Media Operations Use Case for an Augmented Reality Application on Edge Computing Infrastructure
draft-krishna-mops-ar-use-case-02

Abstract

A use case describing transmission of an application on the Internet that has several unique characteristics of Augmented Reality (AR) applications is presented for the consideration of the Media Operations (MOPS) Working Group. One key requirement identified is that the Adaptive-Bit-Rate (ABR) algorithms’ current usage of policies based on heuristics and models is inadequate for AR applications running on the Edge Computing infrastructure.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 21, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
1. Introduction

The MOPS draft, [I-D.ietf-mops-streaming-opcons], provides an overview of operational networking issues that pertain to Quality of Experience (QoE) in delivery of video and other high-bitrate media over the Internet. However, as it does not cover the increasingly large number of applications with Augmented Reality (AR) characteristics and their requirements on ABR algorithms, the discussion in this draft compliments the overview presented in that draft [I-D.ietf-mops-streaming-opcons].

Future AR applications will bring several requirements for the Internet and the mobile devices running these applications. AR applications require a real-time processing of video streams to recognize specific objects. This is then used to overlay information on the video being displayed to the user. In addition some AR applications will also require generation of new video frames to be played to the user. Both the real-time processing of video streams and the generation of overlay information are computationally intensive tasks that generate heat [DEV_HEAT_1], [DEV_HEAT_2] and drain battery power [BATT_DRAIN] on the AR mobile device. Consequently, in order to run future applications with AR characteristics on mobile devices, computationally intensive tasks need to be offloaded to resources provided by Edge Computing.

Edge Computing is an emerging paradigm where computing resources and storage are made available in close network proximity at the edge of the Internet to mobile devices and sensors [EDGE_1], [EDGE_2].

Adaptive-Bit-Rate (ABR) algorithms currently base their policy for bit-rate selection on heuristics or models of the deployment environment that do not account for the environment’s dynamic nature.
in use cases such as the one we present in this document. Consequently, the ABR algorithms perform sub-optimally in such deployments [ABR_1].

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Use Case

We now describe a use case that involves an application with AR systems’ characteristics. Consider a group of tourists who are being conducted in a tour around the historical site of the Tower of London. As they move around the site and within the historical buildings, they can watch and listen to historical scenes in 3D that are generated by the AR application and then overlaid by their AR headsets onto their real-world view. The headset then continuously updates their view as they move around.

The AR application first processes the scene that the walking tourist is watching in real-time and identifies objects that will be targeted for overlay of high resolution videos. It then generates high resolution 3D images of historical scenes related to the perspective of the tourist in real-time. These generated video images are then overlaid on the view of the real-world as seen by the tourist.

We now discuss this processing of scenes and generation of high resolution images in greater detail.

3.1. Processing of Scenes

The AR application that runs on the mobile device needs to first track the pose (coordinates and orientation) of the user’s head, eyes and the objects that are in view. This requires tracking natural features and developing an annotated point cloud based model that is then stored in a database. To ensure that this database can be scaled up, techniques such as combining a client-side simultaneous tracking and mapping and a server-side localization are used [SLAM_1], [SLAM_2], [SLAM_3], [SLAM_4]. Once the natural features are tracked, virtual objects are geometrically aligned with those features. This is followed by resolving occlusion that can occur between virtual and the real objects [OCCL_1], [OCCL_2].

The next step for the AR application is to apply photometric registration [PHOTO_REG]. This requires aligning the brightness and color between the virtual and real objects. Additionally, algorithms
that calculate global illumination of both the virtual and real objects \[GLB\_ILLUM\_1\], \[GLB\_ILLUM\_2\] are executed. Various algorithms to deal with artifacts generated by lens distortion \[LENS\_DIST\], blur \[BLUR\], noise \[NOISE\] etc are also required.

3.2. Generation of Images

The AR application must generate a high-quality video that has the properties described in the previous step and overlay the video on the AR device’s display - a step called situated visualization. This entails dealing with registration errors that may arise, ensuring that there is no visual interference \[VIS\_INTERFERE\], and finally maintaining temporal coherence by adapting to the movement of user’s eyes and head.

4. Requirements

The components of AR applications perform tasks such as real-time generation and processing of high-quality video content that are computationally intensive. As a result, on AR devices such as AR glasses excessive heat is generated by the chip-sets that are involved in the computation \[DEV\_HEAT\_1\], \[DEV\_HEAT\_2\]. Additionally, the battery on such devices discharges quickly when running such applications \[BATT\_DRAIN\].

A solution to the heat dissipation and battery drainage problem is to offload the processing and video generation tasks to the remote cloud. However, running such tasks on the cloud is not feasible as the end-to-end delays must be within the order of a few milliseconds. Additionally, such applications require high bandwidth and low jitter to provide a high QoE to the user. In order to achieve such hard timing constraints, computationally intensive tasks can be offloaded to Edge devices.

Note that the Edge device providing the computation and storage is itself limited in such resources compared to the Cloud. So, for example, a sudden surge in demand from a large group of tourists can overwhelm that device. This will result in a degraded user experience as their AR device experiences delays in receiving the video frames. In order to deal with this problem, the client AR applications will need to use Adaptive Bit Rate (ABR) algorithms that choose bit-rates policies tailored in a fine-grained manner to the resource demands and playback the videos with appropriate QoE metrics as the user moves around with the group of tourists.

However, heavy-tailed nature of several operational parameters make prediction-based adaptation by ABR algorithms sub-optimal \[ABR\_2\]. This is because with such distributions, law of large numbers works...
too slowly, the mean of sample does not equal the mean of
distribution, and as a result standard deviation and variance are
unsuitable as metrics for such operational parameters [HEAVY_TAIL_1],
[HEAVY_TAIL_2]. Other subtle issues with these distributions include
the "expectation paradox" [HEAVY_TAIL_1] where the longer we have
waited for an event the longer we have to wait and the issue of
mismatch between the size and count of events [HEAVY_TAIL_1]. This
makes designing an algorithm for adaptation error-prone and
challenging. Such operational parameters include but are not limited
to buffer occupancy, throughput, client-server latency, and variable
transmission times. In addition, edge devices and communication links
may fail and logical communication relationships between various
software components change frequently as the user moves around with
their AR device [UBICOMP].

Thus, once the offloaded computationally intensive processing is
completed on the Edge Computing, the video is streamed to the user
with the help of an ABR algorithm which needs to meet the following
requirements [ABR_1]:

- Dynamically changing ABR parameters: The ABR algorithm must be
  able to dynamically change parameters given the heavy-tailed
  nature of network throughput. This, for example, may be
  accomplished by AI/ML processing on the Edge Computing on a per
  client or global basis.

- Handling conflicting QoE requirements: QoE goals often require
  high bit-rates, and low frequency of buffer refills. However in
  practice, this can lead to a conflict between those goals. For
  example, increasing the bit-rate might result in the need to fill
  up the buffer more frequently as the buffer capacity might be
  limited on the AR device. The ABR algorithm must be able to
  handle this situation.

- Handling side effects of deciding a specific bit rate: For
  example, selecting a bit rate of a particular value might result
  in the ABR algorithm not changing to a different rate so as to
  ensure a non-fluctuating bit-rate and the resultant smoothness of
  video quality. The ABR algorithm must be able to handle this
  situation.

5. Informative References

[ABR_1] Mao, H., Netravali, R., and M. Alizadeh, "Neural Adaptive
Video Streaming with Pensieve", In Proceedings of the
Conference of the ACM Special Interest Group on Data


Authors’ Addresses

Renan Krishna
InterDigital Europe Limited
64, Great Eastern Street
London EC2A 3QR
United Kingdom

Email: renan.krishna@interdigital.com