Towards QoE-assured 4K Video-on-Demand Delivery through Mobile Edge Virtualization with Adaptive Prefetching

Chang Ge, *Member, IEEE,* Ning Wang, *Member, IEEE,* Gerry Foster, Mick Wilson

***Abstract*—Internet video streaming applications have been demanding more bandwidth and higher video quality, especially with the advent of Virtual Reality (VR) and Augmented Reality (AR) applications. While adaptive streaming protocols like MPEG-DASH (Dynamic Adaptive Streaming over HTTP) allows video quality to be flexibly adapted, e.g., degraded when mobile network condition deteriorates, this is not an option if the application itself requires guaranteed 4K quality at all time. On the other hand, conventional end-to-end TCP has been struggling in supporting 4K video delivery across long-distance Internet paths containing both fixed and mobile network segments with heterogeneous characteristics. In this paper, we present a novel and practically-feasible system architecture named MVP (Mobile edge Virtualization with adaptive Prefetching), which enables content providers to embed their content intelligence as a virtual network function (VNF) into the mobile network operator's (MNO) infrastructure edge. Based on this architecture, we present a context-aware adaptive video prefetching scheme in order to achieve QoE-assured 4K video on demand (VoD) delivery across the global Internet. Through experiments based on a real LTE-A network infrastructure, we demonstrate that our proposed scheme is able to achieve QoE-assured 4K VoD streaming, especially when the video source is located remotely in the public Internet, in which case none of the state-of-the-art solutions is able to support such an objective at global Internet scale.**

***Index Terms*—MPEG-DASH, mobile edge computing, network function virtualization, prefetching, quality of experience, video on demand**

I. Introduction

In recent years, it has become the common vision that video streaming applications will dominate the Internet traffic in the near future, with Cisco Visual Networking Index forecasting that video applications will account for 75% of the overall Internet traffic by 2020 [1]. Meanwhile, with emerging advanced multimedia applications such as Virtual Reality (VR) and Augmented Reality (AR), it has become an essential requirement for future Internet architectures to support the streaming of 4K Ultra HD video content. In the context of the 5th Generation (5G) networks, one of its key objectives is to provide assured Quality of Experience (QoE) to end-users. In this context, the objectives of QoE-assurance are twofold. First, a seamless streaming experience should be provided with minimal interruption to playback. Second, the video quality should be guaranteed with a lower bound on video bitrate. In this paper, we focus on video on demand (VoD) applications, while live video applications are out of this paper's scope.

In recent years, video content providers such as YouTube, Netflix etc. have been adopting the MPEG-DASH (Dynamic Adaptive Streaming over HTTP) standard to provide streaming services [2]. While MPEG-DASH has many benefits such as offering flexibility through on-the-fly quality adaptation and its easy implementation over existing HTTP infrastructure, the fact that DASH uses TCP is a double-edged sword. On one hand, it means reliable content delivery and that video quality degradation caused by e.g., loss of I-frames can be avoided. On the other hand, when a wireless user equipment (UE) streams a video, there are 2 network segments on the end-to-end path that have distinctively different characteristics, which are 1) radio access network (RAN) that is wireless; and 2) the mobile core network and the public Internet that are typically wired. Specifically, the wired segment has high bandwidth-delay-product (BDP) due to the high-capacity over-provisioned backbone links and long latency due to the long data transport distance across the global Internet. In contrast, the wireless segment has much lower BDP due to limited radio resource capacity over the air interface and relatively lower latency. TCP does not perform well on end-to-end paths consisting of two segments with such different characteristics [3]. As we will show later in Section VI, even when there is no RAN resource competition, the Internet is unable to support seamless 4K video streaming in many scenarios.

The problem above can be circumvented by the involvement of content delivery networks (CDNs), where content can be cached at servers near the network edge to reduce content access latency. As we will show later in Section VI, if a video is already available at a CDN server, then its viewers are likely to experience desired QoE due to low access latency. However, when a content provider adopts the CDN approach, it needs to deploy all content on a per-domain basis by default, and cannot flexibly determine a selected subset of (e.g., popular) content items to be deployed at the CDN without incurring extra charges from the CDN operator [4]. Furthermore, the content provider will be unable to dynamically adjust its content manipulation policies based on real-time context of networks and/or users, because such policies are managed by the CDN operator instead. These may discourage content providers from using CDNs due to the lack of flexibility above.

We envisage an alternative solution to CDN that enables

Chang Ge, Ning Wang and Gerry Foster are with 5GIC, Institute for Communication Systems, University of Surrey, Guildford, United Kingdom (email: {C.Ge, N.Wang, G.Foster}@surrey.ac.uk).

Mick Wilson is with Fujitsu Laboratories of Europe, Hayes, United Kingdom (email: mick.wilson@uk.fujitsu.com).

Digital Object Identifier:

MNOs and content providers to directly collaborate through virtualized hardware resource at the mobile network edge, while achieving similar performance to CDN's. Under such a business model, the content provider pays MNOs to rent virtualized hardware resource and flexibly deploy its own content servicing capability and content manipulation policies (e.g., prefetching or adaptation). This enables the content provider to flexibly manage which content needs to be prefetched at the network edge and which content to stay at the content origin on a per-URL basis. Furthermore, since the virtual hardware resource is located within the MNO infrastructure, the content provider can make use of the real-time content information provided by the MNO and flexibly adjust its prefetching policies for better performance. This also allows the content provider to directly offer and manage different service classes to users with different QoE requirements. All these flexibilities above are much more complicated (if not infeasible) to achieve under the CDN model, and we will show in this paper why such flexibilities are important when assuring users' QoE.

In this paper, we introduce a novel video delivery scheme named Mobile edge Virtualization with adaptive Prefetching (MVP), which aims at providing QoE-assured 4K VoD streaming to mobile users at a global Internet scale. In MVP, the content providers deploy their content intelligence (e.g., caching and prefetching) as virtual network functions (VNFs) directly at the Mobile Network Operator's (MNO) infrastructure edge. Specifically, at the "MVP edge" where content intelligence is deployed as a VNF, the following functions are implemented. First, it realizes context awareness on network and users. For network context, it captures the RAN condition that is disseminated by the MNO through the Radio Network Information Service (RNIS) as specified by the ETSI Mobile Edge Computing (MEC) paradigm [5] [6] [7]. For user context, it regularly infers the user's QoE through techniques that are described in Section V. Second, it performs *adaptive* prefetching on a per-user per-session basis, i.e., it pre-downloads video segments from the video source and maintains a progress gap ahead of the user's actual request progress. Such a gap is adaptive and is optimized based on its real-time knowledge on network and user context on-the-fly. Third, it performs video quality adaptation on a per-segment basis also based on its context awareness, such as user mobility pattern etc. Note that such adaptation must not fall below the minimum bitrate required by specific video applications. For example, the most widely-adopted H.264/AVC normally requires 15Mbps for regular 4K video [8] and 30Mbps for 360-degree VR applications [9], while H.265/HEVC requires lower bitrates. In this paper, we use 15Mbps and 30Mbps as two representative video bitrate requirements.

It is worth noting that this is not the first work to propose video prefetching at the network edge. However, as we will discuss in Section II, existing prefetching techniques (at UE or network edge) all follow a rigid policy in terms of number of video segments to be prefetched. While they may work satisfactorily under limited scenarios (e.g., when streaming at 15Mbps, they can assure QoE if backhaul latency is less than 200ms), we will show in Section VI that such prefetching strategies are unable to assure user QoE when the content source in the public Internet is located beyond that latency. To make it more complicated, such latency requirements also depend on specific video applications (e.g., 15Mbps for standard 4K and 30Mbps for 360 VR videos). As such, a more intelligent prefetching mechanism is required.

The significance of this paper lies in its first successful attempt to enable a content provider to autonomously deploy its QoE-assured 4K VoD service that can reach mobile clients anywhere in the global Internet. Specific technical contributions of this paper can be summarize as follows:

- The MVP scheme is the first practically-feasible video delivery system that enables QoE-assured 4K DASH VoD streaming at the Internet scale to mobile UEs without relying on any CDN infrastructure as an intermediary. It also represents an alternative business model that achieves a win-win situation between directly collaborative MNOs and content providers, which is done through deploying content intelligence as VNFs at the mobile edge.
- The MVP scheme is the first prefetching scheme that is comprehensively validated and evaluated in a real LTE-A network infrastructure. Through extensive experiments under various realistic scenarios, we have demonstrated that the MVP scheme is *always* able to achieve QoE-assured 4K VoD streaming at 15Mbps and 30Mbps, which meet the requirements of standard 4K and 360 VR video applications respectively.
- We identify the applicability scenarios of different video delivery enhancement techniques that is required to support QoE-assured 4K VoD delivery to UEs in an LTE network. These are useful for the MVP edge to apply an appropriate level of technique without incurring unnecessary signaling and computing overhead.
- The majority of today's Internet video applications are encrypted using HTTPS, which require end-to-end encryption between end-users and the content source. The MVP scheme is the first to provide an answer on how to embed content intelligence into the mobile network edge while preserving such end-to-end encryption.

II. Background and Related Work

A. MPEG-DASH

MPEG-DASH is one of the over-the-top video delivery protocols that follow the principle of HTTP adaptive streaming. It was standardized in 2014 [2] and has been widely adopted by content providers such as YouTube and Netflix etc. Under DASH, a video is first compressed into multiple qualities or "representations", and each representation is divided into multiple segments (with identical time duration). The information on a video's encoding, compression and segments are contained in a manifest file called MPD (Media Presentation Description). When a DASH client streams a video, it is able to switch video quality between segments. The decisions on quality switching are made by the client based on its knowledge on its own perceived historic network performance (e.g., throughput) and/or its video buffer fill level.

B. 4K Streaming: Challenges

Many schemes have been proposed in the literature (see the next subsection) to improve video QoE. In order to minimize rebuffering duration, which is a key QoE metric, a common practice among those schemes is to adapt to lower video bitrates. Low-bitrate segments are quicker to retrieve due to their small size. Hence, even when the network resources are scarce, at least those low-bitrate segments can still be delivered to video users so that they have something to watch.

However, such a strategy cannot be adopted if the video application requires 4K quality. From a business perspective, if a user has paid a premium price to stream 4K video but the actual video quality keeps falling below 4K, business penalties may be incurred on the content provider and/or the network operator. Also, when a user streams a 360 video through a VR headset, poor video quality (below 4K) may cause the user to get physically sick, which is even more severe than QoE deterioration in standard 4K applications. Therefore, it is more challenging to assure QoE of video applications that require 4K quality (15Mbps and above), because the prefetching scheme cannot avoid rebuffering through serving low-bitrate segments to users. More sophisticated schemes are required to assure QoE of these applications.

C. Video QoE Improvement Techniques

There have been many research works on improving video QoE, which can be classified into the following categories.

1) Video quality adaptation: Many works proposed to dynamically adapt the streamed video quality under fluctuating network conditions, so that rebuffering can be best avoided. Representative works include [10]–[13] which uses historic per-segment download throughput to predict future throughput and choose the optimal video quality. More advanced works, such as [14], also takes into account the measured available end-to-end bandwidth. The scheme in [15] and [16] choose video quality based on user buffer and predicted probability of user experiencing rebuffering respectively. The scheme in [17] jointly considers video quality switching and user buffer. However, as discussed above, these techniques do not meet the requirements of video applications that require guaranteed 4K quality, as quality adaptation below 4K is not allowed.

2) Prefetching at UE: In [18], it is proposed that users using social networking may prefetch videos or video segments to their own devices through recommendations. In [19] and [20], the authors proposed to prefetch different segments of either one video or multiple videos in parallel through multi-path TCP. In [21], it is proposed to pre-download YouTube videos to UEs based on a recommender system, which predicts which video(s) are likely to be watched by users. In [22], the authors proposed to perform adaptive prefetching at the UE, where different prefetching strategies are calculated by considering fluctuating wireless channel conditions, memory constraints and application latency. Such UE-based prefetching do not take into account the unique challenge introduced by the end-to-end path that includes both RAN and public Internet, in which case UE-based prefetching will still experience suboptimal TCP performance. Therefore, our work focuses on prefetching at the network edge, rather than at the UE.

3) Caching at network edge: Works in this category propose to cache popular content at the network edge, which is envisaged for 5G systems in [23]. It is proposed that content cache can be placed at multiple levels within a 4G/5G network architecture. Authors in [24] proposed to pre-cache video segments at the network through users announcing which videos will be watched by them in the near future. In [25], the authors proposed an edge caching system where the MEC server considers both RAN condition information and the Channel Quality Index (CQI) information reported by the UE. These information are used to make decisions on video quality adaptation and caching. In [26], the authors proposed an MEC caching scheme which considers 1) per-segment, per-quality popularity of videos; and 2) RAN condition when making caching decisions. However, as discussed in Section I, while caching works especially well for popular videos, not all videos can be cached at the network edge.

4) Prefetching at network edge: This is the category this work falls into. In an early work [27], the authors developed a generic model that makes 2 decisions: 1) at a network edge proxy, when is prefetching needed; and 2) how many segments need to be prefetched. However, it did not take into account the effect of network (especially latency) on TCP performance and hence, the download duration of each segment. In [28], a network element is deployed to monitor traffic characteristics and send recommendations on video quality to UEs. However, no caching or prefetching is involved in this work. In [29], it is proposed to deploy prefetching proxies at wireless access points, which are capable of monitoring traffic conditions of both backbone network and the wireless channel. These information are used to make decision on which quality to prefetch for the next segment. However, it only prefetches 1 segment at a time, which may not be enough to ensure seamless playback as we will show in Section IV. In [30], a proxy-based prefetching scheme is proposed. However, its prefetching policies are also fixed (i.e., either 1 segment or a fixed n segments ahead). In the following sections, we will show that such kind of rigid prefetching policy does not perform well in a RAN environment with fluctuating network conditions. In contrast, our proposed scheme is capable of dynamically adapting its prefetching policy with respect to user and network context in real-time, which is important in assuring video users' QoE.

III. MVP Architecture Overview

In the network infrastructure that is owned by the MNO, some IT resources (computing and storage) that are located at the network edge (e.g., close to eNodeBs) can be virtualized and leased by the MNO to third-parties such as content providers. We call such virtualized content platform the *MVP edge*. The content providers can use the virtualized storage and computing resources at the MVP edge to deploy their content and intelligence (such as caching and prefetching policies).

A high-level overview of the MVP system architecture is presented in Figure 1. From the UE's perspective, the MVP edge is responsible for handling all of its video segment requests during VoD sessions. If a requested video segment is

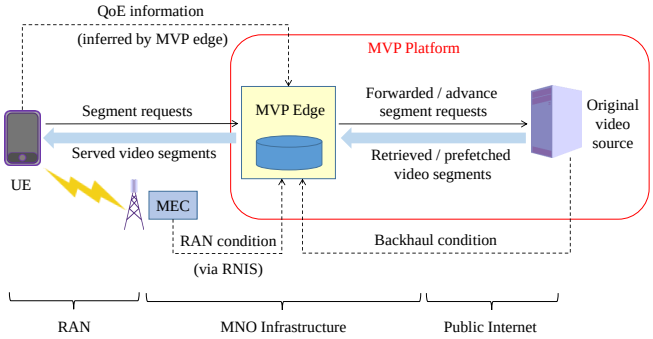


Fig. 1: MVP System Architecture

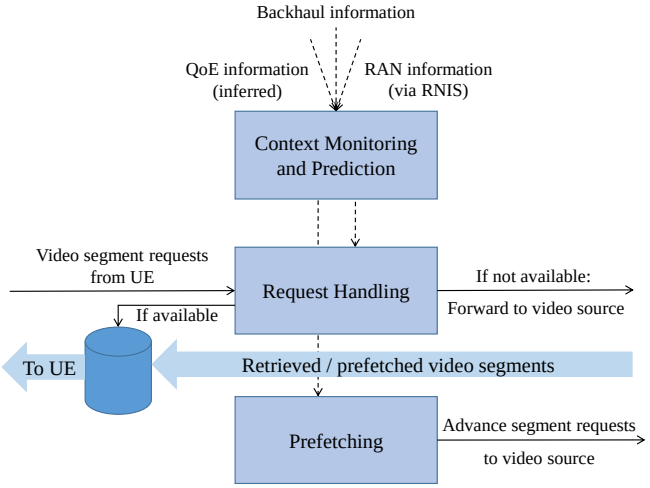


Fig. 2: MVP Edge Functional Block Diagram

available at the MVP edge, it is served to the UE immediately with low access latency since it is located at the network edge. If it is not available, the MVP edge forwards the request to the original video source, retrieves the requested segment and serves it to the UE. Furthermore, the MVP edge's embedded content intelligence also performs prefetching by downloading video segments in advance, i.e., ahead of the UE's request progress. Note that in the MVP platform that covers all the way between the MVP edge and the video source, both entities are owned by the content provider. Therefore, the content retrieval and prefetching operations described above can be applied to encrypted content. In other words, the MNO does not participate in any content operation above.

In Figure 2, we describe the functionalities of the MVP edge in more detail. Specifically, besides the request handling function described earlier, the MVP edge has the functionalities of 1) context monitoring and prediction, 2) adaptive prefetching and 3) video quality adaptation. In terms of context awareness, the MVP edge needs to have real-time knowledge of UEs and networks, which will be used to drive its decisions on prefetching and video quality adaptation.

First, the MVP edge is able to infer each UE and its sessions' QoE (e.g., its video buffer status) and its historic per-segment download throughput without any feedback signaling from the UE (more details in Section V). The MVP edge is also aware of real-time RAN context information, which is periodically disseminated through the MNO-owned MEC server's RNIS module [5]. Specifically, through interaction with low-level API with UEs and eNBs, the RNIS module is able to provide real-time knowledge on RAN load (e.g., the number of active streaming sessions), each UE's allocated RAN bandwidth, RAN congestion, etc [6] [7]. With such context knowledge, the MVP edge is able to optimize the prefetching policy for each VoD session. Furthermore, it is able to assess each sessions' risk on suffering from playback freezing, and hence prioritize high-risk sessions so that a unified QoE may be offered to all sessions.

Second, the MVP edge is aware of the network condition between itself and the video source(s) in terms of bandwidth, latency and packet loss. Such knowledge is used by the MVP edge to estimate how long it takes to download / prefetch video segments from the video source, which allows it to optimally schedule the prefetching operations. Note that the MVP edge does not need to continuously monitor such knowledge. Instead, it only needs to learn it at the beginning of each VoD session, since the download throughput between the MVP edge and the video source are relatively static. This is not only because backhaul links are typically over-provisioned, but also that the MVP edge maintains and reuses its TCP connections with the content source for prefetching, and such long-lived TCP connections' performances are not affected significantly by other background traffic in the WAN [31]. Specifically, at the beginning of each session, the MVP edge is able to resolve the UE requests to a video source. Knowing the video source's IP address, the MVP edge is then able to estimate the network condition between itself and the video source by performing ping-like measurements. Note that since the MVP edge and the video source both belong to the content provider, such internal measurement is feasible even if they are connected in an encrypted way (e.g., through VPN). On the other hand, it is much more complicated for conventional network proxy-based schemes to perform such measurements, since the proxy and the video source are owned by different stakeholders.

Third, the context information above (especially the RAN condition) also enables the MVP edge to make decisions on video quality adaptation while handling UE requests. For example, if a UE's downlink throughput has been much higher than 15Mbps and the RAN has light load, then even if the UE has requested a segment in 15Mbps quality, the MVP edge may serve the same video segment in higher quality (e.g., 30Mbps) to the UE without risking deteriorated QoE. Note that if the video application requires 4K quality, then the MVP edge will perform adaptation within the range of 15Mbps and above. If the application is less demanding in video quality, the MVP edge can also downgrade below 15Mbps. In this paper, we only consider applications that require 4K quality.

In the next two sections, we present two key enabling techniques for the MVP scheme, including 1) QoE-aware adaptive prefetching and 2) accurate real-time inference of user QoE during VoD sessions, which provides inputs to the former

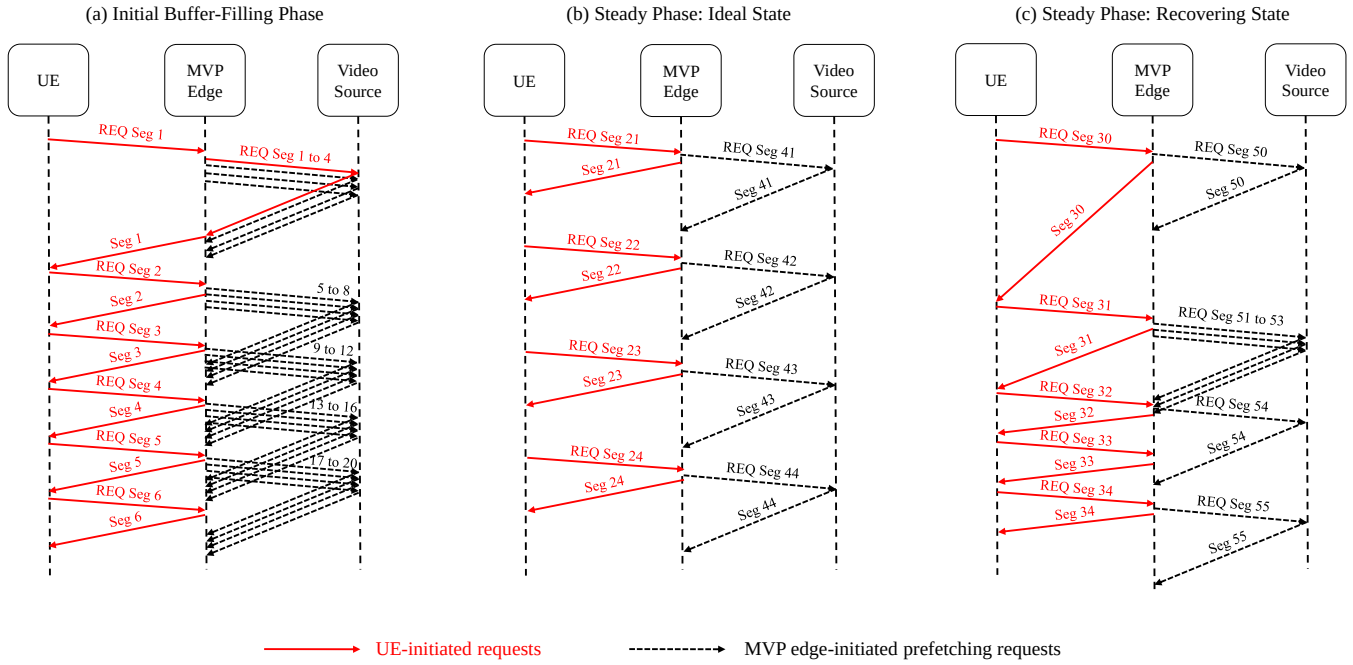


Fig. 3: Illustration of QoE-aware adaptive prefetching policy under different scenarios

without any signaling feedback from UE to the MVP edge.

IV. QoE-Aware Adaptive Prefetching

We have discussed the importance of content localization in Section I, especially in terms of the downlink throughput performance gain due to reduced end-to-end latency. In this section, we present a novel prefetching scheme that is able to potentially support QoE-assured delivery of 4K videos whose original sources are located anywhere in the global Internet. Such an adaptive prefetching technique guarantees video segment availability at the MVP edge before they are requested by UEs. In other words, prefetching will not become the bottleneck of the end-to-end video delivery, even when the data path is cross-continent across the Internet. Furthermore, for prefetched video segments that are popular, they can be kept at the MVP edge's cache for consumption by other users, and such cache can be flexibly managed by the content provider's own policies (unlike in CDNs where cache is managed by CDN operators). Such a feature is out of this paper's scope.

Typically, there are two key decisions that need to be made during a prefetching operation.

First, how many segments should be prefetched ahead of the UE's request progress during a session? On one hand, prefetching too few segments ahead means a higher risk of rebuffering during a session. On the other hand, prefetching too many segments not only causes unnecessary bandwidth usage in the backhaul network, but also occupies unnecessarily more storage space at the MVP edge. Such tradeoff needs to be carefully balanced. In practice, the MVP edge is pre-configured with an upper limit on the number of prefetched segments to avoid excessive prefetching.

Second, which video quality should be prefetched for each segment to be streamed? Since we consider applications that require 4K quality, the video quality must be adapted within the range of 15Mbps and above. Normally, such a decision takes into account predicted downlink throughput etc. There are many adaptation schemes that have been proposed in the literature (such as [32]), which can be directly embedded into our prefetching scheme. Therefore, video quality switching algorithms are outside this paper's scope.

When deciding how many segments to prefetch ahead of a UE's request progress, the related work in the literature follow a relatively rigid approach [30], which prefetches a fixed number of segments ahead of the UE's request progress. Earlier work [27] investigated generic segment prefetching techniques at network proxy, but without specifically taking into account DASH characteristics and the performance impact due to TCP behaviors. Now, we elaborate on the technical issues pertaining to the DASH-based prefetching operation, which requires specific context awareness.

First, when there is long distance between the MNO network infrastructure and the video source in the public Internet, the TCP throughput when prefetching a single segment can be very low, especially when slow-start is taken into account due to video segments' small sizes (generally $<$10MB). Based on our testbed measurement results, the average throughput when prefetching a single 8MB segment over an 1Gbps link with 300ms latency and 0.05% packet loss (typical characteristics of an Europe-Asia content delivery path [33]) is less than 5Mbps (i.e., 0.5% of total physical bandwidth). Such a data rate is not able to support seamless delivery of 4K video segments, whose minimal average bitrate is 15Mbps. In other words, prefetching 1 segment at a time will not be able to match the playback progress and will still cause rebuffering at the UE.

In fact, the MVP edge needs to open at least 3-4 concurrent TCP connections with the video source and prefetch at least 3-4 segments in parallel to try to avoid rebuffering. In general, this kind of decisions need to be made with knowledge of network context, which involve bandwidth, latency and packet loss that are associated with the path from video sources.

Second, in order to allow buffer filling at the UE, the MVP edge's prefetching progress and frequency must be at least as fast as, but in general faster than the UE's request progress. This is because most DASH clients adopt a 2-phase approach during playback. In phase 1 - **the initial buffer-filling phase**, a UE requests segments consecutively to fill up its buffer as quickly as possible. Note that "consecutively" means segment $n+1$ is requested *immediately* after segment n is *delivered* to the UE - it does not mean multiple segments are requested concurrently. Afterwards, if the buffer reaches a preset threshold (e.g., 30s), it enters phase 2 - **the steady phase**. If the UE's downlink throughput is sufficiently high, it stays in an **ideal state** where it requests each new segment every segment length (which is the same for all segments based on DASH specification, e.g., 2s) to maintain its buffer level. If the UE's downlink throughput decreases and causes its buffer to drop below the threshold, it will enter a **recovering state** in which it requests segments consecutively in an effort to refill its buffer. Therefore, it can be inferred that the MVP prefetching policy should be adaptive to match different playback phases and states.

We illustrate our adaptive prefetching policies under 3 different scenarios in Figure 3. Figure 3(a) shows the initial buffer-filling phase. It is seen that as the UE sends requests consecutively, in order for the MVP edge to stay ahead of the UE's progress, it prefetches multiple (e.g., 4 as an illustration) segments at a time. It can be observed from the segment numbers in the figure that the MVP edge gains its advance over the UE in terms of locally stored segments. Note that the actual number of prefetched segments during this stage should be specific to the network context between the MVP edge and the video source, i.e., the backhaul throughput. For example, if the throughput of a single TCP connection over the backhaul network is lower than 15Mbps due to long distance, more video segments need to be prefetched in parallel over concurrent TCP connections, so that the overall aggregated backhaul throughput is high enough for the MVP edge to maintain its advance over the UEs progress, and vice versa.

Note that as each segment needs to be prefetched over a unique TCP connection to maximize overall backhaul throughput, it causes requests to arrive at the video source in a more spiking manner. Considering the typical scenario where multiple concurrent video sessions all require prefetching, this poses a potential risk of overloading the video source, because the number of parallel TCP connections a server can handle concurrently is limited. Therefore, if any session's initial buffer-filling phase requires a large number of video segments being prefetched in parallel, the MVP edge needs to "pace" the prefetching operation for that session. For example, if 20 video segments need to be prefetched, instead of opening 20 parallel TCP connections and downloading them all at the same time, the MVP edge may open 4 TCP connections at a time and downloads a segment from each, and completes the prefetching operation in 5 batches. Such pacing policy is subject to adjustment depending on how aggressive the MVP edge wants to prefetch for a session. This effectively reduces the risk of the video source being overloaded by a potentially large number of concurrent prefetching operations.

Figure 3(b) shows the ideal state in the steady phase. Assuming the UE has smoothly gone through the initial buffer-filling phase, the MVP edge would have gained a healthy advance over the UE's progress. In the ideal state, the UE's downlink throughput is high enough to maintain its buffer level. Therefore, as the UE issues a new request every segment length, the MVP edge also needs to prefetch only the next segment every segment length to maintain its advance.

Figure 3(c) shows the recovering state in the steady phase. There are many factors that may cause the UE to enter the recovering state. For example, when an eNB is operating close to its capacity (which we will demonstrate in Section VI), the radio resource competition among UEs may cause some of them to experience low throughput temporarily. Another example is user mobility, which causes a UE's throughput to fluctuate as it travels along different eNBs' radio coverage. In the figure, it is seen that it took substantially more time for the UE to download segment 30 than usual. When the UE receives segment 30, its buffer has already dropped below the threshold, which means it enters the recovering state and requests segment 31 immediately. Since the MVP edge is aware of the UE's dynamic buffer situation (more details will be described in the next subsection), it prefetches more segments to compensate the UE's reduced buffer. This is because it is predicted that the RAN will recover for the UE soon, and if it does not prefetch more segments, it will not be able to meet the UE's consecutive requests as well as maintaining its advance over the UE's progress. It is shown in the figure that thanks to such a prefetching policy, the UE is able to recover its buffer quickly as soon as the RAN condition improved. In the meantime, the MVP edge maintained its 20-segment advance over the UE's progress.

The illustrations in the 3 scenarios above have demonstrated that in a very dynamic environment like RAN, it is very important for the MVP edge to be aware of the context information of both users (e.g., video buffer) and the networks (including both RAN and public Internet). They have also demonstrated that under such an environment, there is no one-fit-all prefetching policy - it should be updated on-the-fly to match the changing situation as a session goes on.

V. QoE Inference Mechanism

In this section, we discuss our QoE inference mechanism at the MVP edge, which provides important inputs to the adaptive prefetching scheme. By "inference", we mean that such a mechanism does not require any explicit feedback from UEs. Since none of the existing DASH clients has implemented buffer reporting mechanism yet, we adopt the approach where the MVP edge *infers* the UE's buffer length through packet sniffing. Note that such an approach would work for both plain-HTTP and HTTPS-encrypted video traffic.

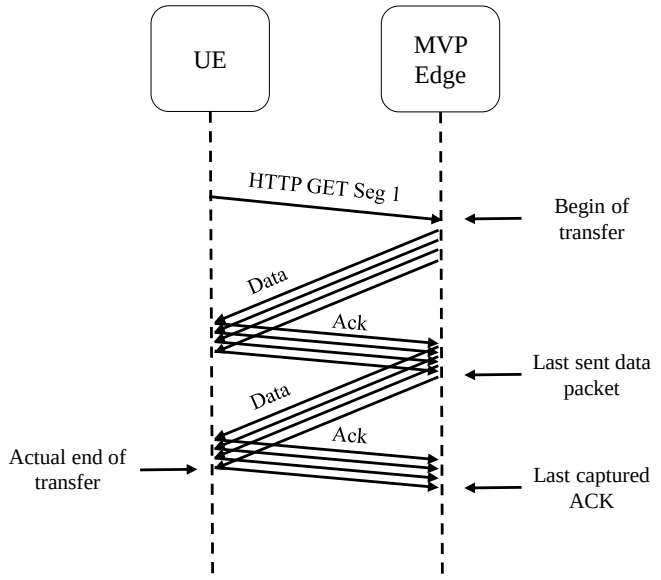


Fig. 4: Illustration of buffer inference mechanism

This is because the MVP edge is owned by the content provider, which means the MVP edge is a termination point of the Secure Socket Layer (SSL) connection and it can decrypt video traffic and perform packet sniffing without breaking the end-to-end security between the end-user and the content provider.

We take the buffer inference mechanism in [34] as a starting point and enhanced it to make it more robust and accurate for operating in an LTE environment with very dynamic network conditions. In a nutshell, packet sniffing is performed at the MVP edge to detect the beginning and end of each video segment transfer to the UE, and such detection needs to be done in real-time to calculate buffer length accurately.

We illustrate in Figure 4 the TCP-level sequence diagram of a typical video segment download process. The process starts when the MVP edge receives an HTTP GET request from the UE. The MVP edge then starts transferring the file in chunks, where each chunk consists of multiple TCP segments. The MVP edge waits until all TCP segments of a sent chunk are acknowledged from the UE before it sends the next chunk. There are 4 key timestamps in such a process that need to be detected in real-time:

- **Begin of transfer**: this is detected when the MVP edge receives the HTTP GET request.
- **Actual end of transfer**: this is when the UE actually receives all data of the transfer, which is *before* it acknowledges them to the MVP edge.
- **Last sent data packet**: this is when the MVP edge sends out the last TCP segment that contains the requested video data.
- **Last captured ACK**: this is when the MVP edge receives the last ACK sent from the UE.

Intuitively, the "actual end of transfer" timestamp is the most accurate one to decide when to add buffer to the UE. However, it cannot be captured at the MVP edge, because it can only be determined through packet capturing at the UE. The mechanism in [34] proposed to use the "last sent data packet" as an alternative. While this was shown to work in [34] (which is based on a fixed network setup), it does not always work in an LTE RAN, because the latency (round-trip time, RTT) over the RAN air interface is non-negligible (20 to 50ms) and highly fluctuating (up to 300ms) by the nature of LTE air interface). This means the difference between the "last sent data packet" timestamp and the "actual end of transfer" timestamp can be as high as 150ms *every time*. While some may argue that this is not a big error considering the segment length which is in the order of seconds, such an error is additive at every segment download and can easily lead to an error of multiple seconds. It is worth noting that although 5G advertises super-low latency compared to LTE (e.g., 8ms under eMBB scenario [35]), when the air interface is imperfect due to e.g., mobility and shadowing etc., its latency is still subject to increasing and fluctuation. Therefore, the challenges above is expected to be present in 5G networks too.

Therefore, we propose to use the mid-point between the "last sent data packet" and the "last captured ACK" to approximate the "actual end of transfer" timestamp. Such a strategy assumes the last data and ACK packets experienced similar latencies, which is reasonable since they are sent within a very short time period. More specifically, the "last sent data packet" timestamp is detected when the total number of bytes that the MVP edge has sent reaches or exceeds the requested video segment's size. The "last captured ACK" timestamp is detected when the ACK packet's ACK number matches the last sent data packet's SEQ number.

With these timestamps accurately detected, the MVP edge is able to calculate the UE's buffer length periodically (e.g., every 0.5s), and hence is able to calculate both the **initial playout delay** and any **rebuffering event**'s occurrence and duration on the UE. As for the **streamed video quality**, it can either be directly extracted from the requested video segment's URL (e.g., the BBC iPlayer application[1]), or we assume that the content provider who owns the MVP edge knows how to translate the segment URL into its quality information. Hence, all 3 major QoE metrics are inferred at the MVP edge. We have validated the accuracy of the QoE inference mechanism through experiments, which confirm that it is able to achieve 100% accuracy in inferring all above QoE metrics.

VI. Performance Evaluation

A. Experiment Setup

The network topology that is used in our experiments is shown in Figure 5, in which the LTE-A C-RAN network testbed infrastructure (3GPP Rel. 14) is hosted by 5G Innovation Center (5GIC) at University of Surrey, UK. We use 5 Huawei Nexus 6P mobile phones (running Android 7.1.1) to conduct the experiments. The UEs are connected to an indoor lampsite, which operates at LTE TDD Band 41 (2545-2575MHz) and offers a maximum downlink throughput of approximately 112Mbps [36]. Note that this is the theoretical

[1]http://www.bbc.co.uk/iplayer

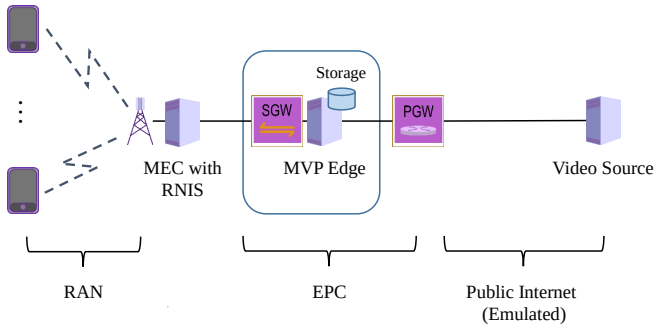


Fig. 5: Testbed network topology

downlink throughput experienced by a *single* UE. When multiple UEs are streaming videos, their aggregated throughput is less than 112Mbps due to radio resource management overhead at the base station.

A MEC server has been implemented and deployed at an aggregation point that is close to the eNBs, and its RNIS module has also been implemented to disseminate knowledge of RAN condition. In our implementation, the RNIS can collect context information on a per-UE basis, which includes each UE's signal strength, downlink throughput, mobility pattern (static/low/high), as well as GPS coordinates (if the user agrees to share them). The RNIS also provides knowledge on each eNB's cell load, especially on the number of UEs who are actively streaming video. These information are periodically updated and stored in an MySQL repository so that the MVP edge can easily access them.

In the evolved packet core (EPC), a virtualized node with the MVP edge implemented and integrated as a VNF is deployed at the S-GW. The MVP edge runs a custom implementation of Jetty[2] web server in Java, which realizes the request handling and prefetching mechanisms described in Section IV. It takes as input from the MEC server's RNIS database to make decisions on prefetching.

To the north of the EPC, we have deployed a remote HTTP server that serves as the video source in the public Internet. It handles requests forwarded from the MVP edge if a requested content is not available locally there. In order to emulate different public Internet conditions between the P-GW and the video source, we adjust network latency and packet loss on the link between them using Linux's built-in netem toolbox. According to Sprint's monitored network performance in 2016 [33], we set the network parameters *between P-GW and the video source* to be the following scenarios:

- Link capacity: 1Gbps (adequate to represent over-provisioned backhaul link compared to the video traffic load in the experiments). Note that we have performed experiments where the backhaul link load varies between 0% and 60%, and the MVP edge achieved similar performance because it uses long-lived TCP connections that are not significantly affected by backhaul traffic [31].
- Latency (round-trip time) and packet loss in the public Internet:
  - 500ms and 0.05%: unconventional cases where the latency is excessively high due to abnormal events such as BGP rerouting in the Internet
  - 300ms and 0.05%: the video source is at a long-range location (e.g., Europe to East Asia)
  - 200ms and 0.04%: the video source is at a mid-range location (e.g., US to East Asia)
  - 100ms and 0.03%: the video source is at a short-range location (e.g., Europe to US)
  - 10ms and 0.01%: the video source is a completely local server (e.g., within UK)

[2]http://eclipse.org/jetty/

TABLE I: Videos used in Experiments

Video	Length	Frame Rate	Video Bitrate (Mbps)	
			4K	360 VR
1	6m47s	30	14.64	29.28
2	5m18s	30	14.53	29.1
3	4m06s	24	14.03	28.9
4	3m47s	30	14.74	29.6
5	2m57s	24	14.07	27.57

These scenarios will be referred to as 500ms, 300ms, 200ms, 100ms and 10ms scenarios respectively in the following text.

Regarding DASH streaming, all UEs use dash.js v2.5.0 as the client, which is a JavaScript-based DASH client software that is implemented as a reference client by DASH Industry Forum[3]. Since we use 5 UEs in the experiments, we prepared 5 different videos by compressing each of them to around 15Mbps and 30Mbps qualities using H.264, which correspond to the minimum quality requirements of standard 4K and 360 VR applications respectively. Each video is then divided into 2s segments during the "DASHify" process, which is a typical segment length for VoD [37]. Their details are listed in Table I. Note that since we focus on 4K and 360 VR applications, video quality adaptation below 15Mbps and 30Mbps are disabled respectively in the MPD manifest.

B. Reference Schemes

We evaluate the performance of the following three video delivery schemes:

- **End-to-end (E2E)**: the conventional video delivery scheme, where the UE streams video directly from the video source. No content storage or intelligence is deployed in the MNO infrastructure in this scheme.
- **4-ahead**: a representative state-of-the-art prefetching scheme, where the prefetching entity always stays 4 segments in advance of the UE's progress [30][4]. This scheme is implemented at the MVP edge without the adaptive prefetching scheme. Also, it does not take into account the context information disseminated through the MEC server's RNIS.

[3]https://github.com/Dash-Industry-Forum/dash.js/

[4]This work recommends prefetching either 1 or 4 segments in advance, and we implement the latter due to its better performance.

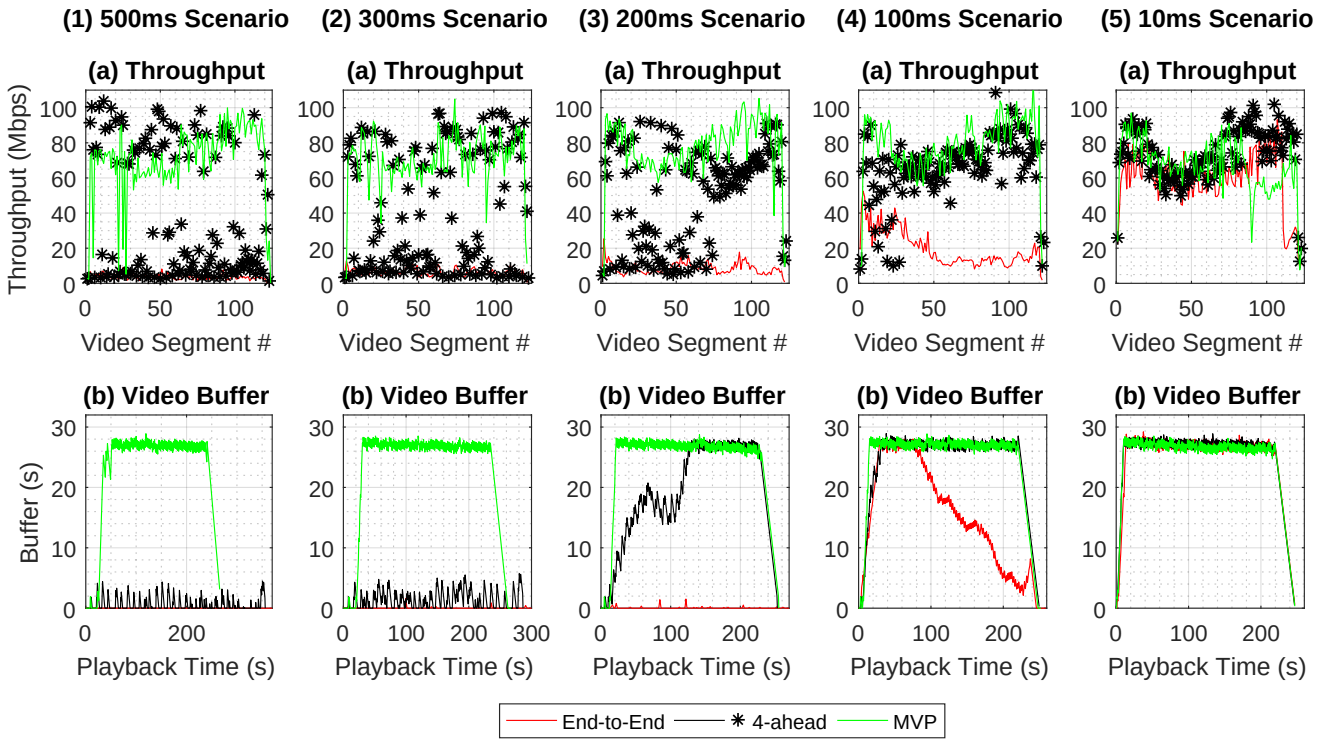


Fig. 6: Throughput and video buffer performance: single-UE, 15Mbps video

TABLE II: Performance statistics: single-UE, 15Mbps video

		500ms latency, 0.05% loss			300ms latency, 0.05% loss			200ms latency, 0.04% loss			100ms latency, 0.03% loss		
		E2E	**4-ahead**	**MVP**	**E2E**	**4-ahead**	**MVP**	**E2E**	**4-ahead**	**MVP**	**E2E**	**4-ahead**	**MVP**
Initial Delay		30.21s	10.96s	9.73s	8.26s	8.47s	9.25s	6.03s	6.49s	5.0s	2.94s	3.65s	3.2s
Rebuffering Duration		794.5s	100.5s	11.34s	348.4s	34.27s	11.9s	229.5s	4.34s	2.6s	0.9s	0.98s	0.39s
Prefetched Segments (%)		N/A	37.1%	95.2%	N/A	40.3%	98.4%	N/A	62.1%	99.2%	N/A	88.7%	99.2%
Downlink Throughput (Mbps)	Mean	3.63	39.31	70.68	6.25	42.45	70.18	8.52	50.9	76.78	19.7	65.9	78.95
	Max	9.03	103.76	99.9	12.81	97.21	104.9	25.43	92.1	105.3	52.54	108.58	114.6
	Min	0.28	1.36	0.14	0.76	3.05	2.72	0.7	4.47	4.0	1.8	8.14	9.5

- **MVP**: our proposed adaptive prefetching scheme, which is also deployed at the MVP edge. It implements all functionalities described in Sections IV and V.

C. Performance Metrics

We evaluate the following performance metrics:

- QoS metrics
  - **Downlink throughput**: this refers to each UE's download throughput of each video segment (in Mbps).
- QoE metrics
  - **Video buffer length**: the video buffer that is available on a UE (in seconds).
  - **Initial playout delay**: the time duration between when a user clicks "play" and when the video starts playing.
  - **Rebuffering duration**: the total time duration that a user spends in playback freezing.
  - **Prefetched segments**: the % of segments that have been prefetched to the MVP edge before they are requested by a UE.

D. Single UE, 15Mbps Video

In the following 2 subsections, we present results of experiments that are conducted using a single UE to stream a video (video 3 in Table I) at 15Mbps and 30Mbps qualities respectively. The main reasons for using a single UE are 1) to evaluate performance when the radio resource is always sufficient (i.e., not the bottleneck) for 4K video since the RAN capacity is significantly higher than the video's bitrate; and 2) to verify that even in such an ideal network environment, the conventional E2E scheme and the 4-ahead prefetching scheme are not *always* able to assure 4K VoD users' QoE, while our proposed MVP scheme can.

The performance results of the UE streaming video at 15Mbps are presented in Figure 6 with summarized statistics in Table II.

We first look at the results under the 500ms and 300ms scenarios. It can be easily observed from Figures 6 (1) and (2) that the MVP scheme significantly outperforms both E2E and 4-ahead schemes. Specifically, when compared with the

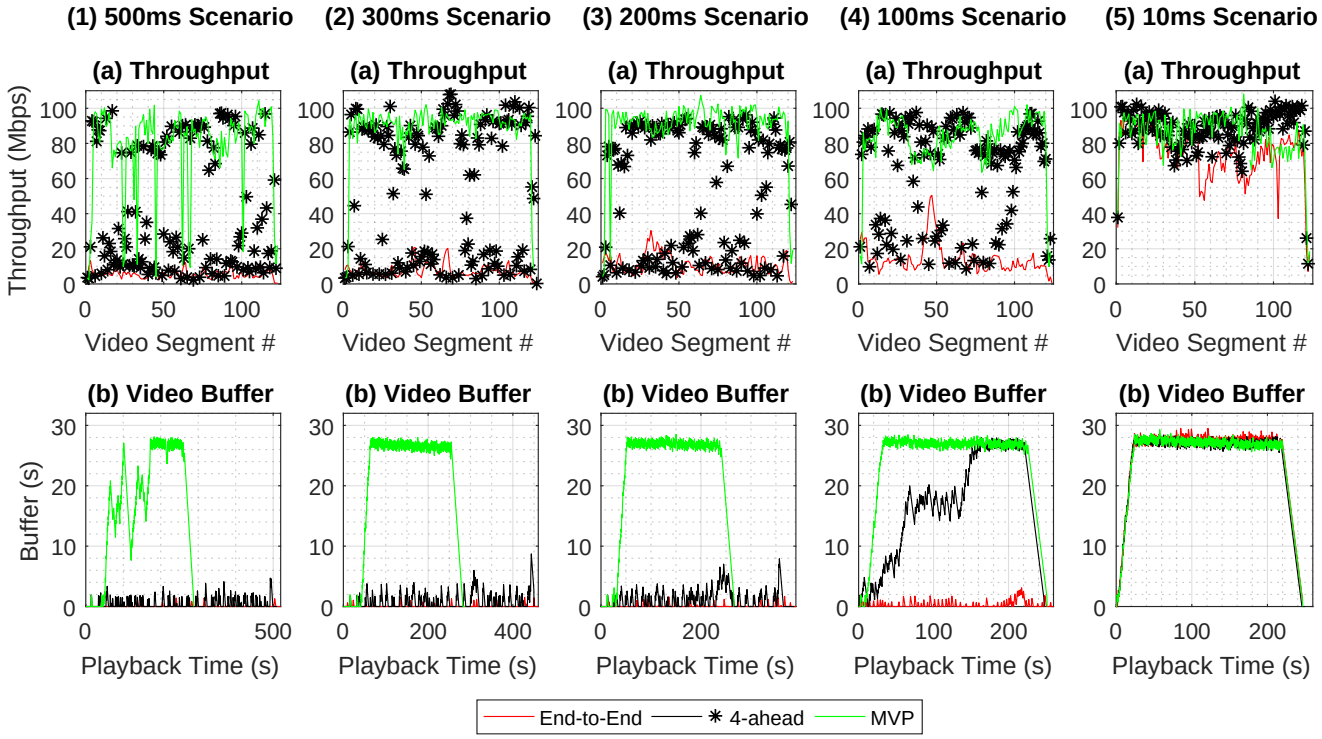


Fig. 7: Throughput and video buffer performance: single-UE, 30Mbps video

TABLE III: Performance statistics: single-UE, 30Mbps video

		500ms latency, 0.05% loss			300ms latency, 0.05% loss			200ms latency, 0.04% loss			100ms latency, 0.03% loss		
		E2E	**4-ahead**	**MVP**	**E2E**	**4-ahead**	**MVP**	**E2E**	**4-ahead**	**MVP**	**E2E**	**4-ahead**	**MVP**
Initial Delay		22.57s	16.1s	15.53s	11.21s	16.78s	16.29s	7.54s	13.7s	13.39s	3.6s	3.68s	2.8s
Rebuffering Duration		1136s	251.7s	26.29s	838.7s	188.1s	24.43s	492.4s	128.7s	6.9s	343.0s	2.2s	0.11s
Prefetched Segments (%)		N/A	37.9%	89.5%	N/A	54%	97.6%	N/A	55.6%	97.6%	N/A	67.7%	98.4%
Downlink Throughput (Mbps)	Mean	5.49	42.64	77.98	7.33	55.1	88.95	11.31	55.4	89.64	14.35	65.63	82.86
	Max	13.32	110.57	104.3	20.82	108	102	30.4	96.22	107.3	50.41	97.6	102
	Min	0.28	2.25	2.21	0.41	0.23	3.46	0.44	4.12	4.21	0.51	8.6	9.94

4-ahead scheme, the MVP scheme dramatically reduced its rebuffering duration from 100.5s to 11.34s (500ms scenario) and from 34.27s to 11.9s (300ms scenario), while increasing the average downlink throughput by 79.8% and 65.3% under the 2 scenarios respectively. These prove that the MVP scheme can realize seamless 4K video delivery at 15Mbps under such extremely challenging scenarios.

The E2E scheme is unable to seamlessly deliver 4K video at 15Mbps because of its low average throughput (3.63Mbps and 6.25Mbps, which are far lower than 15Mbps), which is mainly caused by TCP's congestion control mechanism under high latency and packet loss. On the other hand, the MVP scheme's performance improvement over the 4-ahead scheme is due to the MVP scheme's adaptive prefetching strategy, which is able to prefetch 95.2% and 98.4% of all video segments under the 500ms and 300ms scenarios. In contrast, the 4-ahead scheme is able to prefetch only 37.1% and 40.3% of all segments under the same 2 scenarios, because the long latency at the backhaul link means prefetching 4 segments in advance is not enough to match the UE's playback progress. Under the MVP scheme, the MVP edge starts prefetching the subsequent 20 segments (i.e., segments 3 to 22) at the same time as it begins retrieving segment 2. Therefore, although the download time of the first few segments are still long (which explains the ˜11s rebuffering duration under MVP), the 20 subsequent segments are prefetched to the MVP edge at a very early stage. This means the UE can download them "locally" with high throughput and fill up its buffer quickly (as shown in Figure 6 (1b) and (2b)), thanks to the much lower content access latency. From then on, the MVP edge only needs to prefetch 1 or 2 segments every time a new request arrives (i.e., every 2s) to maintain its advance over the UE's playback progress. Note that the MVP edge has a configurable upper limit for its prefetching progress' advance over UE's progress to avoid excessive prefetching, and we found in our experiments that 40s (i.e., 20 segments with 2s length) is adequate for such a purpose. This means if the UE quits video streaming before it is ended, up to 40s of prefetched video segments (which

TABLE IV: Applicability conditions of techniques required to assure QoE for VoD users (2s segments)

Latency / Packet Loss	**15Mbps Video (Standard 4K)**			**30Mbps Video (360 VR)**		
	E2E	**4-ahead**	**MVP**	**E2E**	**4-ahead**	**MVP**
500ms / 0.05% and worse	Unable	Unable	**Required**	Unable	Unable	At risk
300ms / 0.05%	Unable	Unable	**Required**	Unable	Unable	**Required**
200ms / 0.04%	Unable	At risk	**Required**	Unable	Unable	**Required**
100ms / 0.03%	Unable	**Required**	Overkill	Unable	At risk	**Required**
10ms / 0.01% and better	**Adequate**	Overkill	Overkill	**Adequate**	Overkill	Overkill

are about 75MB in total) would be wasted due to not being watched by the UE.

We next examine the results under the 200ms scenario. It is observed from Figures 6 (3) that the MVP scheme still outperforms both benchmark schemes, but the gap between their performance is closer. Specifically, both 4-ahead and MVP schemes achieved very low rebuffering duration (4.34s and 2.6s respectively). However, it is also observed that the 4-ahead scheme is able to prefetch only 62.1% of all segments in advance, which is reflected by its fluctuating throughput in Figure 6 (3a) and the UE's fluctuating buffer in Figure 6 (3b) during the first 120s. The 4-ahead scheme eventually managed to catch up with the UE's progress, and the UE's buffer filled up quickly afterwards thanks to the abundant RAN capacity as well as the relatively low video bitrate (15Mbps). However, we will show later that when the video bitrate is higher (i.e., 30Mbps), failing to prefetch such a large number of segments will cause significant QoE degradation. In contrast, the MVP scheme managed to prefetch 99.2% of all segments, and no fluctuation in throughput and video buffer are observed.

Under the 100ms scenario, it is seen that the 4-ahead and the MVP schemes achieved similar performance. Specifically, their mean throughput are very close (65.9Mbps and 78.95Mbps), and so are their rebuffering duration (0.98s and 0.39s). The same applies to the 10ms scenario (not shown in Table II), where they both produced zero rebuffering. These are due to that both schemes managed to prefetch almost all (88.7% and 99.2%) segments in advance. These results show that when there is 100ms latency and 0.03% packet loss between the MVP edge and the video source, a rigid prefetching policy that always stays 4 segments ahead of the UE's progress can achieve seamless delivery of 4K video at 15Mbps. In other words, since adaptive prefetching is not needed to assure QoE, the MVP edge does not need to stay aware of the UE and network context, which significantly reduces its signaling and computing load incurred by communication with RNIS and inferring user's QoE.

Under the 100ms and the 10ms scenarios, the E2E scheme produced only 0.9s and 0s rebuffering respectively due to its average throughput of 19.7Mbps and 59.31Mbps respectively. However, it is shown in Figure 6 (4) that under the 100ms scenario, after around 80s, the downlink throughput began to drop and eventually fell below 15Mbps due to TCP behavior and possible RAN resource fluctuation, which nearly caused a rebuffering event. This shows that the E2E scheme is unable to *guarantee* seamless delivery of video at 15Mbps over a 100ms backhaul link. On the other hand, under the 10ms scenario, the E2E scheme is easily able to support seamless streaming at 15Mbps thanks to the much reduced latency, which means prefetching is not needed at all in this case.

E. Single UE, 30Mbps Video

We now examine the performance results of a UE streaming video at 30Mbps, which are presented in Figure 7 with statistics summarized in Table III.

The first key observation is that under all 5 scenarios, the MVP scheme is able to support seamless playback of 30Mbps video, which is consistent to its performance when streaming at 15Mbps. Specifically, it achieved very low rebuffering duration compared to the other 2 benchmark schemes, since it managed to prefetch almost all segments in advance (97.6%, 97.6%, 98.4% and 99.2% under 300ms, 200ms, 100ms and 10ms respectively). There is a slight performance degradation under the 500ms scenario, where it managed to prefetch 89.5% (i.e., 111 out of 124) of all segments in time. This is reflected in Figure 7 (1) where the throughput and video buffer fluctuated in the beginning for a few times. This shows that MVP's default strategy, which is prefetching 40s worth of segments in advance, is a little risky under 500ms latency and 30Mbps video requirement.

Unlike the MVP scheme, the 4-ahead scheme's performance when streaming video at 30Mbps is significantly worse than it when streaming at 15Mbps, which is reflected in the higher rebuffering duration. This is intuitive because under the same backhaul condition, it takes longer to prefetch segments at a higher bitrate. For example, under the 200ms scenario, the 4-ahead scheme produced a rebuffering duration of 128.7s when streaming at 30Mbps (compared to only 4.34s when streaming at 15Mbps), which is shown in Figure 7 (3b). This is due to that only 55.6% of all segments are prefetched in advance, and the resulting throughput is too low to support in-time delivery of video at 30Mbps. This means that under the 200ms scenario, the 4-ahead scheme is unable to support seamless delivery of 30Mbps video. Furthermore, under the 100ms scenario, the 4-ahead scheme managed to prefetch only 67.7% of all segments in time. Although this is sufficient to support seamless video playback at 30Mbps (as the rebuffering duration is 2.2s), Figure 7 (4b) shows that the client's buffer fluctuated for over half of the playback session and is at risk of rebuffering should the RAN resource became insufficient or if the video bitrate further increases.

Note that under all scenarios in the 2 groups of experiments above, the 2 prefetching schemes did not improve initial playout delay over the E2E scheme. This is because the first video segment always needs to be retrieved from the video source in all schemes and cannot be prefetched. Also, the

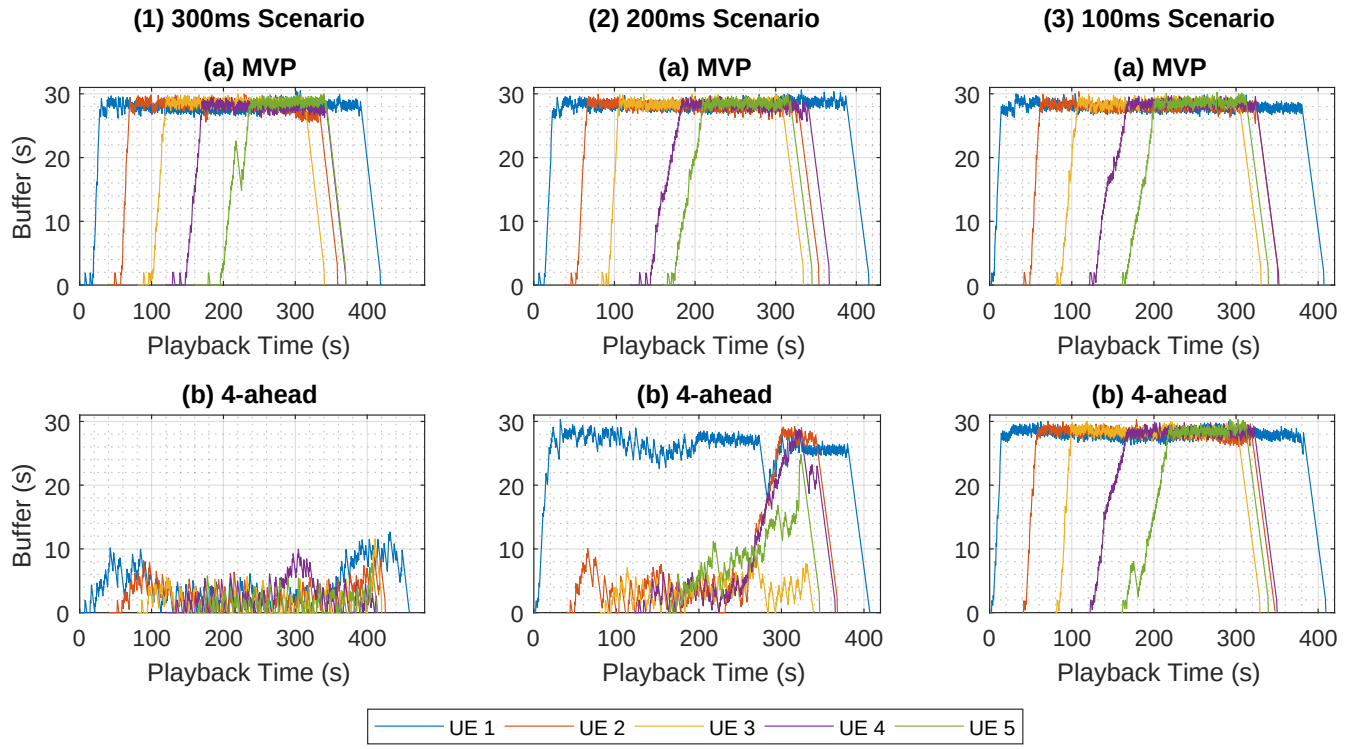


Fig. 8: Multi-UE video buffer performance: 5 UE, 15Mbps video

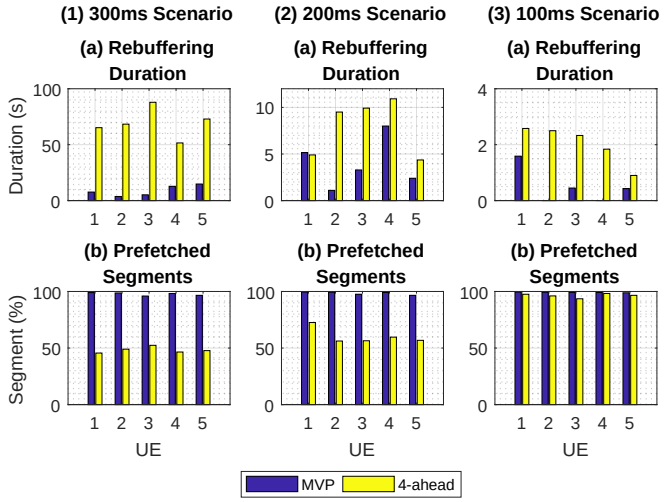


Fig. 9: Multi-UE performance statistics: 5 UE, 15Mbps video

video playback typically starts as soon as the first segment is delivered to a UE (which is the case in many DASH clients such as dash.js), which means the initial playout delay equals the time duration it takes for a UE to download and decode the MPD manifest and the first segment. However, it still can be observed that as the backhaul condition gets better, the overall initial delay at each UE also gets lower.

Through the 2 groups of single-UE experiments above, we draw conclusions on the applicability scenarios of the 3 schemes in order to support QoE-assured 4K and 360 VR video applications. Specifically, at the MVP edge, different levels of video delivery enhancement techniques can be applied for each VoD session with respect to its backhaul condition and video bitrate requirement, which are summarized in Table IV. Note that if a video is streamed at 15Mbps and the backhaul has 100ms latency, the 4-ahead scheme already performs well enough to assure user QoE. However, if a video is streamed at 30Mbps, as long as the backhaul's latency is 100ms or higher, the MVP scheme is required for seamless video playback. Also note that if the backhaul's latency is 10ms, no prefetching is needed to assure user QoE. Considering the 3 schemes' increasing complexity and required context, in order for the MVP edge to avoid unnecessary signaling and computing overhead, it is important that it applies an appropriate technique to each VoD session based on its backhaul condition and its video bitrate requirement.

F. 5 UEs, 15Mbps Video

In this subsection, we present results of experiments using 5 UEs to individually stream 4K videos at 15Mbps in Figures 8 and 9. The 5 UEs stream videos 1 to 5 in Table I respectively, and each UE starts one after another *with a 20-second gap in-between*. The lengths of the 5 videos have been sorted in descending order, so that the duration when there are 5 concurrent VoD sessions is maximized. The main focus of this subsection is 1) to evaluate the MVP and 4-ahead schemes' performance when multiple clients are streaming different videos concurrently; and 2) to verify the applicability conclusions in Table IV. Note that we use 5 UEs to stream at 15Mbps, so that RAN resource is always sufficient to deliver a video segment to a UE in time *as long as it is available at*

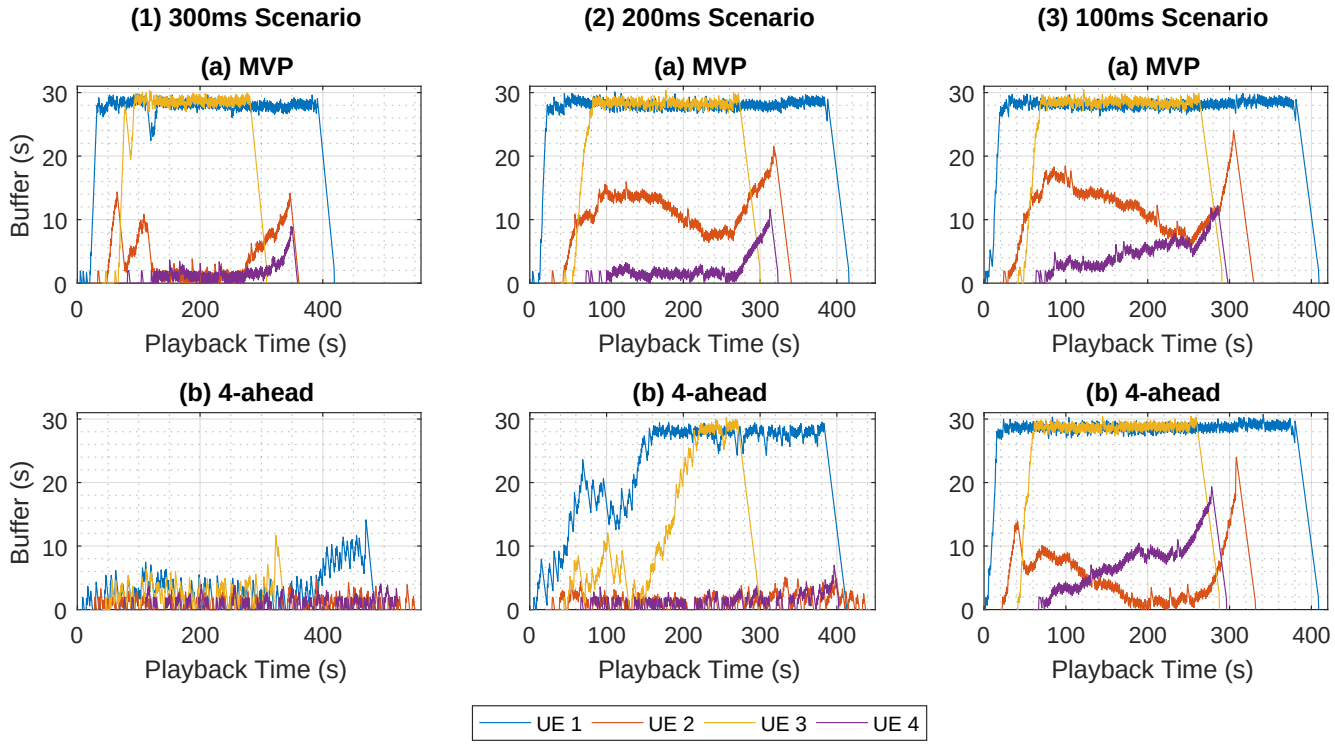


Fig. 10: Multi-UE video buffer performance: 4 UE, mixed (15Mbps and 30Mbps) video

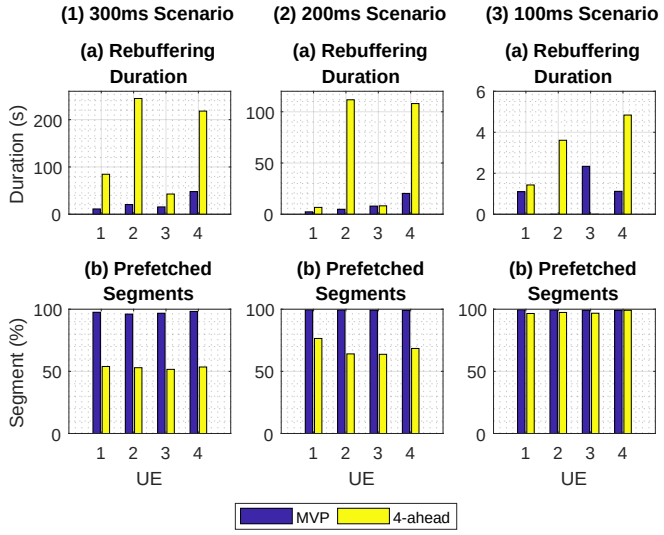


Fig. 11: Multi-UE performance statistics: 4 UE, mixed (15Mbps and 30Mbps) video

the MVP edge (since the RAN capacity is ˜112Mbps). In other words, if a UE experiences rebuffering, it is definitely caused by insufficient performance of the video delivery scheme.

Under the 300ms scenario, the MVP scheme is able to support seamless video playback for all 5 UEs. Specifically, the rebuffering durations experienced by the 5 UEs are 7.7s, 3.7s, 5.2s, 12.8s and 14.8s respectively, and the MVP scheme managed to prefetch 96.6% to 99% of all segments for each UE in advance. In contrast, under the 4-ahead scheme, the 5 UEs experienced rebuffering durations of 65.2s, 68.4s, 87.9s, 51.6s and 72.9s respectively, and only 45.6% to 52.4% of all segments were prefetched in advance. These results verify that when there is sufficient RAN resource, the MVP scheme is able to support seamless video playback at 15Mbps for multiple clients under the 300ms backhaul scenario, while the 4-ahead scheme is unable to do so. Note that the rebuffering events under the MVP scheme are caused by the waiting time while downloading segment 2 (and 3 in some cases), i.e., when the MVP edge is prefetching the subsequent 20 segments. Such waiting time is required for the MVP edge to gain its advance over the client's progress and avoid rebuffering in the future. On the other hand, the 4-ahead scheme spends less time waiting during the initial prefetching stage because it only 4 segments are prefetched. Consequently, it is unable to gain sufficient advance over the client's progress, and rebuffering events kept occurring as the playback continued.

Under the 200ms scenario, the MVP scheme maintains its high performance by prefetching 96.6% to 99.5% of all segments, and the 5 UEs experienced rebuffering durations of 5.1s, 1.1s, 3.3s, 8s and 2.4s respectively. Meanwhile, the 4-ahead scheme's performance has improved as compared to the 300ms scenario, as it produced rebuffering durations of 4.9s, 9.5s, 9.9s, 10.9s and 4.4s for the 5 UEs. Although the 4-ahead scheme's rebuffering durations are higher than the MVP scheme's results, they are generally acceptable from user QoE's perspective. However, as shown in Figure 8 (2b), the buffers of UEs 2 to 5 have fluctuated heavily and are subject to a high risk of rebuffering. This is because only 56.5% to 59.6% of video segments were prefetched for them in advance,

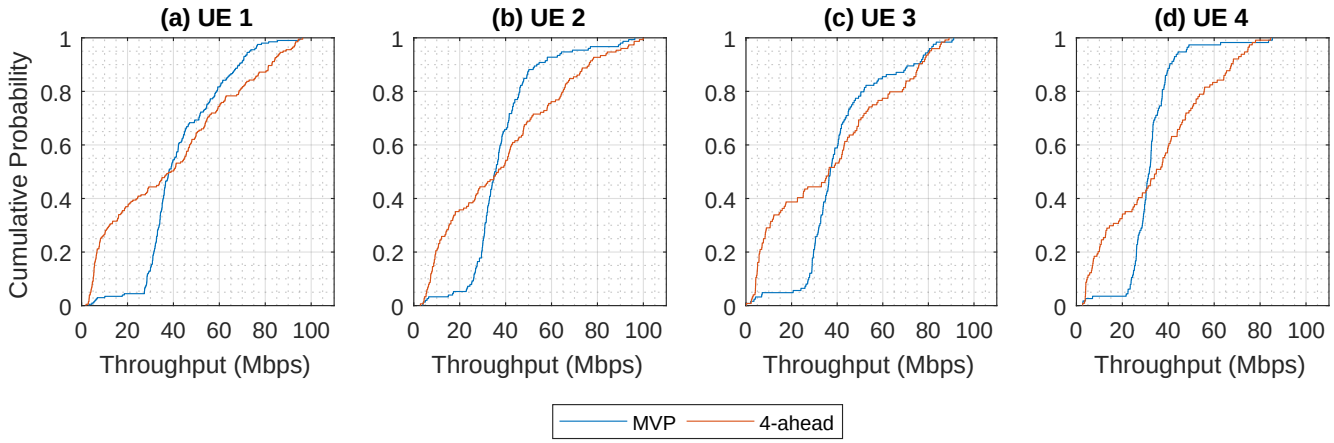


Fig. 12: Multi-UE throughput performance: 4 UE, mixed (15Mbps and 30Mbps) video, 300ms latency and 0.05% packet loss

and as a result, it was more difficult for the UEs to build up their buffers as they need to frequently wait longer to retrieve segments from the remote video source. These results verify that under the 200ms scenario, the MVP scheme can support seamless video playback at 15Mbps for multiple UEs, while the 4-ahead scheme's performance is *just* sufficient despite its significant risk of rebuffering.

Under the 100ms scenario, it is observed that both MVP and 4-ahead schemes achieved similar performance by providing seamless playback of 15Mbps video for all 5 UEs. Specifically, the MVP scheme produced rebuffering durations of 1.6s, 0s, 0.5s, 0s and 0.4s while prefetching 98.9% to 99.5% of all segments, and the 4-ahead scheme produced 2.6s, 2.5s, 2.3s, 1.8s and 0.9s while prefetching 93.5% to 98.2% of all segments. These verify that under the 100ms scenario, the 4-ahead scheme already performs well enough to support seamless playback at 15Mbps.

G. 4 UEs, 15Mbps and 30Mbps Video

In this subsection, we present results of experiments using 4 UEs to stream videos at a mix of 15Mbps and 30Mbps in Figures 10, 11 and 12. The experiment setup is similar to the previous subsection, except that the 4 UEs start streaming videos in the order of 15, 30, 15, 30Mbps with 20-second gaps in-between. These experiments focus on studying how the MVP and 4-ahead schemes' performances are affected by 1) a mix of higher and lower video bitrates and 2) more RAN resource competition than in previous experiments. Note that the video bitrates are chosen here to *just* saturate the RAN capacity, so that there is some mild RAN resource competition (especially between UEs 2 and 4 which stream at 30Mbps).

Under the 300ms scenario, the MVP scheme produced rebuffering durations of 11.2s, 20.5s, 15.6s and 47.9s for the 4 UEs while prefetching 96.1% to 98.2% of segments respectively. Note that UEs 2 and 4 experienced higher rebuffering duration between around 70s and 270s, i.e., when all 4 UEs are streaming concurrently. These are caused by RAN resource competition rather than insufficient performance of the MVP scheme, since almost all segments were prefetched in advance. To verify this, we plot the cumulative distribution function (CDF) of the 4 UEs' per-segment downlink throughput in Figure 12. It is shown that UE 2's downlink throughput mostly fell between 25Mbps and 50Mbps, and UE 4's throughput mostly fell between 20Mbps and 40Mbps. These numbers roughly match the setup that each of the 4 UEs gets allocated with a fair share of 112Mbps / 4 = 28Mbps. Therefore, it is verified that UEs 2 and 4's rebuffering are caused by RAN resource competition only.

Meanwhile, under the 4-ahead scheme, the 4 UEs experienced rebuffering of 84.6s, 244.8s, 42.8s and 218.1s, while only 51.6% to 53.9% of segments were prefetched in time. As a result, as shown in Figure 12, all 4 UEs experienced a wide distribution of downlink throughput between 5Mbps and 70Mbps. Note that UEs 1 and 3's rebuffering durations are similar to their corresponding results in the 5-UE experiments. Since their allocated RAN resources are much higher than the required 15Mbps, the bottleneck is the 4-ahead scheme's performance. On the other hand, UEs 2 and 4 experienced higher rebuffering durations than the 4-ahead scheme's result in the single-UE experiment (188.1s). This is because although a similar % of segments was prefetched (54%), each UE's allocated RAN resource is a lot less than in the single-UE experiment. Hence, in this case, the RAN resource competition *further* affected the 4-ahead scheme's performance.

Under the 200ms scenario, the MVP scheme managed to prefetch 99.1% to 99.5% of all segments in advance, and the 4 UEs experienced 6.7s, 4.9s, 8.2s and 20.3s of rebuffering respectively. Note that UE 4's higher rebuffering duration was caused by RAN resource competition. On the other hand, the 4-ahead scheme managed to prefetch 63.7% to 76.5% of all segments in advance, and the 4 UEs experienced 2.4s, 111.7s, 8s and 108s of rebuffering respectively. UEs 2 and 4's rebuffering durations were similar to the 4-ahead scheme's result in the single-UE experiment (128.7s), which means the RAN resource competition did not affect the 4-ahead scheme's performance during this experiment.

Under the 100ms scenario, both MVP and 4-ahead schemes were able to support seamless playback for all 4 UEs. Both schemes prefetched almost all segments in advance (at least

96.8% by 4-ahead and 99.1% by MVP), and the rebuffering durations were both low (up to 4.8s by 4-ahead and 1.4s by MVP). These results are in-line with all previous experiments.

It is worth noting that while the MVP scheme can prefetch almost all video segments to the network edge in advance, some UEs still experienced rebuffering due to RAN resource competition. Therefore, the MVP scheme can ensure a user's QoE only if sufficient RAN resource is available (i.e., $\geq$ video bitrate) in the first place. From an MNO's perspective, if it wants to offer QoE-assured 4K VoD delivery to its subscribers through the MVP system, it may need to apply admission control or boost its RAN capacity so that each UE can get enough RAN resource to meet its video bitrate requirement.

To summarize, through the 4 groups of experiments above, we have validated Table IV's conclusions on the applicability of E2E, 4-ahead and MVP schemes to assure video users' QoE. Specifically:

- The MVP scheme is always capable of assuring QoE by providing seamless playback, regardless of backhaul condition (up to 500ms latency) and video bitrate requirement (up to 30Mbps).
- When streaming at 15Mbps, if the backhaul latency is 100ms, the 4-ahead scheme already performs well enough to assure QoE.
- When streaming at 30Mbps, as long as the video is not available at a local (e.g., CDN) server, the MVP scheme is required to assure QoE. The 4-ahead scheme's performance is insufficient even when the backhaul latency is only 100ms.
- If a video is available at a local source ($\leq$10ms backhaul latency), no prefetching is needed at all since QoE is already well enough.

VII. Conclusion

In this paper, we have presented a novel video-on-demand (VoD) delivery system named MVP (Mobile edge Virtualization with adaptive Prefetching), which advocates the idea of content providers deploying their content intelligence as a virtual network function (VNF) into the mobile network operator's (MNO) infrastructure. Based on this, we present a novel context-aware adaptive video prefetching scheme and a QoE inference mechanism that can be deployed at the MVP function, which aim to improve user QoE especially when the video source is located remotely in the public Internet. Furthermore, the MNO can deploy mobile edge computing (MEC) servers at its network edge to provide context information feedback on radio access networks' (RAN) condition to the MVP function, which serves as input to its prefetching scheme's decision-making process. Such a system is the first that can support Internet scale 4K VoD delivery with QoE assurance. Furthermore, it fits in current LTE networks and can be readily adopted by MNOs and content providers.

Our proposed system and adaptive prefetching scheme have been deployed in a real LTE-A network infrastructure, and their performance have been evaluated under an extensive range of realistic network scenarios while streaming 4K videos at 15Mbps and 30Mbps. We have demonstrated that based on different backhaul conditions and video bitrate requirements, different levels of video delivery enhancement techniques are needed to assure 4K VoD users' QoE. Specifically, the MVP scheme is always capable of achieving seamless playback even under extreme backhaul conditions and video bitrate at up to 30Mbps. However, it is not always *required* for QoE-assured streaming. For example, if a video is streamed at 15Mbps and the backhaul has 100ms latency and 0.03% packet loss, the 4-ahead scheme is already sufficient to assure QoE. Furthermore, if a video is available at a local content source with 10ms backhaul latency, then no prefetching is needed at all to assure QoE. These conclusions have provided valuable practical insights into how the MVP edge should *selectively* enable prefetching and/or intelligence for different users to avoid consuming unnecessary signaling and computing overhead.

The MVP system also has its limitations and remaining challenges. From a business perspective, if a content provider wants to distribute its content using the MVP model, it may need to collaborate with many MNOs if the target users are geographically distributed among e.g., multiple continents. In this sense, the MVP model is complementary to the conventional CDN-based content delivery model, where the former is more suitable for regional content distribution, while the latter works better for global content delivery. Another potential challenge is that when a UE is traveling while streaming a video, it may leave one MVP edge's coverage and enter another one's. In this case, the 2 MVP edges need to perform extra signaling and coordinate their prefetching activities to ensure such handover does not disrupt the user's QoE. We leave this as a future work.

Acknowledgment

This work is supported by EPSRC CONCERT (EP/L018683/1) and KCN (EP/L026120/1) projects.

The authors would like to acknowledge the support of University of Surrey's 5GIC (http://www.surrey.ac.uk/5gic) members for this work. The authors would also like to thank Hermann Hellwagner (AAU Klagenfurt) and Richard Bradbury (BBC) for their constructive insights on this paper.

References

[1] Cisco VNI Mobile Forecast (2015-2020). [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[2] ISO. ISO/IEC 23009-1:2014 dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=65274

[3] F. Bronzino, D. Stojadinovic, C. Westphal, and D. Raychaudhuri, "Exploiting network awareness to enhance DASH over wireless," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2016, pp. 1092–1100.

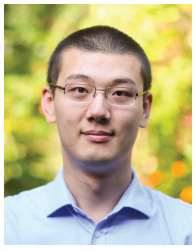
[4] Caching Details — Cloud CDN Documentation — Google Cloud Platform. [Online]. Available: https://cloud.google.com/cdn/docs/caching

[5] Mobile Edge Computing - a key technology towards 5G. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

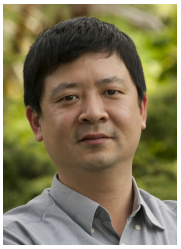
[6] C.-Y. Chang, K. Alexandris, N. Nikaein, K. Katsalis, and T. Spyropoulos, "MEC architectural implications for LTE/LTE-A networks," in *Proc. MobiArch '16*. ACM, 2016, pp. 13–18.

[7] Mobile edge computing use cases & deployment options. [Online]. Available: https://www.juniper.net/assets/us/en/local/pdf/whitepapers/2000642-en.pdf

[8] Watching video in 4K Ultra HD on Fire TV. [Online]. Available: https://www.amazon.co.uk/gp/help/customer/display.html?nodeId=201859000
[9] Recommended upload encoding settings. [Online]. Available: https://support.google.com/youtube/answer/1722171?hl=en-GB
[10] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proc. MMSys'11*. ACM, 2011, pp. 157–168.
[11] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*. ACM, 2012, pp. 97–108.
[12] T. C. Thang, Q. D. Ho, J. W. Kang, and A. T. Pham, "Adaptive streaming of audiovisual content using MPEG DASH," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 78–85, February 2012.
[13] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, Apr. 2014.
[14] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: A QoE-aware DASH System," in *Proc. MMSys'12*. ACM, 2012, pp. 11–22.
[15] A. E. Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-based traffic and resource management for adaptive HTTP video delivery in LTE," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 988–1001, June 2015.
[16] A. Beben, P. Wiśniewski, J. M. Batalla, and P. Krawiec, "ABMA+: Lightweight and Efficient Algorithm for HTTP Adaptive Streaming," in *Proc. MMSys'16*. ACM, 2016, pp. 2:1–2:11.
[17] C. Zhou, C. W. Lin, and Z. Guo, "mDASH: A Markov Decision-Based Rate Adaptation Approach for Dynamic HTTP Streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 738–751, April 2016.
[18] X. Wang, T. Kwon, Y. Choi, H. Wang, and J. Liu, "Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 72–79, June 2013.
[19] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri, "Quality-adaptive prefetching for interactive branched video using http-based adaptive streaming," in *Proceedings of the 22Nd ACM International Conference on Multimedia*. ACM, 2014, pp. 317–326.
[20] ——, "Bandwidth-aware prefetching for proactive multi-video preloading and improved has performance," in *Proceedings of the 23rd ACM International Conference on Multimedia*. ACM, 2015, pp. 551–560.
[21] S. Wilk, D. Schreiber, D. Stohr, and W. Effelsberg, "On the effectiveness of video prefetching relying on recommender systems for mobile devices," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2016, pp. 429–434.
[22] N. Master, A. Dua, D. Tsamis, J. P. Singh, and N. Bambos, "Adaptive prefetching in wireless computing," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3296–3310, May 2016.
[23] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, Feb. 2014.
[24] M. Claeys, N. Bouten, D. D. Vleeschauwer, W. V. Leekwijck, S. Latr, and F. D. Turck, "An announcement-based caching approach for video-on-demand streaming," in *2015 11th International Conference on Network and Service Management (CNSM)*, Nov 2015, pp. 310–317.
[25] J. O. Fajardo, I. Taboada, and F. Liberal, "Improving content delivery efficiency through multi-layer mobile edge adaptation," *IEEE Network*, vol. 29, no. 6, pp. 40–46, Nov. 2015.
[26] C. Ge, N. Wang, S. Skillman, G. Foster, and Y. Cao, "QoE-Driven DASH Video Caching and Adaptation at 5G Mobile Edge," in *Proc. ICN'16*. ACM, 2016, pp. 237–242.
[27] S. Chen, B. Shen, S. Wee, and X. Zhang, "Segment-based streaming media proxy: modeling and optimization," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 243–256, April 2006.
[28] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An SDN architecture with DASH assisting network elements," in *Proc. MMSys'16*. ACM, 2016, pp. 4:1–4:10.
[29] K. Dong, J. He, and W. Song, "QoE-aware adaptive bitrate video streaming over mobile networks with caching proxy," in *2015 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2015, pp. 737–741.
[30] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri, "Helping hand or hidden hurdle: Proxy-assisted HTTP-based adaptive streaming performance," in *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, Aug 2013, pp. 182–191.
[31] S. Ebrahimi-Taghizadeh, A. Helmy, and S. Gupta, "TCP vs. TCP: a systematic study of adverse impact of short-lived TCP flows on long-lived TCP flows," in *Proc. INFOCOM'05*. IEEE, March 2005, pp. 926–937.
[32] N. Bouten, S. Latr, J. Famaey, W. V. Leekwijck, and F. D. Turck, "In-network quality optimization for adaptive video streaming services," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2281–2293, Dec 2014.
[33] Sprint Network Performance. [Online]. Available: https://www.sprint.net/sla_performance.php
[34] R. Huysegems et al., "Session reconstruction for HTTP adaptive streaming: Laying the foundation for network-based QoE monitoring," in *Proc. IWQoS*. IEEE, June 2012.
[35] 3GPP TR 38.913. [Online]. Available: www.3gpp.org/DynaReport/38913.htm
[36] Huawei Lampsite. [Online]. Available: http://carrier.huawei.com/en/products/wireless-network/Small-Cell/LampSite
[37] Optimal adaptive streaming formats MPEG-DASH & HLS segment length. [Online]. Available: https://bitmovin.com/mpeg-dash-hls-segment-length/



Chang Ge is a Research Fellow at the 5G Innovation Centre, Institute for Communication Systems, University of Surrey, UK. He received B.Eng. (Honours) degree in telecommunication engineering from Queen Mary, University of London in 2009, and M.Sc. and Ph.D. in electronic engineering from University of Surrey in 2010 and 2015 respectively. His research interests include efficient video streaming, quality-of-experience management and context-aware network management in 5G networks.



Ning Wang is a Reader at the 5G Innovation Centre, Institute for Communication Systems, University of Surrey, UK. He received his B.Eng (Honours) degree from the Changchun University of Science and Technology, P.R. China in 1996, his M.Eng degree from Nanyang University, Singapore in 2000, and his PhD degree from the University of Surrey in 2004 respectively. His research interests mainly include mobile content delivery, context-aware networking, network resource management.

Gerry Foster is a 5G Systems Architect at the 5G Innovation Centre, Institute for Communication Systems, University of Surrey, UK. His wireless career spans GSM, UMTS and LTE R&D at Motorola, Lucent and Teoco. His current interests include Next Generation Architecture, Content and Context Aware Networks and the Internet of Things. He has worked on 3GPP, ETSI and IETF standards and is currently contributing to the ETSI NGP initiative to evolve network protocols for next generation networks. He is a Chartered Engineer of the IET, and holds a B.Sc. (Honours) in Communications Engineering from Plymouth University.

Mick Wilson is a Research Fellow for Fujitsu Laboratories of Europe in the Future Networks Research Division. He has led FLE research activities in a number of diverse communications research collaboration projects including radio access networks (3G, LTE, WiMAX), sensor networks, autonomic networking and 5G. He has actively contributed to 3G, 4G and 5G standardization within 3GPP, ETSI and IEEE. He is currently acting as standardization manager for the H2020 SESAME project investigating Cloud enabled small cell clusters at the network edge. He was also a director of the UK funded ICT KTN and previously worked for Roke Manor Research on Intelligent Networking and Network Management.