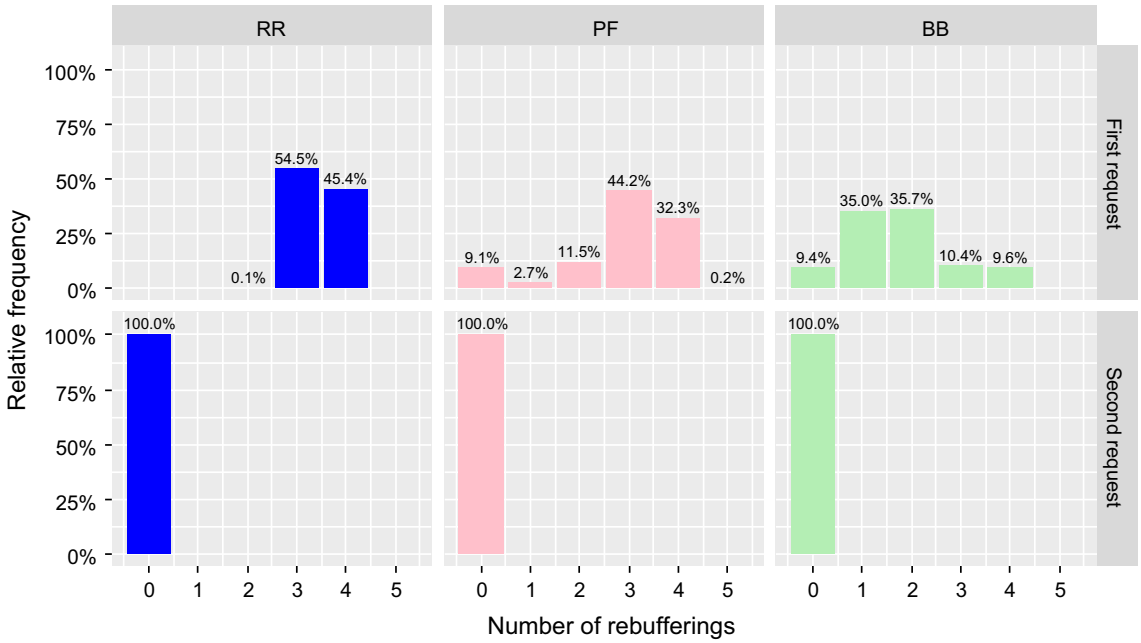


Fig. 4 Relative frequency histogram of the number of rebufferings in each request for DABAST with RR, PF, and BB resource allocation

simulation setup in the previous sections (Tables 1, 2) was also used here. As previously discussed in Sect. 5, we consider the case of HRC when the traffic load is not high enough to cause reoccurring video buffer stalling for users in the cell. As such, we consider a cell with 300 UEs in the simulations. Moreover, a Zipf exponent of 1.5 and 500 videos are considered. We measure the same QoE metrics used in the previous sections. Table 6 shows the mean values along with the MoE values for 95% confidence interval of the measured QoE metrics. The table shows the values for the three versions of DABAST, i.e., BB, BB-HRC, and BB-HRC-SMA. The average value for each simulation run was calculated. The values below show the mean of all the average values from 120 simulation runs.

Table 6 shows that for BB and BB-HRC-SMA, the average number of rebufferings is 0, while BB-HRC resulted in a negligible increase in the number of rebufferings. Table 6 shows that for BB-HRC, the average

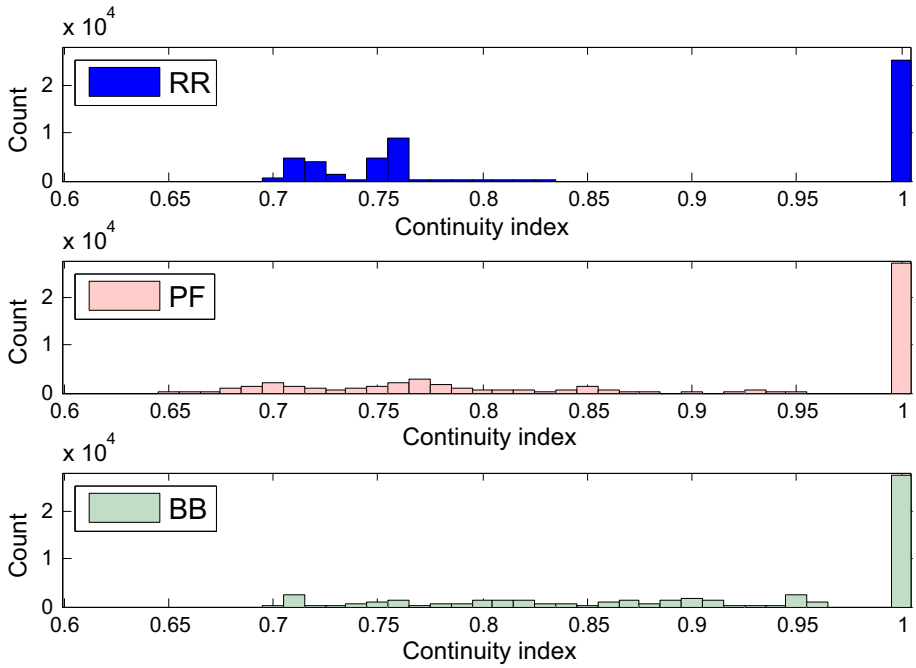


Fig. 5 Histogram of the continuity index for DABAST with RR, PF, and BB resource allocation

Table 6 Simulation results for DABAST (BB vs. BB-HRC vs. BB-HRC-SMA)

	BB		BB-HRC		BB-HRC-SMA	
	Mean	MoE	Mean	MoE	Mean	MoE
Rebufferings	0.0	0.0	0.0036	0.0004	0.0	0.0
Cont. index	1.0	0.0	0.9997	0.0003	1.0	0.0
Initial delay (s)	19.280	0.1457	19.580	0.1694	20.222	0.1474
Video bit rate (kbps)	459.06	0.9697	635.10	4.6248	755.00	5.8524

number of rebufferings is 0.0036. A further inspection of the results of BB-HRC has shown that the maximum number of rebufferings experienced by a UE is 1. This means that out of the 300 UEs, about one UE might experience 1 rebuffering. Regarding BB, it is expected to have the least average number of rebufferings, as it does not employ HRC, which means no video segments are downloaded with high video bit rate. When comparing BB-HRC and BB-HRC-SMA, we can see that in addition to significantly improving the video bit rate, BB-HRC-SMA completely avoided any rebufferings. Thanks to SM-assignment optimization, where every time SM assignment is performed, the SMs are assigned in parallel to the UEs, i.e., the optimizer has a global view of the available SMs and the requesting UEs. With BB-HRC, on the other hand, SMs are assigned to UEs sequentially. The assignment module goes through the requesting UEs in a descending order of their scheduling metric (ascending order of their playout buffer length), and each time allocates to that UE the available SM (if any) with the least load, to achieve load-balancing between SMs. This way, the SM-assignment module might assign to a UE the SM with the least load among the available SMs, although that SM could be the only one available for the next requesting UE. This explains why BB-HRC caused a very slight increase in the average number of rebufferings.

In addition to avoiding rebufferings, BB-HRC-SMA significantly improved the average video bit rate over other versions of DABAST. The results show that BB-HRC-SMA achieved 295.94 kbps (64.47%) and 119.9 kbps (18.88%) increase in the average video bit rate over BB and BB-HRC, respectively.

Regarding the average initial delay, we can see that HRC has caused a very slight increase in the average initial delay. The results show that BB-HRC caused only 0.3 s (1.56%) increase in the initial delay, and BB-HRC-SMA caused only 0.94 s (4.88%) increase in the initial delay over BB. This is expected as many segments with higher video bit rates are transmitted with HRC. The slight increase in initial delays can be explained as follows. Although with HRC the first few segments are still sent with low video bit rate, other (subsequent) segments are transmitted with high video bit rates over cellular resources. This increases the amount of data to be transmitted over cellular resources and increases the contention for cellular resources, which causes slightly higher initial delays for other users that are currently receiving the first few segments. However, this is a small price to pay considering the significant improvement achieved in terms of the average video bit rate.

Figure 6 depicts the Empirical Cumulative Distribution Function (ECDF) of the initial delay for each version of DABAST. From the figure, one can see that all versions of DABAST have very close distribution for the initial delay, with a slight difference in the ECDF of BB-HRC-SMA for values lower than 25 s. This means that the slight increase in initial delay caused by HRC is not experienced by a small group of UEs, but rather distributed over the UEs in the cell. Thanks to BB scheduling, where cellular resources are allocated to UEs with low playout buffer, which distributes the extra delay caused by transmission of high video bit rate segments over all UEs in the cell sharing cellular resources.

We calculated the average maximum initial delay, which is the average maximum delay from all the simulation runs. The average maximum delay values for BB, BB-HRC, and BB-HRC-SMA are 35.31, 35.561, and 35.459, respectively. This means that HRC only caused a negligible increase in these values, which further shows the importance of BB scheduling.

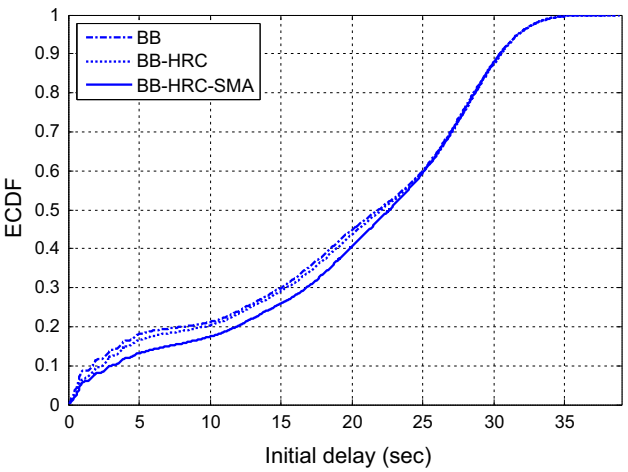


Fig. 6 ECDF of the initial delay for DABAST (BB vs. BB-HRC vs. BB-HRC-SMA)

Table 7 Count of the received segments with each video bit rate

Video bit rate (kbps)	Count		
	BB	BB-HRC	BB-HRC-SMA
384	2,606,646	2,360,752	2,121,270
768	633,354	492,933	491,777
2000	0	386,315	626,953
4000	0	0	0

Table 7 shows the number of video segments received with each video bit rate, for the different versions of DABAST (BB, BB-HRC, and BB-HRC-SMA). When comparing the results of BB and BB-HRC, one can see the with BB-HRC, fewer video segments are received with the lower video bit rates (384 and 768 kbps), and 386,315 video segments are received with high video bit rate (2 Mbps). These are the segments of popular videos that are downloaded and cached in the SMs with high video bit rate. When comparing the results of BB-HRC and BB-HRC-SMA, one can see that with BB-HRC-SMA, 240,638 more video segments are received with video bit rate of 2 Mbps. Most of these 240,638 video segments are received with the lowest video bit rate in the case of BB-HRC. This explains the further improvement achieved by BB-HRC-SMA over BB-HRC in terms of the average video bit rate.

As previously mentioned, we employ Gurobi, a commercial optimization solver, to solve the SM-assignment optimization problem during simulations. We measured the execution times of the SM-assignment optimization problem with Gurobi for the simulations in this subsection (Sect. 6.3). The used machine has a Quad-core Intel i7 processor with a speed of 3.6 GHz $\times$ 8 threads. Measurements have shown that the maximum execution time of the problem was 10 ms, which is adequate for the time scale of SM assignment (tens of milliseconds as previously discussed). We also measured the execution times for the solver with 500 UEs. The maximum execution time in such scenario goes up to 20 ms, which is still adequate for the time scale of SM assignment. However, as discussed in Sect. 5, HRC and SM-assignment optimization are not employed under such scenario with high traffic load.

Finally, let us again have look at the storage capacity needed by SMs in this case. More specifically, we are interested in the HRC case, where video segments are cached in high video bit rate, i.e., 2 Mbps. As mentioned above, the first 4 segments are always cached with the lowest video bit rate, while the remaining segments are cached with high video bit rate (2 Mbps). Recall that each SM stores 2 video files. As such, the storage capacity needed by each SM in this case is 1596.4 MB (1.56 GB). As discussed in Sect. 6.1, this amount of storage capacity is affordable nowadays, considering that smart devices have storage capacity of multiple Gigabytes (e.g., 32–128 GB).

7 Conclusion

The increasing popularity of video streaming is escalating the growth of data traffic to be transmitted over cellular networks and raising the challenge for cellular network operators. Consequently, supporting video streaming services while achieving user satisfaction about the video service has become a main concern for cellular network operators. As such, it is necessary not only to develop new techniques to improve the delivery of video streaming traffic, but also for the new techniques to consider the complex, dynamic, and delay-sensitive nature of video streaming traffic to provide the end users with good quality of experience (QoE) video streaming. In this work, we present our proposed architecture for improving the QoE of video streaming over cellular networks with high user density. The architecture employs the cached and segmented download algorithms, which provide base-station (BS)-assisted progressive caching of video segments in storage members (SMs) and device-to-device (D2D) video transmission in cellular networks. Dynamic adaptive streaming over HTTP (DASH) is also employed by the architecture to allow adaptive video streaming. The architecture is called **DA**SH-based **B**S-**A**ssisted D2D video **ST**reaming in cellular networks (DABAST). We evaluate the performance of DABAST, through computer simulations, in terms of video streaming QoE metrics. Results show that DABAST achieves significant improvements in terms of QoE when compared to conventional video streaming over a cellular network, i.e., DASH streaming without collaborative D2D communication.

We improve the operation of DABAST by introducing QoE awareness to both caching and distribution of video segments. We employ QoE awareness in three aspects of DABAST; cellular resource allocation, caching of video segments, and SM-assignment optimization. We analyze the performance achieved by each QoE-awareness technique in terms of video streaming QoE metrics. We provide a thorough analysis of the results which show that all the proposed QoE-awareness techniques significantly improve the performance of DABAST.

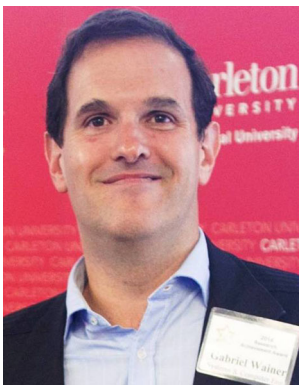
Acknowledgements The authors would like to thank Dr. Stenio Fernandes from the Federal University of Pernambuco, Brazil, for his valuable assistance during this work.

References

1. Ooyala (2018) Ooyala's Q4 2017 Global Video Index. http://go.ooyala.com/wf-video-index-q4-2017. Accessed 10 Oct 2018.
2. Sandvine (2016) 2016 Global internet phenomena: Latin America and North America. Technical report.
3. Brunnström, K., Beker, S. A., De, K., et al. (2014). *Qualinet white paper on definitions of quality of experience*. European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Lausanne, Switzerland.
4. Seufert, M., Egger, S., Slanina, M., et al. (2015). A survey on quality of experience of HTTP adaptive streaming. *IEEE Communications Surveys & Tutorials, 17,* 469–492.
5. ITU. (2016). Vocabulary for performance and quality of service. ITU-T Recommendation P.10, Amendment 5.
6. El, Essaili A., Schroeder, D., Steinbach, E., et al. (2015). QoE-based rraffic and resource management for adaptive HTTP video delivery in LTE. *IEEE Transactions on Circuits and Systems for Video Technology, 25,* 988–1001.
7. Cicalo, S., Changuel, N., Tralli, V., et al. (2016). Improving QoE and fairness in HTTP adaptive streaming over LTE network. *IEEE Transactions on Circuits and Systems for Video Technology, 26,* 2284–2298.
8. Huang, T.-Y., Johari, R., McKeown, N., et al. (2014). A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *Proceedings of the 2014 ACM SIGCOMM* (pp. 187–198). New York, USA: ACM.
9. De Cicco, L., & Mascolo, S. (2014). An adaptive video streaming control system: Modeling, validation, and performance evaluation. *IEEE/ACM Transactions on Networking, 22,* 526–539.
10. Keller, L., Le, A., Cici, B., et al. (2012). MicroCast: Cooperative video streaming on smartphones. In *the 10th international conference on mobile systems, applications, and services* (pp 57–70). New York, USA.
11. Detti, A., Ricci, B., & Blefari-Melazzi, N. (2015). Mobile peer-to-peer video streaming over information-centric networks. *Computer Networks, 81,* 272–288.
12. Al-Habashna, A., Wainer, G., & Fernandes, S. (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* (pp. 468–475). Seattle, USA: IEEE.
13. Al-Habashna, A., Wainer, G., Boudreau, G., & Casselman, R. (2015). Improving wireless video transmission in cellular networks using D2D communication.
14. Al-Habashna, A., Wainer, G., Boudreau, G., & Casselman, R. (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* (pp. 473–480). Montreal, Canada: IEEE.
15. Al-Habashna, A., & Wainer, G. (2017). Improving video transmission in cellular networks with cached and segmented video download algorithms. *Mobile Networks and Applications, 23,* 543–559.
16. Li, B., Wang, Z., Liu, J., & Zhu, W. (2013). Two decades of internet video streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications, 9,* 1–20.
17. Stockhammer, T. (2011). Dynamic adaptive streaming over HTTP: Standards and design principles. In *Proceedings of the 2nd annual ACM conference on multimedia systems* (pp. 133–144). New York, USA: ACM Press.
18. DASH Industry Forum. (2013). Guidelines for implementation: DASH-AVC/264 interoperability points. http://dashif.org/wp-content/uploads/2015/04/DASH-AVC-264-base-v1.03.pdf. Accessed 28 Feb 2018.
19. Tian, G., Liu, Y. (2012). Towards agile and smooth video adaptation in dynamic HTTP streaming. In *Proceedings of the 8th international conference on emerging networking experiments and technologies* (pp. 109–120). Nice, France: ACM Press.
20. De Cicco, L., Caldaralo, V., Palmisano, V., & Mascolo, S. (2013). ELASTIC: A client-side controller for dynamic adaptive streaming over HTTP (DASH). In *20th international packet video workshop* (pp. 1–8). San Jose, USA: IEEE.
21. Conviva. (2013). Conviva releases viewer experience report. https://www.conviva.com/press-releases/conviva-releases-viewer-experience-report/. Accessed 10 June 2018.
22. Mok, R. K. P., Chan, E. W. W., Luo, X., & Chang, R. K. C. (2011). Inferring the QoE of HTTP video streaming from user-viewing activities. In *Proceedings of the 1st ACM SIGCOMM workshop on measurements up the stack* (pp. 31–36). Toronto, Canada: ACM Press.
23. Hossfeld, T., Egger, S., Schatz, R., et al. (2012). Initial delay vs. interruptions: Between the devil and the deep blue sea. In *4th international workshop on quality of multimedia experience* (pp. 1–6.). Yarra Valley, Australia: IEEE.
24. Qi, Y., & Dai, M. (2006). The effect of frame freezing and frame skipping on video quality. In *2006 international conference on intelligent information hiding and multimedia* (pp. 423–426). Pasadena, USA: IEEE.
25. Sackl, A., Egger, S., & Schatz, R. (2013). Where's the music? Comparing the QoE impact of temporal impairments between music and video streaming. In *5th international workshop on quality of multimedia experience* (pp. 64–69). Klagenfurt am Wörthersee, Austria: IEEE.
26. 3GPP. (2015). Evolved universal terrestrial radio access: Physical channels and modulation. Technical report TS 36.211.
27. Agiwal, M., Roy, A., & Saxena, N. (2016). Next generation 5G wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials, 18,* 1617–1655.
28. Asadi, A., Wang, Q., & Mancuso, V. (2014). A survey on device-to-device communication in cellular networks. *IEEE Communications Surveys & Tutorials, 16,* 1801–1819.
29. Kaufman, B., & Aazhang, B. (2008). Cellular networks with an overlaid device to device network. In *the 42nd Asilomar conference on signals, systems and computers* (pp. 1537–1541). Pacific Grove, USA: IEEE.
30. Doppler, K., Rinne, M. P., Janis, P., et al. (2009). Device-to-device communications; functional prospects for LTE-advanced networks. In *IEEE international conference on communications workshops* (pp. 1–6). Dresden, Germany: IEEE.
31. Duan, L., Gao, L., & Huang, J. (2014). Cooperative spectrum sharing: A contract-based approach. *IEEE Transactions on Mobile Computing, 13,* 174–187.
32. Zhang, Y., Song, L., Saad, W., et al. (2015). Contract-based incentive mechanisms for device-to-device communications in cellular networks. *IEEE Journal on Selected Areas in Communications, 33,* 2144–2155.
33. Le, A., Keller, L., Seferoglu, H., et al. (2016). MicroCast: Cooperative video streaming using cellular and local connections. *IEEE/ACM Transactions on Networking, 24,* 2983–2999.
34. Eittenberger, P. M., Herbst, M., & Krieger, U. R. (2012). RapidStream: P2P streaming on android. In *2012 19th international packet video workshop* (pp. 125–130). Munich, Germany: IEEE.
35. Duong, T. Q., Vo, N.-S., Nguyen, T.-H., et al. (2015). Energy-aware rate and description allocation optimized video streaming for mobile D2D communications. In *2015 IEEE international conference on communications* (pp. 6791–6796). London, UK: IEEE.

36. Kim, J., Caire, G., & Molisch, A. F. (2016). Quality-aware streaming and scheduling for device-to-device video delivery. *IEEE/ACM Transactions on Networking, 24,* 2319–2331.
37. Zhu, H., Cao, Y., Wang, W., et al. (2015). QoE-aware resource allocation for adaptive device-to-device video streaming. *IEEE Network, 29,* 6–12.
38. Ghalut, T., Larijani, H., & Shahrabi, A. (2016). QoE-aware optimization of video stream downlink scheduling over LTE networks using RNNs and genetic algorithm. *Procedia Computer Science, 94,* 232–239.
39. Lee, G., Kim, H., Cho, Y., & Lee, S.-H. (2014). QoE-aware scheduling for sigmoid optimization in wireless networks. *IEEE Communications Letters, 18,* 1995–1998.
40. Kolding, T. (2006). QoS-aware proportional fair packet scheduling with required activity detection. In *64th IEEE vehicular technology conference* (pp. 1–5). Montreal, Canada: IEEE.
41. Fodor, G., Dahlman, E., Mildh, G., et al. (2012). Design aspects of network assisted device-to-device communications. *IEEE Communications Magazine, 50,* 170–177.
42. Gurobi Optimization. (2018). Gurobi optimization—The state-of-the-art mathematical programming solver. http://www.gurobi.com/index. Accessed 29 May 2018.
43. Wainer, G. A. (2009). *Discrete-event modeling and simulation: A practitioner's approach*. Boca Raton: CRC Press.
44. 3GPP. (2015). Evolved universal terrestrial radio access; RF system scenarios. Technical report TR36.942.
45. Al-Hourani, A., Chandrasekharan, S., & Kandeepan, S. (2014). Path loss study for millimeter wave device-to-device communications in urban environment. In *2014 IEEE international conference on communications workshops* (pp. 102–107). Sydney, Australia: IEEE.
46. Dahlman, E., Parkvall, S., & Sköld, J. (2014). *4G LTE/LTE-advanced for mobile broadband* (2nd ed.). Amsterdam: Elsevier.
47. Kilinc, C., Ericson, M., Rugeland, P., et al. (2017). 5G multi-RAT integration evaluations using a common PDCP layer. In *IEEE 85th vehicular technology conference* (pp. 1–5). Sydney, Australia.
48. 3GPP. (2016). Study on new radio access technology: Radio access architecture and interfaces. Technical report 38.801.
49. Cha, M., Kwak, H., Rodriguez, P., et al. (2007). I tube, you tube, everybody tubes. In *Proceedings of the 7th ACM SIGCOMM conference on internet measurement* (pp 1–14). New York, USA: ACM Press.
50. ITU-T. (2012). Infrastructure of audiovisual servicescoding of moving video. ITU-T Recommendation H.264.
51. Ahsan, S., Singh, V., & Ott, J. (2014). Characterizing internet video for large-scale active measurements. arXiv preprint arXiv:14085777v1.

Ala'a Al-Habashna received his Master of Engineering degree from Memorial University of Newfoundland in 2010, and his Ph.D. degree from Carleton University in 2018, both in Electrical and Computer Engineering. Currently, Dr. Al-Habashna is a Postdoctoral Fellow and an Instructor at Carleton University, Ottawa, Canada. He received the fellowship of the School of Graduate Studies at Memorial University of Newfoundland in 2010. Furthermore, Dr. Al-Habashna has won multiple awards including excellence and best-paper awards. He worked as a reviewer for many conferences and journals including the IEEE ICC and the Multimedia Tools and Applications journal. Dr. Al-Habashna is also the co-chair of the Communications and Networking Simulation (CNS) Symposium of the Spring Simulation multi-conference, and he is a Technical Program Committee Member in many conferences such as the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS). His current research interests include discrete-event modeling and simulations, signal detection and classification, cognitive radio systems, 5G wireless networks, communication networks architecture and protocols, IoT applications, machine learning, and multimedia communication over wireless networks.



Gabriel Wainer, FSCS, SMIEEE received the M.Sc. (1993) at the University of Buenos Aires, Argentina, and the Ph.D. (1998, with highest honors) at UBA/Université d'Aix-Marseille III, France. In July 2000, he joined the Department of Systems and Computer Engineering at Carleton University (Ottawa, ON, Canada), where he is now Full Professor and Associate Chair for Graduate Studies. He has held visiting positions at the University of Arizona; LSIS (CNRS), Université Paul Cézanne, University of Nice, INRIA Sophia-Antipolis, Université de Bordeaux (France); UCM, UPC (Spain), University of Buenos Aires, National University of Rosario

(Argentina) and others. He is one of the founders of the Symposium on Theory of Modeling and Simulation, SIMUTools and SimAUD. Prof. Wainer was Vice-President Conferences and Vice-President Publications, and is a member of the Board of Directors of the SCS. Prof. Wainer is the Special Issues Editor of SIMULATION, member of the Editorial Board of IEEE Computing in Science and Engineering, Journal of Defense Modeling and Simulation (SCS). He is the head of the Advanced Real-Time Simulation lab, located at Carleton University's Centre for advanced Simulation and Visualization (V-Sim). He has been the recipient of various awards, including the IBM Eclipse Innovation Award, SCS Leadership Award, and various Best Paper awards. He has been awarded Carleton University's Research Achievement Award (2005, 2014), the First Bernard P. Zeigler DEVS Modeling and Simulation Award, the SCS Outstanding Professional Award (2011), Carleton University's Mentorship Award (2013), the SCS Distinguished Professional Award (2013), and the SCS Distinguished Service Award (2015). He is a Fellow of SCS.