Enhancing Video QoE Over High-speed Train Using Segment-based Prefetching and Caching

Yue Cao, Ning Wang, Celimuge Wu, Xu Zhang and Chakkaphong Suthaputchakun

Abstract

The big picture of 5G will bring a range of new unique service capabilities, where ensuring Quality of Experience (QoE) continuity in challenging situations such as high mobility, e.g. on-board User Equipments (UEs) in High Speed Train (HST) is one of sharp killer applications. In this paper, we propose a Mobile Edge Computing (MEC) driven solution to improve QoE, for UEs in the HST with perceived Dynamic Adaptive Streaming over HTTP (DASH) video demands. Considering the challenging wireless communication conditioning (e.g., path loss and Doppler Effect due to high mobility) between HST and Base Station (BS) along the railway for enabling progress and seamless video consuming, the case study shows the benefit of MEC functions mainly from content prefetching and complementarily from content caching, over benchmark solution where UEs solely download video segments through challenging wireless channel.

I. Introduction

The capability of handling high mobility scenarios has been commonly recognized as a key feature for the Fifth Generation of mobile networks (5G), and one of the killer applications in this case is the support of Quality of Experiences (QoE) assurance in video applications when User Equipments (UEs) are in high mobility modes, e.g., during a High-Speed Train (HST) journey [1]. In the legacy 4G networks, the direct communication between on-board UE and the encountered Base Stations (BSs) along the railway, is unable to provide adequate or assured data throughput for supporting video applications, mainly due to the substantially varied signal strength, path loss, as well as the Doppler Effect.

A. Motivation

In the context of 5G research and innovation, while cutting edge radio technologies are being developed for supporting substantially increased throughput capacities, network and content layer intelligence will also play an important role in supporting seamless QoE assurance for UEs in a complementary manner. Motivated by this gap, we focus on the emerging Dynamic Adaptive Streaming over HTTP (DASH) [2] video application as the representative scenario, in which case a video content is chunked into small segments to be independently downloaded and consumed by the HST on-board UEs.

B. Advance over Literature & Challenges

Previous works [3] mainly implemented video streaming for fixed Internet, where throughput-based solutions suffer from the challenges of delay variation, while buffer-based solutions examine queue length at clients side for video adaption. For provisioning of video streaming application in challenged, intermittent connectivity environments, [4] consider the coordination between stable WiFi communication (when the train stops at station) and cellular network (while train is on-the-move), for video downloading application. Analytic results have reflected that it is difficult to maintain undisrupted playback, especially in case of high bit rates.

Y.Cao (yue.cao@lancaster.ac.uk) is with the School of Computing and Communications, Lancaster University, UK; N.Wang is with the Institute for Communication Systems, University of Surrey, UK; C.Wu is with the Department of Computer and Network Engineering, Japan; X.Zhang is with the Department of Computer Science and Engineering, Xi'an University of Technology, China; C.Suthaputchakun is with the Electrical and Computer Engineering Department Engineering School, Bangkok University, Thailand.

Following above trajectory, new challenges to fulfill such application for on-board users on the (on-the-move without stop) HST are as follows:

- If a video session is based on the direct cellular connection between the UE and associated BSs along the HST journey, the actual data throughput performance can vary substantially at different time period.
- Also, the large number of on-board UEs in HST (hundred level) competing the wireless communication resource will be much critical, than the use case of enabling video service between pairwise on-board UEs in vehicles (or traditional Internet applications).

As such, the key technical strategy is how to smooth UE experienced data rate during DASH video consumption sessions. This is achieved through smart content prefetching and caching techniques, with critical throughput shared by multiple UEs.

C. Methodology

The rapid growth of mobile applications have placed severe demands on cloud infrastructure, which has led to moving computing and data services towards the edge of the cloud, resulting in a novel Mobile Edge Computing (MEC) [5] architecture. MEC could reduce data transfer times, remove potential performance bottlenecks, increase data security and enhance privacy while enabling advanced applications. The MEC inherently covers the features of communication, computation and caching. While the principle of such caching strategies have been extensively studied in the use case of content preloading at railway stations [4], in this paper we focus on the prefetching (the communication feature of MEC positioned at the HST), and leave computation resource allocation and caching for broad readers.

Here, we introduce a system that primarily enables video content prefetching [6], and complementary caching [7] operated by the MEC server, to be fulfilled by the HST as a moving content proxy for serving video application demands from on-board UEs. To enable aforementioned functions, the HST will be equipped with MIMO antennas for front-haul connection with BSs, on-board WiFi for UEs connection, as well as the content storage space for maintaining prefetched/cached segments to be ready for serving UEs. Specifically, triggered by the initial DASH video request events, the HST (acting as a moving content proxy for handling incoming requests) can prefetch a sequence of follow-up video segments when the instantaneous front-haul data throughput (between the on-board MIMO antennas and an encountered BS) is high, and temporarily store them at the local cache before the connectivity deteriorates when the HST is moving away from its associated BS:

- When the front-haul connectively is poor, the HST can locally serve via on-board WiFi the prefetched content during the period, till it gets close to the next BS when segment prefetching can be resumed.
- In addition, depending on the content popularity statistics during the train journey, video segments can be further cached locally in order to serve future requests of the same content from common UEs.

D. Innovations

Under what conditions will prefetching really take effects? It can be inferred that prefetching will only become effective, when the available data throughput is significantly dynamic along time. As such, it is interesting to investigate how the peak-valley distance between the available data throughput (shared by on-demand UEs) can affect the overall QoE perfecting effectiveness:

- In case the available data throughput is either too high or too low for per UE's session, prefetching may not take effect. Specifically, there is no need to prefetch, if the valley total data throughput can adequately support direct streaming, between all on-board UEs and encountered BSs.
- Also, the limited front-haul peak throughput basically does not provide any opportunity to prefetch.

Understanding such constraints will shed light on under what conditions the prefetching technique should be implemented, depending on the throughput shared by each UE.

How to make smart situation/condition-based decision making? For instance, the expected playback duration of prefetched video segments should not go beyond the time that UEs consume buffered video at devices.

- This brings the decision making on which UEs' sessions should be scheduled timely by the HST, in line with the above condition on whether to prefetch.
- Complementarily, the HST as the content proxy should have the knowledge about "on-board" content popularity (depending on how many times it is requested), which can be used for making content caching/replacement decisions.

II. Basic Operations on Prefetching And Caching

A. QoE Improvement Technology over Streaming

In literature, the most common ones include initial start-up delay, disruption and frequency, overall video quality etc. These metrics apply to both on-demand and live streaming. Furthermore, for live streaming receivers, there is the additional metric of live stream latency, which indicates how far is the receivers playback behind the video sources production progress. It is discussed in [8] that abruptly downgrading video quality during playback would significantly deteriorate QoE.

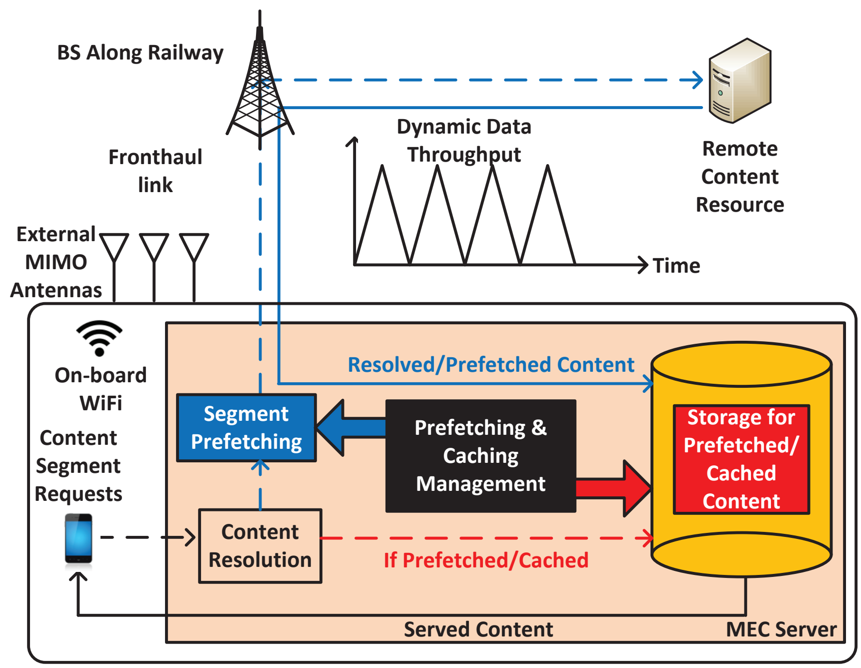

Fig. 1. Overview of Edge Computing System

According to the proposed system, instead of using the direct cellular connection between individual on-board UEs and the encountered BSs along the railway, the HST itself acts as a moving content proxy for locally serving DASH video requests originated from UEs. As such, the end-to-end connectivity involves the internal communication link (e.g. on-board WiFi system), and the common cellular-based communication (link between the HST with MIMO-based antenna and encountered BSs during journey). While this resembles the scenario of having conventional on-board WiFi services already available nowadays, the key difference is the embedded content intelligence for prefetching at the MEC server positioned at HST.

As shown in Fig. 1, a DASH-based video content is chunked into multiple small but independent segments, which can be progressively downloaded by UEs based on TCP during playback (solid line). Upon launching a DASH video session, a UE starts to make subsequent segment requests (dash line) to the MEC server. Here, the MEC server has the following content-oriented functionalities in the architecture.

B. Content Resolution

In line with the system cycle in Fig. 2, this is to resolve each incoming segment request, to either the remote content source through the external cellular link (just in case the wireless channel resource is sufficient), or the local content storage where the requested segment has been perfected or cached.

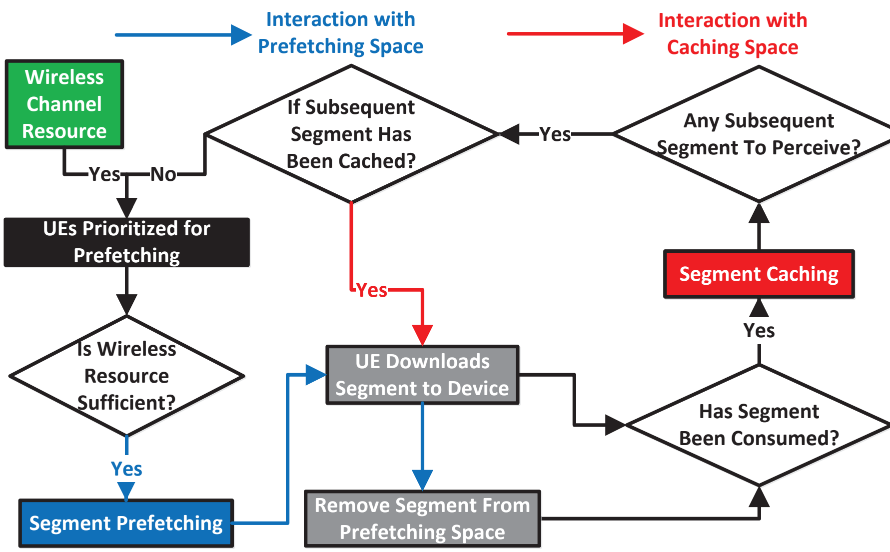

Fig. 2. System Cycle

C. Segment Prefetching

1) General Procedure: Once the MEC server has received the request of initial video segment which has to be resolved to the remote content source, it will additionally launch a sequence of requests for follow-up segments, without waiting for the incoming requests from UE side. Once the content segments requested by the MEC server has been delivered back, they will be locally stored at the content storage space, and wait to be served upon the arrival of actual requests originated from UE.

In order to systematically fulfil the prefetching task, the MEC server needs to determine how many DASH segments can be prefetched in advance for each UE's on-going video session, which is based on its knowledge about the available data throughput, the number of active sessions, as well as other factors such as the content characteristics (e.g. video segment size and length). The common strategy is to prefetch as many segments as possible, such that these prefetched segments can be locally served to the users sustainably during the period (particularly when the throughput becomes limited).

2) Analysis on Benefit of Prefetching: In this context, one technical issue is an accurate knowledge about the available throughput (to be shared by on-board UEs), based on which the MEC server is able to determine how much to prefetch for each UE's session. The good news is that the train speed is normally predictable during the whole journey, and hence this provides relatively deterministic information to compute instantaneous throughput by the MEC server. Details on how to compute the actual throughput will be presented in the next section.

We define following notations to facilitate the analysis of prefetching decision making.

- Segment consumption time: C
- Video streaming rate of UE: R
- Segment size: S
- Number of UEs downloading from HST WiFi: $D \geq 1$
- Number of UEs' sessions being scheduled for prefetching: Z where $D \geq Z \geq 1$
- Downlink data rate from BS to HST: $B_{cur} \in \{B_{min}, B_{max}\}$

- Number of segments buffered at UE device: X
- Number of segments downloaded but not yet buffered: K
- HST onboard WiFi speed: W

With the segment size given as $S = R \times C$, the gain of prefetching happens between:

$$\left(\frac{S \times D}{W} + \frac{S \times Z}{B_{max}}\right) \leq \frac{(X+K) \times S}{R} < \left(\frac{S \times D}{W} + \frac{S \times Z}{B_{min}}\right) \quad (1)$$

Note that $X = \frac{30}{C}$ refers to the number of segments X can be buffered up to 30s at UE device (i.e., a device can buffer 15 segments with $C = 2$s), as specified in DASH. X can be monitored by MEC server collocated at HST, as UEs are connected to HST through on-board WiFi. Here we have:

- On the one hand, Equation (1) means the UE (each shares $\frac{B_{cur}}{Z}$ data rate) perceived time to consume buffered (and downloaded but not yet buffered) segments, as for $\frac{(X+K)\times S}{R}$, much not be always shorter than the shortest time $\frac{S\times Z}{B_{max}}$ to fetch[1] a segment from the BS to HST (by also taking into account the shared WiFi among other $(D-1)$ UEs to download the segment from HST to UE device). Otherwise, the disruption will always occur.
- On the other hand, $\frac{(X+K)\times S}{R}$ should be always shorter than the longest time $\frac{S\times Z}{B_{min}}$ to prefetch a segment (plus segment downloading time $\frac{S\times D}{W}$). Otherwise, no disruption will occur.

Here, as $K > 1$ is achievable thanks to the MEC functioned prefetching at HST, it much helps to alleviate the disruption, than the case where UE directly fetches segments from the BS (no subsequent segments will be requested in advance, with $K = 1$). We observe that a shorter segment consumption time C and lower UE perceived rate R, are beneficial to disruption free. Furthermore, in case of long C, e.g., 10s, less segments can be buffered while it also takes a longer time to fetch a segment from the BS. *Therefore, the influence of K would be of much importance to alleviate the disruption (waiting to fetch subsequent segments from BS), deemed as the benefit of prefetching.*

D. Segment Caching

As soon as a requested content segment has arrived at the content storage (regardless whether it is yet to be served to the original requesting UE or it has already been delivered), the content segment is cached at the storage space so that it can be served again locally upon future requests from other UEs. Depending on the caching space limit, less popular content segments can be evicted from the cache, and in the literature there have been some mature content eviction mechanisms for handling this, such as Least Frequently Used (LFU) or Least Recently Used (LRU) [7]. *It has been well known that caching itself can reduce the access delay, and thus not bring prefetching delay from the BS to HST.*

III. Techniques for Realising Prefetching/Caching by MEC Server

Now we present specific algorithms for enabling smart prefeching and caching operations at the MEC server. The first task is to compute the available data throughput, which is used as knowledge input to determine how many segments can be prefetched in a given period of time. As mentioned previously, due to the predictable train mobility patterns (especially its velocity), data throughput can be derived from context information associated with the HST.

A. Mobile Backhaul Channel Quality Estimation

The prefetching gain mainly comes from efficient exploitation of the mobile backhaul-channel time diversity. Basically, the channel quality[2] of mobile backhaul varies widely in time, due to the mobility of HST. Given an accurate prediction of the backhaul channel quality, the HST is able to form an optimal

[1] The term fetch and download are exchangeable, as both refer to one segment per each action.

[2] Ideally, it is assumed that the effect of trees, train infrastructure (electric power supply), and the uncertain dynamics that occur in the movement of high-speed trains. While other emerging 5G physical technologies, e.g., millimeter waves, will be of interest for integration.

(or sub-optimal) decision on the data volume that should be downloaded through the mobile backhaul. Despite its promise in theory, practical implementation of the prefetching technology in HST applications faces the critical issue of high mobility, which is challenging the hypothesis of accurate predication of the backhaul channel quality.

1) Physical Layer Challenges: The coherence time is an important and fundamental element to investigate the impact of mobility at physical layer. It is utilized to measure the channel quality correlation between two received signals. More explicitly, the channel quality often changes considerably, when the time difference between two received signals goes beyond the coherence time. As far as the HST application is concerned, the following shows the coherence time for a set of typical velocities assuming the mobile backhaul operating at 3 GHz central frequency.

- 500 km/h HST Velocity; 1.39 kHz Max Doppler Frequency; 0.13 ms Coherence Time
- 300 km/h HST Velocity; 0.83 kHz Max Doppler Frequency; 0.22 ms Coherence Time
- 120 km/h HST Velocity; 0.33 kHz Max Doppler Frequency; 0.54 ms Coherence Time

Basically, the coherence times are a fraction of millisecond, and they are shorter than the typical frame length (i.e., 1 ms) recommended by 3GPP for both current and next-generation mobile communication systems. In other words, the backhaul channel quality will change during the transmission of a data frame, and more considerably it will be very different between two consecutive frames. Such renders the channel quality prediction and communication rate adaptation rather a challenging task.

In order to tackle this challenge, one of possible approaches is to allow the HST mobile backhaul operating at lower carrier frequencies, e.g., 700 MHz, as recently shortlisted by OFCOM for 5G mobile systems. Such will immediately increase the coherence time by more than 4 times. Nevertheless, the updated coherence time is still too small to support the accurate channel-quality prediction, and in higher mobilities (e.g. 300 km/h or above) the backhaul channel will still vary considerably within the frame duration.

Fortunately, the high-mobility HST backhauls mostly operate in wide areas that are not rich in multipath propagations. Despite small in coherence time, the backhaul channel quality is dominated only by a small number of reflectors or scatters; and in this case the backhaul channel varies almost deterministically. Then, deterministic time-varying channel models (such as the basis expansion models [9]) can be employed to facilitate the channel quality prediction; and the air interface should be carefully designed and optimized.

2) Estimation of Downloading Rate: In our case study, we calculate the downlink transmission rate for the HST to prefetch video frame from the BS, as follows:

1) **Step 1:** Mapping from basic physical layer/radio waveform parameters to the output Signal-to-Interference-plus-Noise Ratio (SINR), based on calculation from the link-level perspective.
2) **Step 2:** Mapping from the output SINR from Step 1, to the achievable throughput, by considering the system-level parameters and overhead.

In Step-1, the system settings such as bandwidth, waveform modulation, BS transmission power, channel path-loss and Doppler Effect etc., are taking into the consideration. To make our calculations keep in step with the progress of the 5G standardizations, we use as much parameters that have been agreed by The 3rd Generation Partnership Project (3GPP) in recent 5G standard meeting as possible. Other parameters yet to be decided will follow the Long Term Evolution (LTE) based system. For example, we assume 100 MHz bandwidth with subcarrier spacing being 15 KHz is allocated to the system [10]; In addition, a new waveform (e.g., filtered orthogonal frequency division multiplexing (f-OFDM) [11]) with 4$\%$ guard band is assumed. In such case, 6400 subcarriers are available for data transmission. The transmission power (46 dBm), antenna gain (14 dB at BS and 0 dB at UE) and noise figure (9 dB at UE) follow the values defined in LTE by 3GPP [12]. The channel path-loss model for the viaduct high speed railway is modelled as Equation (3) in [13]. With these settings, the output SINR and capacity will be first calculated by computer simulations.

In Step-2, the mapping from the wireless channel capacity (based on the output SINR) to achievable throughput depends on multiple parameters of the system, such as the radio frame structure and system overhead (both time interval and signalling, etc.). Based on the LTE-like system, the system throughput can be approximated by [14]:

$$\text{System Throughput} = \gamma \log_2 \left(1 + \frac{1}{\beta}\text{SINR}\right) \tag{2}$$

where γ is the system bandwidth efficiency, and β is the parameter to adjust for the SINR implementation efficiency of system (e.g., LTE). The typical values $\gamma = 0.75$; $\beta = 1.25$ for SIMO system (2 antennas at the BS) [13] will be adopted in case study.

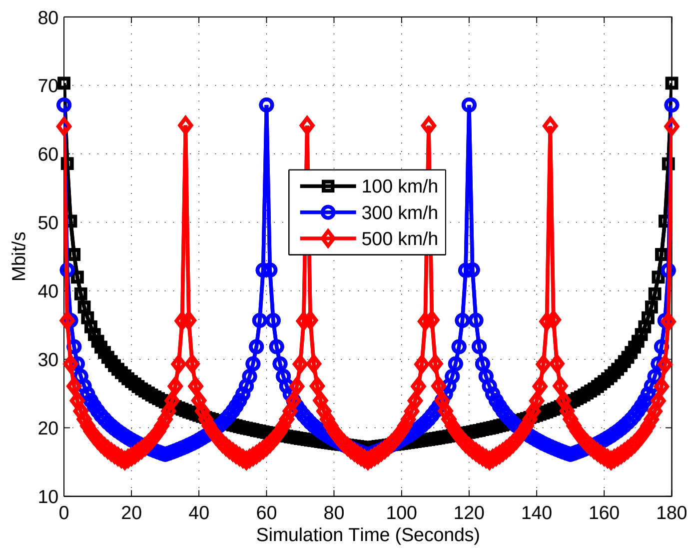

Fig. 3. Downlink Data Rate

Fig. 3 shows throughput to be used in case study. Supposing that the minimum distance between HST and BS is 15 m, the cellular radius is 2500 m and adjacent BS distance is 5000m [1]. Due to the high mobility of HST, both the path loss and Doppler Effect influence the downlink data rate from the BS to HST. A key observation is obtained, where a faster HST speed (e.g., 300-500 km/h v.s. 100 km/h) affects the availability of connectivity in terms of time duration (width of waveform, mainly due to the path-loss) and degree (the vertical value of waveform, mainly due to the Doppler Effect).

B. Proposed Prefetching Strategy

1) Whether to prefetch: Due to the highly dynamic downlink data rate and unpredictable UE requests, it is difficult to decide whether to prefetch a segment (the newly generated request inevitably would occupy the B_{cur}, thus impacts the prefetching for existing UEs). Here, we propose a heuristic solution by following the analysis in Equation (1):

- Initialize Z as 1.
- By knowing the current downlink data rate shared by each UE, the time $\frac{S \times Z}{B_{cur}}$ to prefetch a subsequent segment, must not be longer than the time $\frac{(X+K) \times S}{R}$ to consume segments in UE's buffer (additionally consider those have been prefetched by the HST but not yet been buffered), given by $\frac{S \times Z}{B_{cur}} \leq \frac{(X+K) \times S}{R}$.
- If above step does not hold, the HST will schedule prefetching for this UE, as most likely disruption will occur. Then, the Z is increased by 1.
- Above two steps are repeated, until all on-board UEs have been processed. In the meanwhile, Z will reduce by 1, upon a scheduled segment prefetching has been finished.

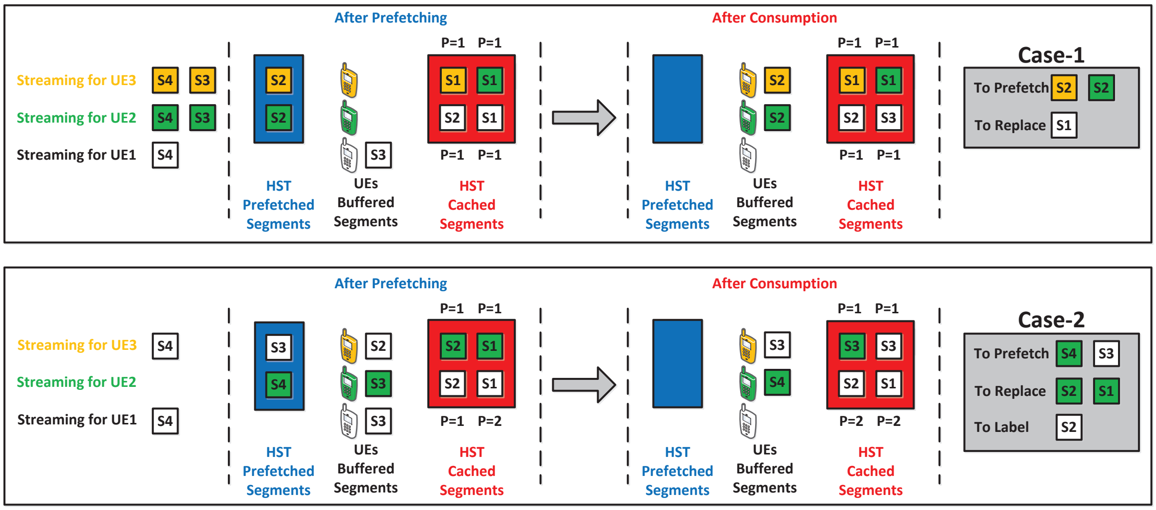

Fig. 4. Example of Prefetching and Caching Strategies

2) Whom to prefetch: We further propose a fairness driven policy, to prioritize the prefetching scheduling among UEs. Here, the UE with the least value of $(X + K)$, will be scheduled by the HST with the highest prefetching priority. Two cases are illustrated as an example in Fig. 4:

- **Case-1:** Given that only 2 UEs are allowed for prefetching by meeting the policy of "*Whether to prefetch*", with data rate $\frac{B_{cur}}{2}$ shared by each UE, the segment S2 for UE2 (green color), and segment S2 for UE3 (yellow color) are to be prefetched. This is because these two UEs either have not buffered any segment, or the HST has not been prefetching segments for them.
- **Case-2:** If the segments to be prefetched have already been cached by the HST, those cached segments will be excluded for segments to consume, e.g., segments S2 (white color) for UE3 is excluded but directly buffered from caching space. Besides, if two UEs buffer the same number of segments in devices, the one with the least number of segments to be prefetched is with a higher priority (e.g., although UE1 and UE3 both request the same segment, the former is scheduled due to an earlier request time). Here, the segment S4 (green color) and segment S3 (white color) are scheduled for prefetching.

C. Proposed Caching Strategy

Two operations are involved, namely the caching decision (whether to cache a segment), and replacement decision (which segment to replace) when the caching space exhausts.

1) Whether to cache: We here apply a generic policy, that the HST will cache any consumed segment by UEs. Note that, the HST may refuse to cache a consumed segment if it wont be popular in future, where certain technologies on prediction of content popularity could be integrated for this technical purpose.

2) Which to replace: In case of caching space exhaustion, the segment with the lowest popularity is replaced, with the one has been recently consumed. Besides, the one with an earliest caching time is replaced most likely, just in case two compared segments are with the same popularly. This can be referred to Fig. 4:

- **Case-1:** The segment S1 for UE1 (white color) is replaced, with the segment S3 (white color). This is because the former is with the same popularity $(P = 1)$, but the request of S1 arrived prior to others.
- **Case-2:** If the HST has cached those segments being requested by existing UEs, these segments in the caching space will be labelled. This however triggers a protection mechanism, where the labelled segments will not be replaced (even with the lowest popularity), until they have been consumed.

Since the segment S2 (white color) is labelled, S1 and S2 (both with green color) in caching space are replaced with S3 (white and green color).

IV. Case Study

We deliberatively disable the adaptation function in order for the users to receive the stream with a guaranteed video bitrate for the sake of experiences. As such, we set the fixed video bit rate at 1.5Mbps which can correspond to a HD level video quality. As can be inferred, such a scenario without adaptation technically increases the degree of challenging on the problem formulation of prefetching video segment for the sake of assured video quality. In our previous works [8], we follow exactly the same strategy for assuring 4K level of video quality without adaptation.

Following the wireless channel discussed in Section III-A with 300 km/h HST speed, the evaluation is implemented by the Opportunistic Network Environment (ONE). The entire simulation time is 18000s, the UE requests are generated per 85s, with a rate of 1.5 Mbit/s video streaming between $[0 \sim 16780]$s. Each video content consists of 122 segments, with 10s per segment length, the HST WiFi is with 1000 Mbit/s. These settings partially refer to our previous study [7] in fix scenario. The simulation resolution is 1s, during which prefetching, caching and calculation involved for wireless channel are calculated.

For fair comparison to the proposed MEC based system, we also evaluate the **Benchmark**, in which UEs fetch perceived segments from associated BSs directly, with HST only relays requests to BSs and actual segments from them to UEs, but not behave as a proxy to enable prefetch segments and caching. Referring to Fig. 1, therein UEs will interact with BS directly (just through on-board WiFi, as without MEC server positioned).

- **Average First Access Delay (AFAD):** It is given as the average delay per UE has initially buffered the first segment in device.
- **Average Disruption Frequency (ADF):** The average disruption frequency per UE to consume the perceived length of video.
- **Average Disruption Duration (ADD):** The average disruption period per UE to consume the perceived length of video.
- **Average Perceived Length (APL):** Per UE consumed certain length of perceived video (the length depends on how many segments have been consumed).
- **Fully Served Ratio (FSR):** The ratio between the number of UEs fully consume their perceived video content, over total number of UEs.

Among above metrics, AFAD (how long it takes the UE to start playback upon initial request), ADF & ADD (frequency and duration of video playback disruptions in terms video stalling) and APL (how long has the video been played back since the beginning) reflect service QoE of on-board users.

A. *Major Benefit of Prefetching*

We set a sufficiently large HST storage can prefetch 122$\times$209 segments, to examine the fundamental feature of prefetching herein. Results are given in TABLE I, Prefetching outperforms Benchmark under the default settings, with alleviated ADD and ADF, as well as faster AFAD. This certainly follows Equation (1) on how prefetch improves QoE. If varying the UE video rate R, we observe the more UEs compete B_{cur} for prefetching, the worse UEs QoE will be (mainly due to not timely buffering segments at devices size).

Additionally, in case of shorter video streaming (by reducing per segment length C), we observe the benefit of prefetching becomes subtle, due to the less influence of K in Equation (1). This is mainly because more number of segments (due to shorter C) can be buffered at device side (which already helps to alleviate disruption).

We further vary the moving speed of HST. Noticeably, the QoE becomes degraded with 500 km/h case, with disruption happens. This is because by referring in Fig. 3, the width of throughput would be even insufficient to prefetch segments.

TABLE I
EVALUATION RESULTS OF CASE STUDY

	APL	ADD	ADF	AFAD	FSR
—Default Setting—					
Prefetching	122	0s	0	0.96s	100%
Benchmark	122	0.44s	0.39	6.32s	100%
—Influence of UEs Request Rate—					
60s					
Prefetching	96.43	3074.15s	66.60	12.60s	61.64%
Prefetching&Caching	122	0.37s	0.22	0.34s	100%
Benchmark	94.39	4063.81s	89.94	48.77s	53.76%
100s					
Prefetching	122	0s	0	0.63s	100%
Benchmark	122	0s	0	3.18s	100%
—Influence of Video Streaming Length—					
122×5s					
Prefetching	122	0s	0	0.35s	100%
Benchmark	122	0s	0	0.70s	100%
122×2s					
Prefetching	122	0s	0	0.14s	100%
Benchmark	122	0s	0	0.15s	100%
—Influence of HST Speed—					
100 km/h					
Prefetching	122	0s	0	0.98s	100%
Benchmark	122	1.54s	1.35	6.64s	100%
500 km/h					
Prefetching	122	2.97s	1.75	1.65s	100%
Benchmark	121.99	48.77s	33.35	10.22s	99.49%
—Influence of UE Perceived Rate, Video Streaming Length—					
2 Mbit/s, 92×10s					
Prefetching	92	0s	0	1.28s	100%
Benchmark	92	0.52s	0.45	6.36s	100%
4 Mbit/s, 46×10s					
Prefetching	46	0s	0	2.47s	100%
Benchmark	46	1.43s	0.72	8.16s	100%
14 Mbit/s, 13×10s					
Prefetching	13	0s	0	6.90s	100%
Benchmark	13	9.35s	3.34	11.38s	100%

We next share our lesson about the provisioning of perceived length in line with corresponding UE streaming rate R. Here, a shorter video length should be limited for higher R. This follows a basic theory that, as the total data capacity under a certain HST speed is fixed (as the dedicated throughput is derived from Equation (2)), the volume of video streaming data from UEs should not exceed that for disruption free.

B. Complementary Benefit of Caching

As fundamental study on prefetching is focused in this article, we simply set 50% of total HST storage space for caching operation (LFU based) that set above, and maintain the rest for prefetching. UEs generate request from 1000 unique content, of which their requests follow Zipf distribution with a skew value of 0.9. We focus on the 60s per request rate case, where enabling caching substantially improves QoE, as compared to the case with Prefetching enabled only. In other situation where user requests rate is infrequent than 60s, the benefits of caching is not obvious, since the storage is large enough to process video frame.

V. Future Directions

A. Storage Management

When interworking with prefetching, we envisage additional complexity in content management. First of all, prefetched but not yet consumed segments (i.e., the segments already arrived at MEC server but not yet delivered to the requesting UE) should be exempted from eviction operations, as they are highly likely to be requested from the UE in the near future. Given that prefetching and caching share common storage space, storage management concerning the coordination between prefetching and caching operations should be based on the specific content popularity characteristics. For instance, in case individual UEs request different video content without exhibiting common interests, prefetching needs to be prioritized in terms of storage space allocation, since caching is not beneficial for diverse content request patterns.

B. Advances on Prefetching and Caching

Fruitful UEs profile should be taken into account for prefetching, e.g., the actual journey on where to get-off HST requires additional effort on prefetching strategy. Besides, the prediction on video content popularity certainly benefits to the improvement from smart caching, particularly for a highly dense UEs requests case (e.g., the 60s requests rate in our case study). Of course, future test-bed demonstration [8] will be plus to further investigate the impact of practical environmental issue of wireless channel on the actual QoE performance.

C. Security and Privacy

The encryption and privacy will be important, where if the videos are of value they will most likely be end-to-end encrypted such that caching for different users is simply impossible. While as far as Information Centric Networking (ICN) architectures have been positioned in content delivery network, most multimedia data would be transported to the network edge.

VI. Conclusion

In this paper, we proposed a MEC based video content prefetching and caching framework, to serve on-board UEs with satisfied QoE over the HST use case. The advantage of prefetching and caching have been investigated, through analysis and a case study. Furthermore, multiple dimensional factors have been evaluated, so as to guide the provisioning on variety of video streaming service (in terms of video resolution) over HST. Results show the fundamental benefit of prefetching function on fast access, meanwhile with service QoE guaranteed.

References

[1] Y. Dong, P. Fan, and K. B. Letaief, "High-speed railway wireless communications: Efficiency versus fairness," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 925–930, Feb 2014.

[2] "23009-1:2014 dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats," 2010. [Online]. Available: https://www.iso.org/standard/65274.html

[3] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over http," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1842–1866, thirdquarter 2017.

[4] K. Kanai, T. Muto, J. Katto, S. Yamamura, T. Furutono, T. Saito, H. Mikami, K. Kusachi, T. Tsuda, W. Kameyama, Y. J. Park, and T. Sato, "Proactive content caching for mobile video utilizing transportation systems and evaluation through field experiments," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2102–2114, Aug 2016.

[5] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. of the Sixth International Conference on Advances in Future Internet*. Citeseer, 2014.

[6] 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.

[7] C. Ge, N. Wang, S. Skillman, G. Foster, and Y. Cao, "QoE-Driven DASH Video Caching and Adaptation at 5G Mobile Edge," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ser. ACM-ICN '16, Kyoto, Japan, 2016.

[8] C. Ge, N. Wang, W. K. Chai, and H. Hellwagner, "Qoe-assured 4k http live streaming via transient segment holding at mobile edge," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1816–1830, Aug 2018.

[9] Y. Ma and R. Tafazolli, "Channel estimation for ofdma uplink: a hybrid of linear and bem interpolation approach," *IEEE Transactions on Signal Processing*, vol. 55, no. 4, pp. 1568–1573, April 2007.

[10] G. T. 38.913, "Requirements for further advancements for evolved universal terrestrial radio access (e-utra) (lte-advanced)," Tech. Rep., 2016.

[11] L. Zhang, A. Ijaz, P. Xiao, and R. Tafazoli, "Multi-service systems: An enabler of flexible 5g air-interface," *IEEE Communications Magazine*, Accepted in 2017.

[12] L. Liu, C. Tao, J. Qiu, H. Chen, L. Yu, W. Dong, and Y. Yuan, "Position-based modeling for wireless channel on high-speed railway under a viaduct at 2.35 ghz," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 834–845, 2012.

[13] G. T. 36.942, "Evolved universal terrestrial radio access (e-utra): Radio frequency (rf) system scenarios," Tech. Rep., 2010.

[14] P. Mogensen, W. Na, I. Z. Kovács, F. Frederiksen, A. Pokhariyal, K. I. Pedersen, T. Kolding, K. Hugl, and M. Kuusela, "LTE capacity compared to the Shannon boun," in *IEEE VTC' 07*, 2007.