Evaluating Multimedia Protocols on 5G Edge for Mobile Augmented Reality

Jacky Cao†, Xiang Su‡†, Benjamin Finley*, Antti Pauanne†, Mostafa Ammar§, Pan Hui*||
† University of Oulu, Oulu, Finland
‡ Norwegian University of Science and Technology, Gjøvik, Norway
* University of Helsinki, Helsinki, Finland
§ Georgia Institute of Technology, Atlanta, USA
|| Hong Kong University of Science and Technology, Hong Kong

***Abstract*—Mobile Augmented Reality (MAR) mixes physical environments with user-interactive virtual annotations. Immersive MAR experiences are supported by computation-intensive tasks, which are typically offloaded to cloud or edge servers. Such offloading introduces additional network traffic and influences the motion-to-photon latency (a determinant of user-perceived quality of experience). Therefore, proper multimedia protocols are crucial to minimise transmission latency and ensure sufficient throughput to support MAR performance. Relatedly, 5G is a potential MAR supporting technology and is widely believed to be faster and more efficient than its predecessors. However, the suitability and performance of existing multimedia protocols for MAR in the 5G edge context have not been explored. In this work, we present a detailed evaluation of several popular multimedia protocols (HLS, MPEG-DASH, RTP, RTMP, RTMFP, and RTSP) and transport protocols (QUIC, UDP, and TCP) with a MAR system on a real-world 5G edge testbed. The evaluation results indicate that RTMP has the lowest median client-to-server packet latency on 5G and LTE for all image resolutions. In terms of individual image resolutions, from 144p to 480p over 5G and LTE, RTMP has the lowest median packet latency of 14.03±1.05 ms. Whereas for jitter, HLS has the smallest median jitter across all image resolutions over LTE and 5G with medians of 2.62 ms and 1.41 ms, respectively. Our experimental results indicate that RTMP and HLS are the most suitable protocols for MAR.**

***Index Terms*—multimedia protocols, mobile augmented reality, 5G, edge computing**

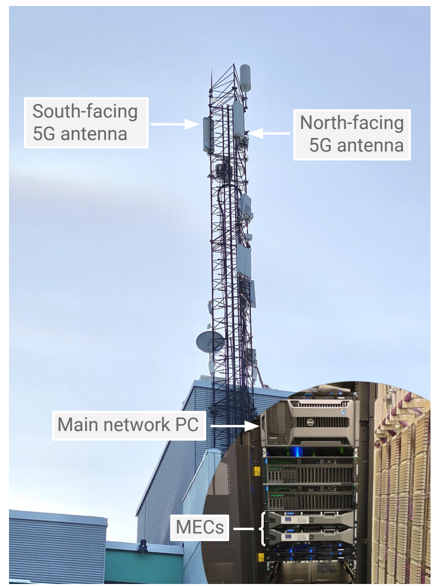

Fig. 1: A view of the 5G Test Network used as our testbed which contains, amongst other components, a cellular tower, 5G and LTE basestations, and MECs.

I. Introduction

Mobile augmented reality (MAR) supplies users with additional information and an enhanced perception of their surroundings through superimposed virtual augmentations on real-time camera feeds [1], [2]. Thus camera images are the main data source for MAR applications, and processing these frames requires significant computation which impacts device usability and battery life. Computational offloading addresses this challenge by leveraging the computation and storage capabilities of external servers. Both cloud and edge offloading-based MAR systems have been proposed to improve performance and minimise the overall delay. While cloud computing provides centralised powerful remote resources, edge computing places resources closer to end users [3] thus lowering motion-to-photon latencies [4] and presenting fewer bottlenecks [5]. Furthermore, edge computing enables better security and privacy protection by, for example, limiting the effect of distributed denial of service attacks and allowing for location-based authentication [6]. To leverage the benefits of edge computing, suitable communication protocols for multimedia transfer in MAR systems are crucial.

However, no existing multimedia protocol is designed and optimised for MAR. An ideal protocol is envisioned to 1) handle transmission of various data types (with different priorities) under diverse network conditions (e.g., diverse latency, jitter, bandwidth, and loss); 2) achieve a good balance between fairness (between users) and overall exploitation of network capacity; and 3) provides low latency and high fault tolerance to allow real-time communications. In further elaboration of the general requirements, the specific Quality of Experience (QoE) of MAR depends highly on the round trip latency [7]. For example, when users explore augmented environments, they expect results relevant to current viewable objects or surroundings. However, an unstable network connection may introduce additional data transfer latency and thus the results may be obsolete, leading to degraded experiences. To maintain a high QoE, the complete round trip latency should ideally be lower than the camera capture frame rate, typically 30 FPS or approximately 33 ms [8]. Related studies in Virtual Reality (VR) have also shown that round trip latencies of 50 ms or less maximise QoE [9].

Given the above-mentioned considerations, more research is required to evaluate the suitability of existing protocols in MAR, especially in the context of the 5G edge. This work focuses on evaluating diverse multimedia and transport protocols in a typical MAR system on a 5G testbed. This setup allows for controlled experiments to fully understand different performance trade-offs. The network used is the 5G Test Network (5GTN) [10] in Oulu, Finland (Fig. 1) which supports several wireless technologies, which provide connectivity access (5G, LTE, WiFi, IoT networks, etc.), Mobile Edge Computing (MEC) servers, a cloud core and multiple Evolved Packet Cores, and the ability to connect to the wider public Internet through a VPN. We present 5G evaluation results along with LTE and WiFi results as two comparative baselines. We evaluate a combination of selected application-layer multimedia protocols (HLS, MPEG-DASH, RTP, RTMP, RTMFP, RTSP) and transport-layer protocols (QUIC, UDP, TCP), and compare key network QoS metrics, including the client-to-server latency, jitter, and throughput.

Our 5G evaluation results indicate that RTMP and HLS are the most suitable protocols for MAR applications on 5G. While RTMP has the lowest median packet latency across all tested image resolutions of 19.35±8.63 ms, the protocol which produces the lowest packet latency for each resolution is not the same. Namely, from 144p to 480p, RTMP provides the lowest latency with 14.03±1.05 ms, HLS with 14.80±1.07 ms for 720p to 2K, and QUIC with 18.52±2.95 ms for 4K. The latency results with LTE connections also show that RTMP offers the lowest median packet latency of 18.69±5.61 ms amongst the different tested protocols and resolutions. Measured jitter for both 5G and LTE connections demonstrate that HLS has the least jitter with median values of 2.62 ms and 1.41 ms, respectively. Therefore both RTMP and HLS could be considered as suitable protocols to be used in MAR applications for multimedia transferal.

Our work is one of the first efforts to evaluate multimedia protocols for MAR on the 5G edge. Our contributions are:

- *A comparison of a diverse selection of multimedia protocols in a typical MAR system.* We provide a quantitative performance evaluation of protocol combinations, including multimedia protocols (HLS, MPEG-DASH, RTP, RTMP, RTMFP, and RTSP) and transport protocols (QUIC, UDP, and TCP).
- *An evaluation with a real-world 5G testbed.* We conduct our evaluation with 5GTN testbed (including 5G base station, edge server, etc) and compare the performance of 5G, LTE, and WiFi connections.
- *A comprehensive performance analysis through various QoS metrics.*

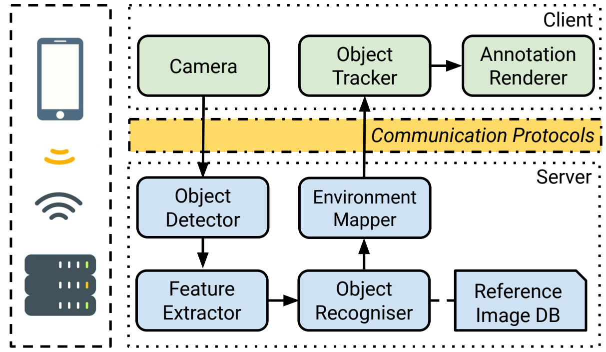

Fig. 2: A typical pipeline of the MAR systems.

The remainder of the work is organised as follows. Section II introduces 5G edge, MAR, and QoS metrics. Section III presents an overview of the selected protocols and related work in the area of multimedia protocol analysis. Section IV details the experimentation setup and presents the collected network measurement results and analytics. Section V discusses key insights, the potential impact of other variables, and limitations; while Section VI concludes the work.

II. MAR and 5G Edge

To facilitate high-quality MAR user experiences, 5G edge computing can be employed [11]. 5G is the fifth generation standard for wireless communication. 5G paradigm shift enables several key features and improvements over 4G, including ultra-reliable low-latency communications, enhanced mobile broadband for increased data transmission rates, and massive machine-type communications to support extreme device densities supported by high-density base station and antenna deployment. Several 5G features significantly benefit MAR as extensive computation offloading requires low latency and high bandwidth, both of which are supported by 5G. Additionally, the reduced round trip latency introduces benefits, such as lower device energy consumption [12], which can improve the MAR experience for users.

A. MAR System on 5G Testbed

Our MAR system uses an Android client application that captures images and generates object recognition requests, and a server application that performs object recognition and sends results to the client. The system follows a general MAR pipeline (Fig. 2): 1) the client device camera captures images at 30 FPS; 2) the images are sent to the server at the same rate and the server performs object detection; 3) for detected objects, their features are extracted; 4) extracted feature points are analysed by the object recogniser and retrieves object information from a reference image database; 5) for recognised objects, the environment mapper calculates object bounding boxes; 6) the box vertices are sent to the client object tracker; and 7) the annotation renderer draws the virtual information on the client display. Our developed MAR system is considered

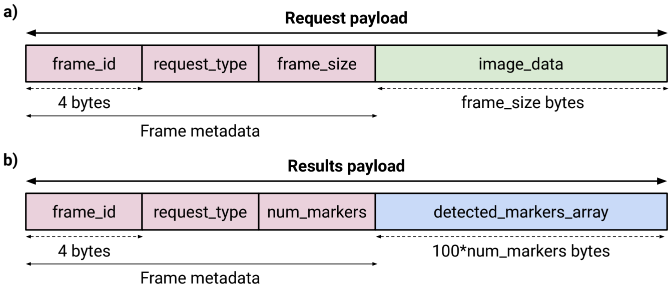

Fig. 3: Format of data arrays send from a) client to server and b) server to client.

typical because it follows the architecture and system design of several existing client-edge MAR implementations [8], [13].

In our MAR system, data is transferred between client and server during two different steps: 1) the client sends object recognition requests to the server, and 2) the server replies with the request results. For both transfers, the data is structured and encoded into a byte array and passed to the network protocols. Fig. 3 breaks down the format and information stored in the request and result arrays. Both arrays contain initial headers which include information for client and server logic, e.g., the message request type determining if the request is for object recognition or registering the client to the server.

B. QoS Metrics for MAR

The data transfer performance between client and server can be quantified using network QoS metrics. These metrics can be used (in conjunction with QoE models and other metrics) to optimise the actual user experience. For example, minimising network latency lowers the round trip system latency, which impacts the user experienced responsiveness in applications, such as AR and VR [9]. For MAR, in particular, we focus on three key QoS metrics, i.e. latency, jitter, and throughput (as defined below). We measure these metrics for the channel link from client to server; we emphasise this direction as the heaviest communication load arises from multimedia transfer.

1) **Latency**, or client-to-server latency is the delay of packets sent from the client to the server. This latency does not include the server inference time as the focus of this work is on the multimedia protocols.
2) **Jitter** is the variation in the latencies of packets from the client to server (see RFC 3393 [14]). Packets are sent continuously from the client. However, network congestion or queuing issues can lead to varying latencies between packets and thus higher jitter.
3) **Throughput** is the load of the communication link, calculated by measuring the amount of data transferred within a specific period from client to server.

III. Multimedia and Transport Protocols for MAR

Transferring multimedia data from client to server for MAR can be accomplished over application layer multimedia protocols or simply transport layer protocols. In our evaluation, we study a selection of application layer protocols (HLS, MPEG-DASH, RTP, RTMP, RTMFP, RTSP) and transport layer protocols (QUIC, UDP, and TCP). Fig. 4 represents the placement of each of the protocols within the protocol stack, with QUIC straddling the line between application and transport layers as QUIC can be classed as either of the two.

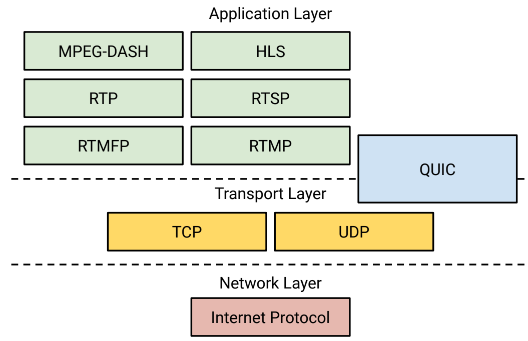

Fig. 4: Multimedia and transport protocols within the protocol stack. QUIC can be referred as both an application and transport layer protocol.

A. Overview of Selected Protocols in the Evaluation

HTTP Live Streaming (HLS) HLS [15] is for delivering continuous and long-form video over the Internet. The receiver can adapt the media bit rate to current network conditions to maintain uninterrupted playback at the best quality.

Dynamic Adaptive Streaming over HTTP (MPEG-DASH) MPEG-DASH [16] is for dynamic adaptive streaming to deliver MPEG media over HTTP. Streams are broken down into small segments and sent to receiving devices to be replayed as a continuous stream.

Real-Time Transport Protocol (RTP) RTP [17] provides network transfer for real-time data, including video and audio, typically over UDP. RTP does not manage resource reservation or guarantee real-time QoS, but a feedback mechanism is used to control RTP streams, the real-time control protocol (RTCP).

Real-Time Messaging Protocol (RTMP) RTMP [18] uses parallel reliable streams (TCP) to carry video, audio, and data between peers simultaneously. RTMP has several variations, RTMPS tunnels data over TLS/SSL, RTMPE uses Adobe encryption, RTMPT uses HTTP to transport data, and RTMFP (see next) uses UDP for data transfer.

Real-Time Media Flow Protocol (RTMFP) RTMFP [19] is a general-purpose endpoint-to-endpoint protocol, which supports real-time data transport of video, audio, and data. The underlying transport protocol is UDP and supports independent parallel message flows, variable message reliability, multi-point congestion control, and security mechanisms.

Real Time Streaming Protocol (RTSP) RTSP [20] sets up and controls data delivery for media streaming, e.g., on-demand video or live audio streaming. Requested media can consist of multiple audio and video streams that are delivered as time-synchronised streams from servers to clients.

Quick UDP Internet Connections (QUIC) QUIC [21] is a connection-orientated transport protocol for low latency and secure data transport over UDP. Packets are authenticated and encrypted and use TLS security measures. Multiplexed connections are established between endpoints to transfer data to maintain high QoS in multimedia and web applications, as well as support secure and fast data transmission.

User Datagram Protocol (UDP) UDP is a general transport protocol that uses datagram packets to communicate between network hosts. There is no reliable delivery mechanism, so packets are not guaranteed to reach their destination, nor in the order of transmission. UDP supports multicast and broadcast from a single host to multiple receivers.

Transmission Control Protocol (TCP) TCP is a transport protocol for reliable data transfer, flow and congestion control, and direct point-to-point connections. Connections form using a handshake mechanism and are maintained with acknowledgement packets. The protocol has a historical data buffer, ensuring no data is lost between hosts.

B. Comparison of Multimedia Protocols

There are limited prior works exploring multimedia protocols performance in MAR scenarios. Therefore, the most related works are comparisons of multimedia and general protocols in the context of other networks and applications.

Nuñez and Toasa [22] compare RTMP, RTSP, and HLS multimedia protocols in terms of network stability, mobile device battery life, and CPU and RAM resource consumption. The comparison includes protocol evaluations while traversing LTE, 3G, and 2G cellular connections. Although, the work is limited to RTMP, RTSP, and HLS multimedia protocols, and does not compare other multimedia protocols, nor any general transport protocols. While [23] studies RTMP, HLS, and RTSP over UDP/TCP network performances. This work presents the received client frame rate when network metrics, such as packet loss rate and bandwidth are varied. Compared with our work, network performance is an experimental variable and not directly measured, furthermore they use only one image resolution. Aloman et al. [24] evaluate MPEG-DASH, RTSP, and RTMP over 4G and WiFi in terms of user QoE. In comparison to our study, their primary focus is the effect on QoE and not QoS. A comparison of RTP, HTTP, HLS, MPEG-DASH performances with cloud-assisted adaptive streaming is presented in [25]. The work considers protocol performance when transmitting data to cloud servers. Comparatively, we utilise an edge server for MAR and extensively evaluate different protocols with three network connection types.

For general transport protocol evaluation in traditional IP networks, [26] evaluates SCTP for signalling transport and compares the performance with UDP and TCP implementations. The work compares network performance metrics for UDP and TCP transport protocols with a simulated network. Furthermore, [27] compares the latencies of QUIC, SDPY, and HTTP while modifying bandwidth and packet loss, they conclude that network conditions determine which protocol performs best at a given time.

TABLE I: Summary of evaluation parameters

Parameter	Values
Protocol	HLS, MPEG-DASH, RTP (UDP/TCP), RTMP, RTMFP, RTSP (UDP/TCP), QUIC, UDP, TCP
Media resolution	144p, 230p, 360p, 480p, 720p, 1080p, 2K, 4K
Multimedia rate	30 FPS
Network type	5G, LTE, WiFi
Individual eval. time	10 minutes

Finally, as a promising new protocol, QUIC has been extensively evaluated. Megyesi et al. [27] evaluate QUIC, SDP, and HTTP based on metrics such as page load time and packet loss. The study concludes that the best protocol depends on network conditions. Bishop and Akamai [28] perform a similar comparison with HTTP/2 versus QUIC+SPDY 3.1 while utilising 2G and LTE and concludes that with 2G 90% of pages load faster with QUIC while with LTE 60% of pages load faster with QUIC. Finally, Soni and Rajput [29] compare the protocol security and network performance of QUIC against TCP and UDP, and find that QUIC maintains low data retrieval times and high throughputs as data transmission size increases.

IV. Experimentation and Analytics

A. Experimentation Setup and Evaluation Metrics

The MAR client application runs on a OnePlus 7 Pro 5G smartphone. The server runs on an edge PC with an Intel Core i7-9750H CPU, 32 GB memory, and an NVIDIA GeForce RTX 2080 Max-Q GPU and is connected to the 5GTN testbed through Ethernet. The 5GTN testbed contains a base transceiver station (BTS) and two antennas operating in the C-Band spectrum for 5G at 3500 MHz with a bandwidth of 60 MHz. Additionally, LTE is provided by an LTE Picocell at 2600 MHz with a bandwidth of 10 MHz. The distance from the outdoor 5G BTS to the indoor smartphone is ≈ 30 m with no direct line of sight (NLOS), while the distance from the indoor LTE Picocell is ≈ 10 m also with NLOS. Finally, the indoor edge PC generates a 5 GHz 802.11ac WiFi hotspot and the distance from the PC is approximately one meter with a line of sight (LOS). We argue that these are typical conditions, e.g., mobile network access points (APs) are less often in direct LOS with clients, especially indoors. However, we plan to expand the variety of conditions analysed in future research.

The client transmits a device-captured raw camera stream to the server using the multimedia protocols without pre-processing (such as resolution downscaling). The protocol connections are initiated once between the client and server. New sessions and connections are only created after each test is completed. Server responses containing results, a combination of object noun strings and arrays containing the vertices of object bounding boxes, are joined and converted to a byte array and sent to the client through a separate UDP connection.

We evaluate a total of eight multimedia and three transport protocols using our MAR system over 5G, LTE, and WiFi. We utilise the FFmpeg library[1] for the implementations of eight

[1]https://ffmpeg.org/

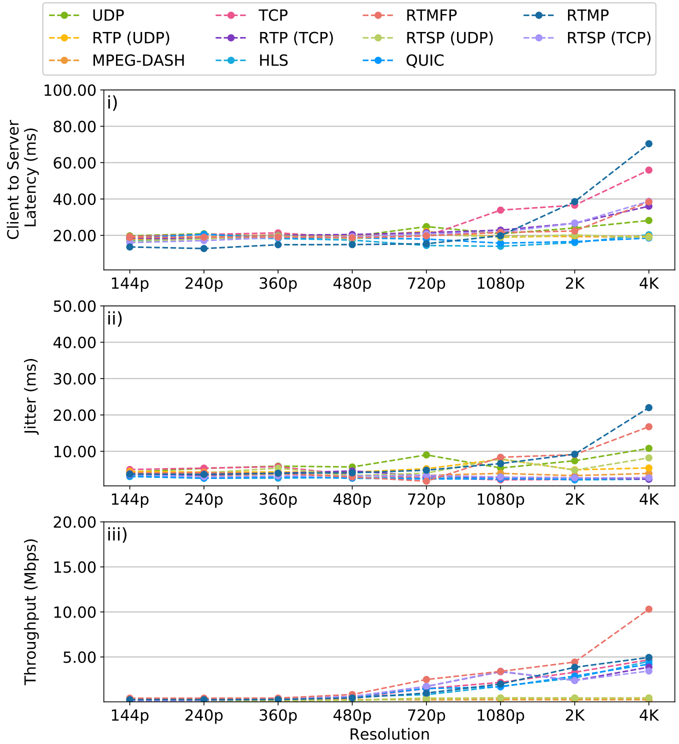

Fig. 5: Evaluation results for different multimedia and transport protocols, and resolution sizes with 5G testbed.

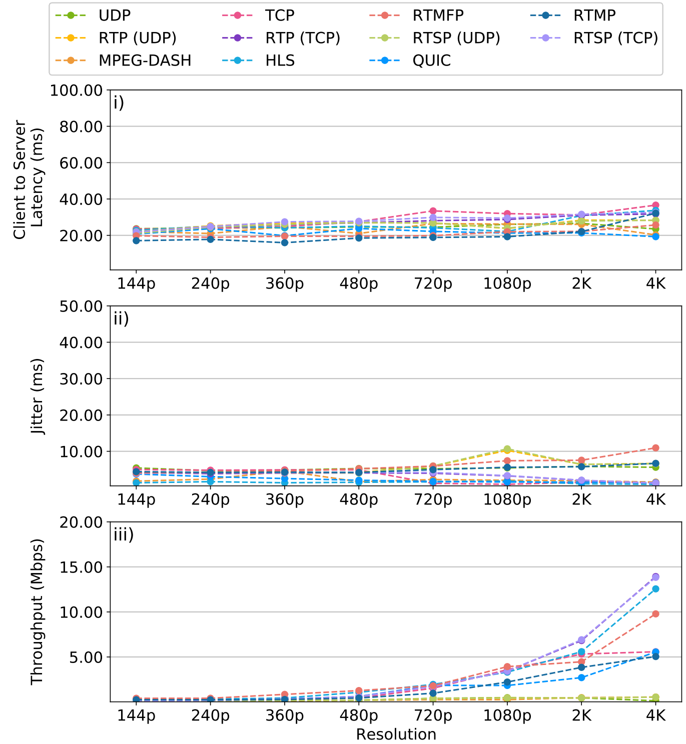

Fig. 6: Evaluation results for different multimedia and transport protocols, and resolution sizes while connected to LTE.

protocols, including HLS, MPEG-DASH, RTP, RTMP, RTSP, UDP, and TCP. For RTSP and RTP, we specify the underlying transport protocol (i.e., UDP or TCP), we will differentiate these four different types by specifying the transport protocol in parenthesis, for example, RTP (UDP) or RTSP (TCP). We use separate libraries for RTMP[2], and QUIC[3] implementations. We include transport protocols (UDP, TCP, QUIC) to provide comparisons for the multimedia protocols, as all of them use these protocols as an underlying backbone for data transport.

Table I presents the parameters used in the evaluation. They define 264 separate test scenarios. When collecting data, we configure the client application to make requests for a ten minute testing period as this allows for a good indication of the protocol performance under active network conditions while averaging out noise or momentary changes in network quality.

We maintain a 16:9 aspect ratio for each image resolution as the video codecs function best when utilising width and height dimensions which are multiples of 16, 8, and 4 [30]. Furthermore, we evaluate resolutions in a range at multiples of eight as this produces a good spread for testing. These resolutions are also typically found in media streaming and smartphone cameras. The rate of capture is limited to 30 FPS to ensure consistency across the several resolutions because 60 FPS capture rates are often limited for 1080p+ resolutions.

For collecting performance data, we utilise the passive network measurement tool Qosium [31], which captures and logs traffic. A Qosium software probe is attached to both the OnePlus smartphone and the edge PC. The data is collected by a corresponding Qosium scope run as another process on the edge PC. To ensure that we obtain accurate results from Qosium, we synchronise the client and server OS clocks using the main-worker precision time protocol. Additionally, the smartphone is kept fully charged to maintain constant system performance and we disable background processes.

B. *Results and Analytics*

Fig. 5 presents evaluation results of the protocols on the 5G testbed, namely the client-to-server packet latency, jitter, and throughput metrics (more details in Section II-B). Furthermore, Fig. 6 and Fig. 7 present corresponding evaluation results on LTE and WiFi. The figure data points represent the median of the metrics from each 10 minute testing period.

We note that the number of packets transmitted during the testing period also varies with the testing resolution. At the smallest resolution, i.e., 144p, there is on average 33.2 packets sent per second, and comparatively, at the 4K upper limit, there is an average of 235.7 packets sent per second.

Latency in 5G vs LTE & WiFi. For 5G, the median and standard deviation client-to-server packet latency (Fig. 5(i)) across all protocols and resolutions is 19.35±8.63 ms. The latencies of all protocols follow a general increasing trend as image resolution increases. Among the 11 protocols on 5G, RTMP provides the lowest packet latency when transmitting

[2]https://github.com/MonaSolutions/librtmfp
[3]https://github.com/deradev/mpquicScheduler

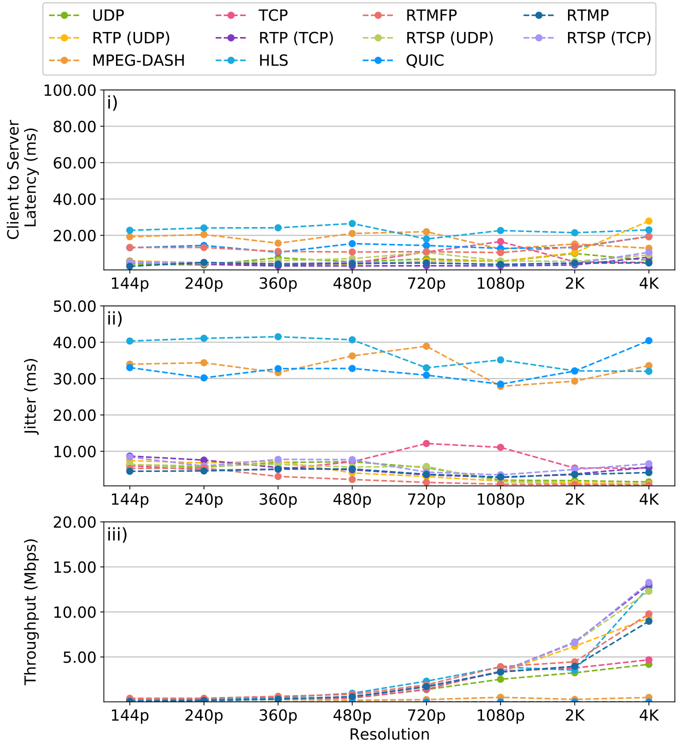

Fig. 7: Evaluation results for different multimedia and transport protocols, and resolution sizes while connected to WiFi.

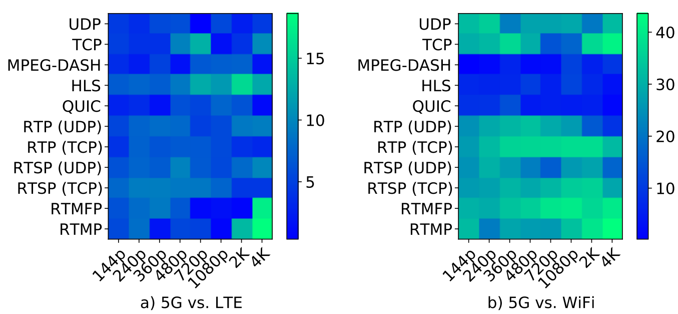

Fig. 8: Heatmaps of the percentage difference in client-to-server packet latency for the protocols at each resolution for the three network types: (a) 5G vs. LTE, and (b) 5G vs. WiFi.

images with resolutions from 144p to 480p, with a median of 14.03±1.05 ms. For 720p to 2K resolutions, HLS provides the lowest latency of 14.80±1.07 ms, and for 4K, QUIC provides the lowest latency of 18.52±2.95 ms. Across all the resolutions, RTMP has the lowest median latency with 15.14 ms and the highest is TCP with 20.90 ms. Corresponding LTE results (Fig. 6(i)) also support that RTMP provides the lowest packet latency across resolutions with a median of 18.69±5.61 ms, and the highest is TCP with 29.39±2.48 ms. RTMP experiences an 18.99% improvement when using a 5G connection as opposed to LTE. Finally, the WiFi results indicate that RTMP has the lowest median latency of 2.96 ms.

Performance comparisons of client-to-server latency for different protocols between 5G, LTE, and WiFi are presented as percentage difference heatmaps in Fig. 8. We observe from Fig. 8(a) that, from LTE to 5G, HLS has the overall largest relative decrease in packet latency among the protocols with 10.31%, while UDP has the smallest relative decrease of 4.08%. Overall, 5G provides a median 6.36% improvement in packet latency over LTE. However, WiFi still outperforms 5G by 28.2%. From 5G to WiFi, the protocols with the smallest relative decrease in latency are MPEG-DASH, HLS, and QUIC. The other multimedia protocols show a larger relative decrease, in some cases nearly double. This is attributed to the close geographical location of the WiFi hotspot, and because only two devices are connected to the hotspot, whereas our test network is a live network with substantially more devices, which consume communication resources. The WiFi connection is, therefore, less susceptible to other bursty network traffic which may impact the QoS metrics.

Jitter in 5G vs LTE & WiFi. From Fig. 5(ii), we observe over 5G, multimedia protocol jitter generally increases as image resolution increases. Though over 4G and WiFi (Fig. 6(ii) and 6(iii)), this is not the case and patterns are less clear. Specifically, for WiFi and LTE the jitter of many of the connection orientated (e.g., TCP-based) protocols appears to decrease after 1080p. This could be related to the congestion control algorithm found within these connection-based protocols.

Additionally, the general jitter inconsistency on 5G and LTE connections could be related to network congestion and the resulting instability due to traffic from large numbers of users and devices. For the WiFi connection, the increased jitter for multimedia protocols, such as HLS, RTP (UDP), and QUIC may indicate why their corresponding packet latency results are larger than the general trend of the other WiFi multimedia protocols results. [38] has similar findings where introducing background traffic on WiFi networks leads to generally higher latency delays and significantly higher jitter.

Throughput in 5G vs LTE & WiFi. 5G throughput is expected to increase as resolution increases because 5G supports higher data rates. Correspondingly, all three network types (5G, LTE, and WiFi) show increasing throughput as seen in Fig. 5(iii), 6(iii), and 7(iii). As opposed to minimising throughput, as with latency and jitter, the throughput should be maximised to allow clients to offload data as quickly as possible to servers. For 5G, RTMFP has the highest median throughput of 2.84 Mbps across the different resolutions, while RTSP (UDP) has the lowest with 0.0719 Mbps, between both these values, there is a percentage difference of 35.09%. In comparison, the throughput of RTP (TCP) and RTSP (TCP) are the highest on LTE and WiFi, they have a median value of 3.38 Mbps and 3.25 Mbps, respectively for the different connections. RTMFP has the next highest throughput on LTE with a median value of 2.86 Mbps, and on WiFi with 2.80 Mbps. For MPEG-DASH, RTP (UDP), RTSP (UDP), and UDP, they appear to follow a lower throughput trend than the other protocols while on 5G and LTE, they have a percentage difference of 28.59% and 31.71%, respectively. These UDP-based protocols may have a lower value as their packets are of

TABLE II: Comparison of the general requirements of a MAR protocol with the features of QUIC, HLS, and RTMP.

MAR Protocol Requirements [32]	**QUIC [33]–[35]**	**HLS [15]**	**RTMP [18]**
Designed purposes	Web apps and video streaming	Video streaming	Video, audio, and data messages
Low-latency and high fault tolerance	Low-latency through combining protocol version negotiation with other handshakes (cryptographic and transport). Fault tolerance through ack-based retransmitting of lost packets.	Low-latency HLS implementation [36] has low-latency support through generation of partial segments, preload hints, and rendition reports.	Fault tolerance mechanisms ensures that lost packets are retransmitted and received by the end devices.
Fairness to other concurrent network connections	Minimises per-packet bandwidth and computational costs by packaging many frames within a single packet.	Adaptive bitrate streaming changes received video streams according to network conditions.	Control messages allow for dynamic peer bandwidth change.
Multicast communications	No multicast communication, [37] provides a proposal for implementation.	Does not natively support multicast data streaming.	Standard implementation does not support multicast, however, UDP-based RTMFP supports this.
Ensuring user privacy	All packets, except acknowledgement packets, utilise TLS 1.3 encryption.	Media segments may be encrypted using AES-128 encryption method, or SAMPLE-AES.	Packets are not encrypted, although, RTMPS uses a TLS/SSL connection, and RTMPE supports encryption with a proprietary encryption mechanism.

a smaller size compared to TCP, and UDP does not manage congestion or flow control.

V. Discussion

From our 5G evaluation results, the key insights are:

- RTMP is the best performing protocol when considering the client-to-server packet latency, i.e., RTMP has the lowest median packet latency across all tested resolutions for the 5G, LTE, and WiFi networks. However, for higher resolutions of 720p to 2K, HLS has lower latency, and for 4K QUIC has the lowest packet latency.
- The jitter results indicate that on 5G and LTE, HLS experiences the smallest packet latency variations.
- The 5G throughput results show that RTMFP has the highest throughput capability.

Therefore, to fulfil the round trip latency requirement of MAR, RTMP is the most suitable protocol for lower resolutions, although, with higher resolutions, HLS and QUIC have lower packet latencies.

We additionally compare the general requirements for MAR protocols with the features of QUIC, HLS, and RTMP. Summarised in Table II, each protocol supports fault tolerance and minimises or adapts their individual bandwidth cost through different techniques. Both RTMP and HLS are native multimedia streaming protocols, and with the 5G latency and jitter results, either protocol would be suitable for transferring multimedia data in latency-constrained MAR applications, depending on the resolution of the source media data.

Impact of MAR systems, devices, and mobility models. We consider our MAR system to be a typical MAR scenario and the 5G test network to be comparable (in terms of setup and technologies) with commercial business providers. For example, Ericsson provides 5G edge networks for augmented and virtual reality for home and enterprise use cases [39], and their edge products have powerful GPUs to support computer vision tasks, such as object detection. In such a scenario, users are typically constrained to one location with a single MAR application running at a given time. Our experimentation reflects this and our results represent the baseline performance if these multimedia protocols were deployed in such scenarios.

On the other hand, we acknowledge that many types of MAR systems and technical or contextual factors need to be considered and evaluated in future work. Firstly, we only study an MAR system with multimedia data transmission to an edge server. However, cloud servers may support MAR, although the total round trip latency will increase. Secondly, we only evaluate MAR on smartphones and do not consider other MAR hardware (e.g., head-mounted displays). Thirdly, we only evaluate a static MAR scenario. As users move with their devices, network connection quality varies (due to varying signal propagation and handovers) and thus impacts the MAR experience. We will consider mobility scenarios as a part of future work. Fourthly, we only use a single encoding and pre-processing pipeline. In reality, media encoding varies and performing pre-processing, such as grayscaling or cropping video and images to regions of interest, significantly changes the size and frequency of video streams. Existing video analytic pipelines [40], [41] leverage these techniques as optimisations for transferring data to servers for object detection.

Limitations. In addition to the above-mentioned factors, we note two other limitations. Firstly, our research focuses on multimedia and transport protocols. However, lower layer wireless communication protocols can also impact system performance. We will look to use specialised low layer measurement tools, such as Mobile Insight [42], to capture detailed states and events (e.g., handovers) on these layers. Secondly, our research uses protocols in their default settings, better network performance may be achieved through fine-grained adjustments of protocol and networking parameters.

VI. Conclusions

In this work, we evaluated the performance of eight multimedia protocols and three transport protocols using a MAR system over a 5G edge testbed with QoS metrics, including client-to-server packet latency, jitter, and throughput. The evaluation also included measurements over LTE and WiFi connections for comparison. For both 5G and LTE, RTMP had the overall lowest median packet latency, of 15.14 ms, across the different resolutions. However, the lowest latency for the individual resolutions differed. Specifically, for 5G from 144p to 480p, RTMP had the smallest median packet latency of 14.03 ± 1.05 ms, then HLS from 720p to 2K with 14.80 ± 1.07 ms, and QUIC for 4K media with 18.52 ± 2.95 ms. We then further compared the described general MAR protocol requirements with the technical features of these three protocols. From our overall analysis, RTMP and HLS are the most promising candidates as MAR protocols in 5G.

In future work, we will further quantify the impact of connection quality on protocol performance with 5G and LTE mobile networks. In addition, we will evaluate alternative multimedia protocols with scalable and collaborative MAR scenarios, i.e., with multiple servers and clients, and test other transport protocols such as SCTP, MPTCP, and protocols designed for 5G networks, for example, Advanced 5G-TCP [43]. Finally, we will develop and utilise QoE-derived metrics and use video streaming in the system as opposed to sending raw video frames to evaluate protocol performance for MAR. This will complement the current metrics, as well as indicate the general end-to-end MAR performance.

REFERENCES

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE computer graphics and applications*, vol. 21, no. 6, pp. 34–47, 2001.

[2] T. Höllerer and S. Feiner, "Mobile augmented reality," *Telegeoinformatics: Location-based computing and services*, vol. 21, 2004.

[3] H. Chang, A. Hari, S. Mukherjee, and T. Lakshman, "Bringing the cloud to the edge," in *IEEE INFOCOM WKSHPS*. IEEE, 2014, pp. 346–351.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE IoT journal*, vol. 3, no. 5, pp. 637–646, 2016.

[5] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Challenges and opportunities in edge computing," in *2016 IEEE SmartCloud*. IEEE, 2016, pp. 20–26.

[6] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18 209–18 237, 2018.

[7] W. Zhang, B. Han, and P. Hui, "On the networking challenges of mobile augmented reality," in *ACM VR/AR Network*, 2017, pp. 24–29.

[8] ——, "Jaguar: Low latency mobile augmented reality with flexible tracking," in *ACM MM*, 2018, pp. 355–363.

[9] Huawei, "White paper for 5g clould vr service experience standards," Tech. Rep., 2019.

[10] 5GTN. (2020) 5GTN. [Online]. Available: https://5gtn.fi

[11] X. Su, J. Cao, and P. Hui, "5g edge enhanced mobile augmented reality," in *ACM MobiCom*, 2020, pp. 1–3.

[12] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos, Z. Chen, P. Pillai, and M. Satyanarayanan, "Quantifying the impact of edge computing on mobile applications," in *Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems*, 2016, pp. 1–8.

[13] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *ACM MobiCom*, 2019, pp. 1–16.

[14] C. Demichelis and P. Chimento, "Ip packet delay variation metric for ip performance metrics (ippm)," Tech. Rep., 2002, rFC 3393.

[15] J. R. Pantos and W. May, "Rfc 8216: Http live streaming," *IETF, Request for Comments*, 2018.

[16] I. Mpeg, "Information technology — dynamic adaptive streaming over http (dash) — part 1: Media presentation description and segment formats," *ISO/IEC MPEG, Tech. Rep*, 2019.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: A transport protocol for real-time applications," 1996.

[18] H. Parmar and M. Thornburgh, "Adobe's real time messaging protocol," *Copyright Adobe Systems Incorporated*, pp. 1–52, 2012.

[19] M. Thornburgh, "Adobe's secure real-time media flow protocol," *draft-thornburgh-adobe-rtmfp-10 (work in progress)*, 2013.

[20] H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund, and M. Stiemerling, "Real-time streaming protocol version 2.0," *IETF, December*, 2016.

[21] J. Iyengar and M. Thomson, "Quic: A udp-based multiplexed and secure transport; draft-ietf-quic-transport-34," *Internet Engineering Task Force: Newark, DE, USA*, 2021.

[22] L. Nuñez and R. M. Toasa, "Performance evaluation of rtmp, rtsp and hls protocols for iptv in mobile networks," in *CISTI*. IEEE, 2020, pp. 1–5.

[23] S. Bielievtsov, I. Ruban, K. Smelyakov, and D. Sumtsov, "Network technology for transmission of visual information," in *Selected Papers of the XVIII International Scientific and Practical Conference on Information Technologies and Security (ITS 2018).–CEUR Workshop Processing*, 2018, pp. 160–175.

[24] A. Aloman, A. Ispas, P. Ciotirnae, R. Sanchez-Iborra, and M.-D. Cano, "Performance evaluation of video streaming using mpeg dash, rtsp, and rtmp in mobile networks," in *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2015, pp. 144–151.

[25] S. Kesavan and J. Jayakumar, "Network performance analysis of cloud based multimedia streaming service," *International Journal of New Computer Architectures and their Applications*, vol. 4, no. 3, pp. 156–166, 2014.

[26] G. Camarillo, R. Kantola, and H. Schulzrinne, "Evaluation of transport protocols for the session initiation protocol," *IEEE network*, vol. 17, no. 5, pp. 40–46, 2003.

[27] P. Megyesi, Z. Krämer, and S. Molnár, "How quick is quic?" in *2016 IEEE ICC*. IEEE, 2016, pp. 1–6.

[28] M. Bishop and E. Akamai, "Hypertext transfer protocol version 3 (http/3) draft-ietf-quic-http-18," *Internet Requests for Comments, IETF Internet Draft, Tech. Rep.*, 2019.

[29] M. Soni and B. S. Rajput, "Security and performance evaluations of quic protocol," in *Data Science and Intelligent Applications*. Springer, 2021, pp. 457–462.

[30] brightcove. (2020) Optimal video dimensions. [Online]. Available: https://support.brightcove.com/optimal-video-dimensions

[31] Kaitotek Oy. (2020) Qosium. [Online]. Available: https://www.kaitotek.com/qosium

[32] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui, "Future networking challenges: The case of mobile augmented reality," in *IEEE ICDCS*. IEEE, 2017, pp. 1796–1807.

[33] R. Hamilton, J. Iyengar, I. Swett, and A. Wilk, "Quic: A udp-based secure and reliable transport for http/2," *IETF, draft-tsvwg-quic-protocol-02*, 2016.

[34] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.

[35] M. Thomson and S. Turner, "Using tls to secure quic," *draft-ietf-quic-tls-23 (work in progress)*, 2019.

[36] J. R. Pantos, "Http live streaming 2nd edition draft-pantos-hls-rfc8216bis-08," *IETF, Informational RFC*, vol. 8216, 2020.

[37] L. Pardue, R. Bradbury, and S. Hurst, "Hypertext transfer protocol (http) over multicast quic," *Internet Engineering Task Force.(August 2018). draft-parduequic-http-mcast-03*, 2018.

[38] O. Wellnitz and L. Wolf, "On latency in ieee 802.11-based wireless ad-hoc networks," in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*. IEEE, 2010, pp. 261–266.

[39] B. Ethirajulu, "How 5g and edge computing can enhance virtual reality," 2020.

[40] T. Zhang, A. Chowdhery, P. Bahl, K. Jamieson, and S. Banerjee, "The design and implementation of a wireless video surveillance system," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 426–438.

[41] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: scalable adaptation of video analytics," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 253–266.

[42] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, "Mobileinsight: Extracting and analyzing cellular network information on smartphones," in *ACM MobiCom*, 2016, pp. 202–215.

[43] I. Petrov and T. Janevski, "Advanced 5g-tcp: Transport protocol for 5g mobile networks," in *IEEE CCNC*. IEEE, 2017, pp. 103–107.