Context-aware and Reliable Transport Layer Framework for Interactive Immersive Media Delivery over Millimeter Wave

Hemanth Kumar Ravuri[1], Jakob Struye[2], Jeroen van der Hooft[1], Tim Wauters[1], Filip De Turck[1], Jeroen Famaey[2], Maria Torres Vega[1,3*]

[1]IDLab, Ghent University-imec, Ghent, Belgium.
[2]IDLab, University of Antwerp - imec, Antwerp, Belgium.
[3]eMedia Lab, KU Leuven, Leuven, Belgium.

*Corresponding author(s). E-mail(s): maria.torresvega@kuleuven.be; Contributing authors: hemanthkumar.ravuri@ugent.be; jakob.struye@uantwerpen.be; jeroen.vanderhooft@ugent.be; tim.wauters@ugent.be; filip.deturck@ugent.be; jeroen.famaey@uantwerpen.be;

Abstract

In order to achieve truly immersive multimedia experiences, full freedom of movement has to be supported, and high-quality, interactive video delivery to the head-mounted device is vital. In wireless environments, this is very challenging due to the massive bandwidth and ultra-low delay requirements of such applications. Millimeter wave (mmWave) networks promise ultra-high speed owing to the availability of high-capacity bands at a frequency range of 30 GHz to 300 GHz. However, they are prone to signal attenuation due to blockage and beam misalignment due to mobility, leading to packet loss and retransmissions. This can lead to the head-of-line blocking problem on the transport layer which results in playout stalls and delivery of lower quality data that can be highly detrimental to a user's quality of experience (QoE). Complementary to research efforts trying to make mmWave networks more resilient through lower-layer enhancements, this paper presents a transport layer solution that provides an adaptive and reliable transmission over mmWave networks-based on partially reliable QUIC. Using context information retrieved periodically from the client to adapt according to the networking conditions induced due to mobility and obstacles, the essential part of the video content (i.e., in the viewport of the end user) is transmitted reliably, while less important content (i.e., outside of the viewport of the end user) is sent unreliably. Our decision-making logic is able to effectively deliver 22.5% more content in the viewport reliably. This is achieved without additional playout interruptions or quality changes for scenarios with high-bitrate volumetric video streaming evaluated over realistic mmWave network traces. In case the server can perfectly predict the network bandwidth, playout interruptions can be avoided altogether.

Keywords: millimeter Wave, IEEE 802.11ad, Virtual Reality, Point Clouds, QUIC

1 Introduction

Interactive Virtual Reality (VR) aims at providing a completely immersive experience to the user, letting them navigate freely in a scene with six degrees of freedom (6DoF). Recent advances in head-mounted device (HMD) technology have enabled a shift from tethered devices, relying on local servers for content generation, to standalone devices, which generate content fully on-device. Such standalone devices are expected to become one of the key enablers of interactive VR, as they do not limit a user's degree of motion or cause a tripping hazard, due to the cable [1]. However, such devices have limitations, e.g., low onboard processing capabilities. To solve this, processing could be offloaded onto remote servers, which would be connected wirelessly to the HMDs (untethered). Figure 1 represents an example of an untethered interactive VR system.

In order to deliver high-quality content and enable truly immersive and interactive experiences for the user, wireless networks are expected to support High-Rate and High-Reliability Low-Latency Communications (HR2LLC) [2, 3]. Currently, wireless transmissions operate on low-frequency bands (up to 5 GHz) with limited resources in terms of data rate, making it difficult to meet the imposed stringent bandwidth and latency requirements.

With its abundant spectrum resources, the millimeter Wave (mmWave) band (30 GHz to 300 GHz) is a prime candidate for meeting the requirements set by interactive VR. This band allows for ultra-high throughput and ultra-low latency communications [1, 4]. However, mmWave is susceptible to blockage and the signal attenuates quickly with distance. Moreover, since mmWave operates at high frequency, it has less penetration power and therefore requires a clear Line of Sight (LoS) between the transmitter and receiver for optimal performance. To mitigate these issues, beamforming has been adopted as a strategy, where the energy is focused on specific directions to ensure high signal strength where desired. While beamforming is useful, it can be difficult to implement in a mobile environment, where the beam must be adapted quickly to avoid misalignment of transmitter and receiver [5]. This becomes particularly challenging in a VR environment, which involves high mobility as the users walk around interacting with each other and other objects in a scene. Here, objects in the physical environment and the bodies of other users can create a blockage [1, 6]. Figure 2 illustrates such a scenario, where the hand of one user obstructs the LoS between

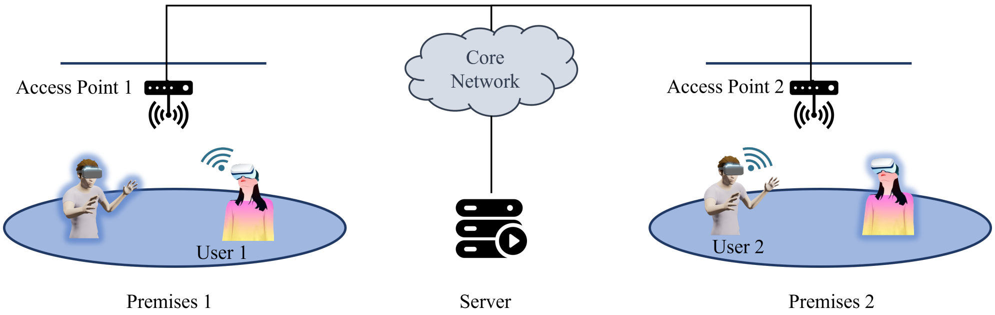


Fig. 1: mmWave-based untethered interactive immersive media delivery system.

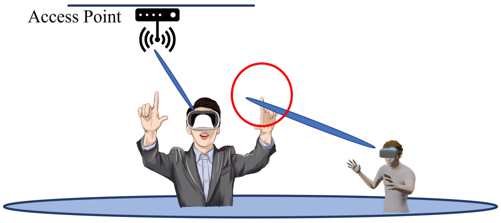


Fig. 2: Blockage created by a user in a wireless VR scenario.

the wireless HMD of another user and the access point. Such inter-user and self-body blockages can deteriorate the performance of mmWave links, increase video stalls, and reduce video quality, which will significantly affect the Quality of Experience (QoE). To cope with these issues, there is a need for network management and optimization techniques.

Nowadays, work is being done on different layers of the protocol stack to address the challenges posed by mmWave networks. Most of the approaches focus on designing the infrastructure and cross-layer solutions that integrate the physical layer and media access control layer (PHY-MAC) [7, 8]. Some works are also aimed at providing dual connectivity for multiple channels [9]. However, the conflicts between the mmWave channel and the upper layer are still severe [7]. In our recent work [1, 6], we experimentally evaluate the performance of the most widely deployed reliable transport layer protocol, Transmission Control protocol (TCP), while delivering VR content on mmWave networks using a novel testbed capable of repeatably creating blockage through mobility. Our evaluation shows that mobility can cause the throughput to drop briefly by almost 50%, creating a playout freeze in a video streaming session. Furthermore, throughput degrades severely when there is no LoS. In addition, TCP's congestion control over-corrects even for brief periods of throughput drops, leading to highly variable segment download times. Finally, TCP suffers from the Head-of-Line (HOL) blocking problem on lossy networks, making it unviable for delivering interactive VR content over mmWave networks [10]. In conclusion, our results suggest that the successful implementation of mmWave technology in future wireless networks is arguably dependent on an efficient adaptation of the transport layer protocols over mmWave.

Recently, we have proposed a partially reliable transport layer based on Quick UDP Internet Connections (QUIC) to deliver immersive media on lossy networks [10]. Using the natural restriction on human sight (120 degrees peripherally), we argue that only the part of the content in the user's viewport needs to be sent reliably and the rest unreliably. Evaluation shows that the framework outperforms reliable protocols such as TCP and QUIC under impaired networking conditions like high packet loss. These results encouraged us to deploy the framework on mmWave to support interactive VR. However, the framework is static in terms of varying the reliability, i.e., only a pre-fixed amount of content (e.g., number of point cloud objects) can be sent reliably and the rest unreliably. Given that the use case under consideration involves high mobility, the user's viewport keeps changing thereby changing the number of objects visible to the user. Therefore, it is important to adapt the amount of data being sent reliable based on the user viewport. As mentioned earlier, mmWave networks involve highly dynamic networking conditions, which need corresponding adaptation in terms of quality to avoid congestion and playout freezes. To this end, we propose a server-based streaming framework that delivers interactive VR content on mmWave networks with adaptive reliability and quality based on the context information gathered from the client.

The contributions of this paper are as follows:

1. We propose a server-side streaming framework based on partially reliable QUIC at the transport layer for delivering immersive media content. The server is equipped with a decision-making module that adapts the quality and reliability based on the changing conditions of the mmWave networks. The choice of having decision-making on the server side is to avoid sending multiple requests from the client, which can add to the delays amounting to at least one Round-trip time (RTT) per request, which is not suitable for interactive VR applications. The server receives context information from the client, and uses this information to make decisions on quality and degree of reliability. The context comprises information about the user, network, and content. To the best of our knowledge, this is the first work that integrates the concept of adaptive reliability with mmWave networks.
2. We propose a heuristic to take quality and reliability decisions using the context information. This heuristic considers networking parameters such as the perceived throughput by the client, and user-related information such as the position of the user in the scene, and the current viewport, to take decisions.
3. Then, we perform experimentation driven by realistic mmWave traces. Firstly, we present the traces collected to emulate mmWave networks, using commercial-off-the-shelf mmWave routers and a configurable robot to provide mobility. The testbed has been designed in such a way that the robot, which acts as a client, induces mobility into the setup. The server has been connected to a router that acts as an access point, and furniture has been deployed to create a blockage between the access point and the client. Next, we evaluate the proposed framework using a client-server-based implementation, where the client is a headless player capable of

receiving and playing out volumetric video (point clouds). Here, we provide various details about various function blocks of the client and the server. Along with the proposed mechanism, the implementation supports reliable QUIC and static partially reliable QUIC for the sake of evaluation. Then, we perform an extensive experimental evaluation using collected throughput traces. As part of the evaluation, we vary parameters such as the frequency of delivery of context information from the client to the server and incorporate scenarios such as perfect bandwidth prediction to fit in the proposed transport layer framework over mmWave networks.

Our evaluation shows that with a clear LoS between the sender and the receiver, mmWave networks provide high bandwidth consistently for the seamless delivery of immersive media. In such conditions, the proposed mechanism and the static partially reliable QUIC outperform the reliable variant of QUIC in terms of throughput. Next, when there is an obstacle obstructing the LoS, the proposed mechanism and the static partially reliable mechanism deliver content at a higher quality compared to the reliable variant of QUIC. However, a huge number of playout freezes are observed in all three cases, with the proposed mechanism having a comparatively lower (5%) number of freezes. With an increase in context frequency, the number of freezes come down considerably, with the proposed mechanism having 50% lower freezes while delivering 22.5% more objects in the user viewport reliably. In addition, the number of freezes comes down to almost zero when perfect bandwidth prediction is deployed on the server side. By delivering more content reliably with fewer playout interruptions and higher quality, the proposed framework improves the user's QoE.

The remainder of this paper is organized as follows. A brief background on partially reliable QUIC, achieving 6DoF over mmWave networks, and an overview of the related work are presented in Section 2. Our envisioned mmWave-based interactive immersive media delivery system and its functioning principles are presented in Section 3, followed by the experimental setup and results in Section 4. Finally, this paper is concluded in Section 5.

2 Background

This section provides a short background description of our prior work on partially reliable QUIC, followed by the state of the art on achieving 6DoF over mmWave. Finally, it presents the related work using QUIC over mmWave.

2.1 Partially Reliable QUIC

Currently, reliable video streaming applications rely on the hypertext transfer protocol (HTTP), which uses the transmission control protocol (TCP) at the transport layer. However, TCP has been plagued by longstanding issues like HOL blocking, slow connection setup times, and slower retransmissions. HOL blocking refers to the case where the client must wait until earlier requests are processed to send a new request. This increases the service's response time, as the sequence of packets is held in the queue when one or more packets of earlier requests are lost. These factors impact the throughput and add to the delay, thereby proving detrimental to the application's performance. Such delay can lead to a video-stalling event (a so-called freeze), which can negatively impact the user's QoE and visual attention [10]. As discussed previously, our prior work shows that TCP is not a viable solution for streaming content over mmWave networks [1]. Recently, QUIC has gained traction in the research community, as it promises to overcome the shortcomings of TCP without compromising on reliability. However, while QUIC vastly reduces the connection establishment time and HOL blocking, thus increasing interactivity, it still underperforms while delivering multimedia due to retransmissions under lossy conditions. To cope with these, QUIC offers the possibility to support unreliable delivery, like that of the user datagram protocol (UDP) [11]. While live-video streaming applications usually opt for completely unreliable protocols, such an approach is not optimal for immersive media delivery, since it is not affordable to lose certain data that might affect the end user's QoE. A partially reliable QUIC-based data delivery proposed in our recent work [10], aims at bringing the best of both worlds together, as the mechanism that supports both reliable (streams) and unreliable (datagrams) delivery.

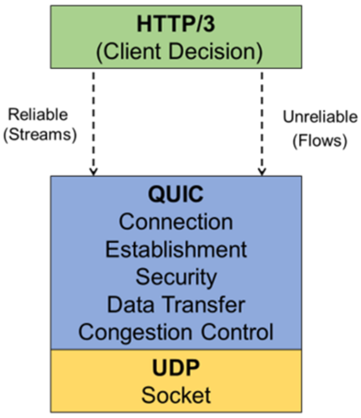


Fig. 3: Partially Reliable QUIC.

To realize partial reliability, we envision the co-existence of reliable streams and unreliable datagrams under one connection. The steps involved in realizing the envisioned mechanism are presented in this section. First, a QUIC-enabled client initiates the connection with a server. Ideally, the connection setup process involves a negotiation of transport layer parameters, like the congestion control algorithm, maximum packet size, etc. Both endpoints must exchange an additional parameter max_datagram_frame_size to support partial reliability. This parameter represents the maximum size of a datagram frame that an endpoint is willing to receive. The absence of this parameter indicates that the endpoint is not compatible with unreliable data delivery. Endpoints are required to comply with the restrictions implied by these parameters to successfully establish a connection. Once the connection has been established, as shown in Figure 3, it is in the purview of the client to prioritize data and choose the type of delivery mechanism. The client can either send requests to the server reliably using GET requests or unreliably using datagrams. The server responds to the reliable request by sending a response message followed by the body message on reliable streams. These can be identified by the client using a common identifier called the Stream ID. In the case of unreliable requests, the server directly responds to the message by sending the requested body in a series of datagrams. These datagrams can be identified using a common Flow ID. The congestion control and flow control apply similarly for both streams and flows. Since the datagram extension is still a work in progress, there is currently no provision to let the client know when the end of the message is reached. One of the options is to send a standard message in the last datagram. However, there is no guarantee that the message will reach the client. The second option is to use the control stream on the HTTP layer to reliably send an end-of-the-data message to the client [10].

This summarizes the functionality of the partially reliable delivery mechanism using QUIC. Our proposed framework moves the decision-making to the server side and adapts the reliability based on the changing networking conditions. We introduce the concept of context, where the client tracks the information constantly and updates the server periodically.

2.2 Achieving Interactive VR Over mmWave Networks

As discussed previously, mmWave networks have challenges related to mobility and blockage. Research has presented various approaches to deal with them. The following sections present a summary of such approaches. The three main categories of anti-blockage approaches include multi-connectivity, beamforming, and relay assistance [12].

Multi-connectivity is an approach to maintain a reliable session by providing multiple access points to the user equipment simultaneously. This enables consistent data transmission: even if one of the connections is interrupted due to blockage, the other(s) can still be utilized. While multi-connectivity-aided mmWave networks can improve the network capacity and the outage probability, there are open questions regarding balancing the system complexity and achievable performance [12].

As an alternative blockage method, beamforming steers the signals from the transmission antenna array in the desired direction, forming a directional beam. In practice, most mmWave systems deploy large-scale antenna arrays and consequently fully digital beam forming is impractical as each antenna element requires a radio frequency chain [12]. In addition, beamforming also poses challenges in mobile environments, as a beam must be steered quickly according to the motion to avoid beam-misalignment and outage issues.

Finally, relay-aided communications help to circumvent the obstacles and extend the coverage. When a direct transmission link gets blocked, relaying through alternate paths aids in offering higher data rates than direct links, and saves transit power. However, placement and mobility of a relay station is an open research challenge. Furthermore, mobility behavior of communication entities may cause frequent handover among relay stations resulting in complex scheduling, wastage of power, and additional delay. Also, incorporating intelligent reflective surfaces as an alternative for relay stations is being considered widely, however it is still an open research challenge [13].

These approaches are all at the infrastructure level, which incurs cost and time for establishment. However, these can be complementary to the framework presented in this paper.

Most of the work done at the transport layer has been limited to multipath protocols, which utilize the dual connectivity at the infrastructure level to deliver content [14]. Such transmission is realized by connecting the client device through two channels, i.e., LTE and mmWave, at the same time to form a composite channel. This means that when an application communicates with a traditional TCP socket, the socket can transparently handle multiple sub-flows on different interfaces, such as Wi-Fi, cellular networks, and Ethernet. The most widely researched protocol in this context is Multipath TCP (MPTCP). For immersive media delivery, a multipath approach could be used to stream low-quality baseline representation of the VR content on the conventional sub-6 GHz Wi-Fi, while high-quality user viewport-specific enhancement representation that synergistically integrates with the baseline representation can be sent over the high-frequency mmWave interface. The original design goals of MPTCP are throughput improvement, redundancy, and congestion balance. However, MPTCP is known to suffer issues related to simultaneously handling congestion control and flow control of parallel sub-flows. This problem is exacerbated in mmWave channels, which jitter rapidly and restart transmission frequently [7]. Furthermore, such a multipath approach is only possible when there are at least two paths available, which may not always be the case.

So far, very few works have focused on delivering 6DoF content on mmWave networks. In their work [15], Chakereski et al. evaluate the performance of mmWave networks and free space optics (FSO) as candidate frameworks to deliver 6DoF VR content. To this end, they analytically characterize the key components of the envisioned dual-connectivity streaming system that integrates edge computing and scalable 360° video tiling, and formulate an optimization problem to maximize the immersion fidelity delivered by the system, given the Wi-Fi and mmWave/FSO link rates, and the computing capabilities of the edge server and the user's VR headsets. The authors carry out simulation-based experiments to assess the performance of the proposed system using actual 6DoF navigation traces from multiple mobile VR users. While this work relies on dual connectivity, which is an infrastructure-level solution, the current paper proposes a transport layer solution that uses the available infrastructure. Hence, as mentioned earlier, such works can be complementary to each other.

Zhang et al. experimentally demonstrate the challenges of streaming high-quality volumetric videos to multiple users and conclude that the user viewport-similarity can be leveraged to effectively optimize the network resource utilization using multicast over mmWave [16]. The authors present a holistic research agenda for improving the performance and QoE for multi-user volumetric video streaming on commodity devices. Their agenda includes joint viewport prediction and blockage mitigation for multiple users, multicast grouping based on viewport similarity, customized mmWave beam design for efficient multicast, and mmWave-aware multi-user video rate adaptation. This work highlights the importance of each block in their research agenda and the open challenges therein. However, this work is more on the proposal level while this paper has proposed, implemented, and evaluated the performance of 6DoF streaming mechanism over mmWave networks.

Struye et al. propose CoVRage [17], a receive-side beamforming solution tailored for VR HMDs. They argue that in wireless VR, where the antenna is integrated into the HMD, any head rotation in turn changes the antenna's orientation. As such, beamforming must adapt, in real-time, to the user's head rotations. An HMD's built-in sensors providing accurate orientation estimates may facilitate such rapid beamforming. They have explored an algorithm that estimates how the Angle of Arrival (AoA) at the HMD will change in the near future, and covers this AoA trajectory with a dynamically shaped oblong beam, synthesized using subarrays. Again, this work is more focused on providing a lower-layer solution such that a consistent connection is ensured. This can indeed be complementary to the proposed solution, where the focus is on the upper layers.

The current work assumes that there is only one path available and adapts the delivery mechanism according to the conditions of the path and delivers the content at the highest possible quality while aiming at reducing the playout interruptions. Hence, the work is agnostic to the underlying infrastructure, and additional infrastructure can only be complementary to it.

2.3 QUIC over mmWave Networks

To the best of our knowledge, this is the first work to deploy the concept of adaptive reliability of QUIC over mmWave networks. There are a few works that evaluate the performance of the QUIC protocol over mmWave networks, focusing on either evaluating the performance of the multipath extension of QUIC, or that of the congestion control algorithms. In the domain of Multi-path QUIC (MPQUIC), Wu et al. [18] perform an empirical evaluation of state-of-the-art multipath schedulers based on real 5G data, for both static and mobile scenarios. To this end, they deploy MPQUIC as the enabler for multipath transmission. The authors highlight the benefits of multipath over single-path transmissions, but also illustrate that it is not trivial to achieve high performance with current multipath schedulers using predefined policy-based algorithms, since the uncertainty and dynamics of 5G networks are not well captured over time. Furthermore, they observe that MPQUIC opens up several opportunities by making use of available features of QUIC and futuristic extensions such as the unreliable datagram. The current work deploys the unreliable datagram extension of QUIC to achieve partial reliability under one connection. This paper takes the framework to the next level by adapting the reliability itself based on the underlying networking conditions for seamless immersive media delivery. In the domain of congestion control evaluation, Haile et al. [19] propose a variant of bottleneck bandwidth and round-trip propagation time congestion control (BBR) called the receiver-driven BBR (RBBR) to improve the performance of QUIC on highly variable wireless networks. Their algorithm shows significant improvement on 4G LTE networks, but the performance difference between mmWave and 4G is not very significant. The authors suggest that more work needs to be done on congestion control for QUIC while deploying it on mmWave networks. The current work deploys CUBIC, a loss-based congestion control algorithm, instead of BBR on the transport layer.

So far, only a few works have appeared on deploying partially reliable data delivery for video streaming. However, none of the works adapt reliability based on the

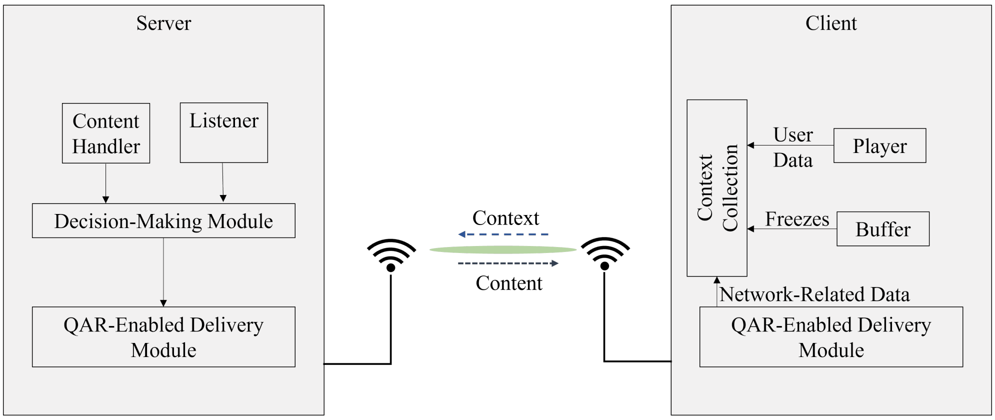


Fig. 4: Envisioned mmWave-based interactive immersive media delivery system.

underlying networking conditions. One work that comes close is the study by Michel et al. [20], who take a different approach in realizing a partially reliable data delivery mechanism using QUIC. They design a variant of forward error correction (FEC) called flexible erasure correction and integrated it with QUIC. This enables an application to select the level of reliability, ranging from completely reliable (using retransmissions) and completely unreliable (using various FEC mechanisms to recover the lost data). Through experimental analysis, the authors show that under lossy conditions, their approach outperforms the reliable variant of QUIC in terms of download time. However, while transferring files with size higher than 1 GB, their approach adds substantial overhead, leading to a drastic decrease in the throughput and very high CPU utilization. In addition, FEC requires an additional step of encoding and decoding the content, which adds to the overall latency and is not desirable in the applications such as interactive virtual reality. The work in this paper does not add any additional computational latency or overhead in terms of throughput, since we utilize QUIC's datagram extension to realize adaptive reliability.

3 Envisioned Client-Server-based Framework

Figure 4 presents our proposed immersive media delivery system with higher reliability, lower service delay, and higher quality over mmWave networks. We envision a system where the decision-making is done on the server side based on the periodic context information sent by the client. The functionalities of various building blocks of the envisioned system are described below.

3.1 Content Handler

The content handler carries out the most important objective of the server, i.e. to process and store multiple representations of immersive media content (e.g., point clouds) encoded at different qualities and divide it temporally into segments. Based on the processing capabilities, the segment duration can go down to $1/f_n$ where f_n

Table 1: Incoming events to the server.

Event	HTTP/3 Message
Connection Request	`HTTP/3_SETTINGS` frame
Service Request	`HTTP/3_GET` request
Context Update	`HTTP/3_POST` request

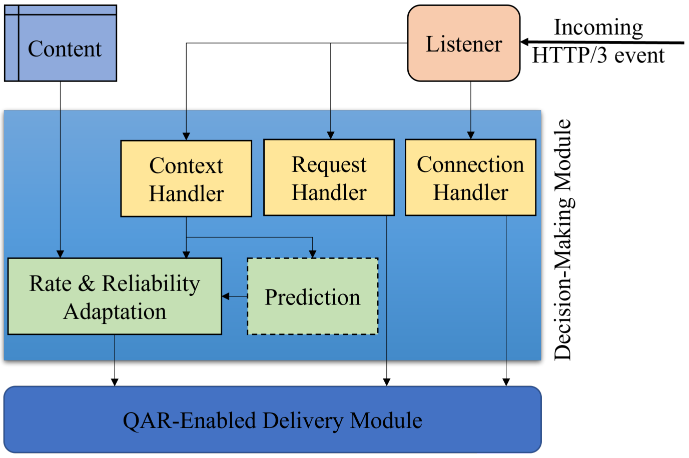


Fig. 5: Schematic of the decision-making module.

denotes the number of frames per second of video. Currently, immersive media like point clouds use MPEG's video point cloud compression (V-PCC) for achieving a higher compression ratio up to 1000 [21]. Current coding techniques are not known to be loss-resilient and cannot handle packet loss. In such cases, the content can be further encoded using a FEC mechanism [22] based on the expected packet loss ratio to make sure that the client retrieves the content. However, this depends upon the available bandwidth, as this step adds overhead and also induces additional processing latency.

3.2 Listener

The listener runs as an independent entity, polling the socket for incoming events carried as `HTTP/3` messages from the client. The client can initiate three major types of events, namely, *connection request*, *service request*, and *context update*. Table 1 presents the events and their corresponding `HTTP/3` messages. The listener filters the message and triggers the corresponding event processors, which are part of the decision-making module. Furthermore, the server can deploy the listener to monitor the connection-related statistics such as the round-trip time (RTT), experienced packet loss, and the number of retransmissions that can be used by the decision-making module.

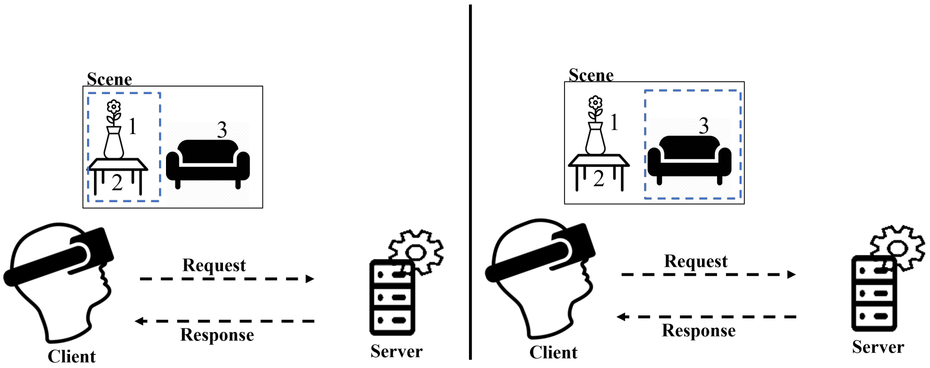


Fig. 6: An illustration of the user changing their viewport in a scene.

3.3 Decision-Making Module

The decision-making module is the core of the framework, where the major part of the processing and ruling is done. Figure 5 presents the schematic of the module, illustrating the data flow. As discussed previously, an incoming event is filtered by the listener and sent to the corresponding handler for processing.

3.3.1 Handlers

As in typical client-server scenarios, the client initiates the connection at the transport level using partially reliable QUIC (QUIC)'s crypto handshake. While connection-level options pertaining to the core transport layer protocol are set in the initial crypto handshake, settings specific to `HTTP/3` are conveyed in the `SETTINGS` frame. After the QUIC connection is established, an `HTTP/3_SETTINGS` frame is sent by the client as a control stream, which is handled by the connection handler by responding with its `HTTP/3_SETTINGS` frame. Once the settings are negotiated and the connection is established, the client sends an `HTTP/3_GET` request to start the video streaming session. The request handler responds by sending all the objects, reliably at the base quality, to fill the client buffer. Typically, connection and request handlers are triggered only once during a session. These handlers are envisioned to be running in the application layer, and they are mapped to the transport layer, i.e., QPR through QUIC-with adaptive reliability (QAR)-enabled delivery module. Finally, the context handler processes the user context updates through the `HTTP/3_POST` request. The handler is triggered periodically with updates, where the BODY of the request is parsed to extract the context information, which is then utilized for the rate and reliability adaptation.

3.3.2 Rate and Reliability Adaptation

Based on the available user context and the characteristics of the content, the quality and the delivery adaptation decisions are taken here. The mechanism to arrive at a decision itself can vary depending upon the type of parameters available in the context, context frequency, and content-related parameters such as the number of objects to be sent and the available quality representations of the content. However, as a general rule of thumb, we envision a system where all the objects in the viewport are sent reliably. Research suggests that the stereoscopic field of view (FoV) for humans is around 60 degrees and the peripheral FoV is about 120 degrees [2]. In the past, adaptive video streaming techniques have exploited this concept to deliver only the content in the FoV at the highest available quality, and the rest at a lower quality [23]. Given that the goal is to achieve true 6DoF, we can assume that the user interacts dynamically with the scene. Along with the user's gaze, their position keeps changing in order to interact with various objects in a scene. As presented in Figure 6, the user looking at objects 1 and 2 may dynamically change their gaze towards object 3. In such a situation, state-of-the-art QPR fixes the relative number of objects that are sent reliably, while a more dynamic approach is required to stream highly interactive immersive video over highly dynamic mmWave networks.

Since, mmWave networks are prone to signal attenuation due to blockages, the available bandwidth is expected to fluctuate, especially when a user is moving. In such a situation, along with reliability, the quality of the content being delivered should also be adapted according to the changing network conditions. Thus, optimal quality/reliability adaptation algorithms have to be deployed to make sure that the client's QoE is not affected.

Algorithm 1 presents a heuristic that can adapt quality and reliability while delivering volumetric media like point clouds. Table 2 presents the context information sent by the client to the server. The algorithm uses this information sent for taking adaptation decisions. The context is discussed in detail in the subsection about the client.

A rate/reliability adaptation heuristic is presented in Algorithm 1, where the visible area of the point cloud objects to the user is used to rank them. Given the user context for a set of frames, the visible area per object is calculated using the user's position, focus, and the object placement in the scene. This heuristic determines the most appropriate quality level for each object using the perceived bandwidth of the client during the previous iteration of delivery. To this end, a size function is used which determines the file size corresponding to the given frame range, object, and quality. The heuristic starts by assigning the lowest possible quality and initializing the delivery mode to be reliable for all the objects (lines 1). If the client has just started playing, the server delivers the lowest quality reliably in order to fill the buffer (lines 2). Once the buffer is full, next, the total budget is calculated based on the available bandwidth and total duration of the content in the range of frames being sent (line 4). If the total file size of all objects exceeds the budget, then the lowest quality is allotted to each object (lines 4-6). Then, the delivery mode is determined by checking the visible area of each point cloud object to the user during that duration. If the visible area equals zero, then the point cloud is not in the viewport of the user. Such objects are set to be delivered unreliably (lines 7-11). If the total file size is less than the budget bits, the heuristic goes ahead with ranking the point clouds based on the visible area. Then, starting with the object with the highest visible area, the heuristic increases the quality until there is no bandwidth left or all the objects are assigned the highest quality (lines 12-19).

Algorithm 1: A heuristic to allot quality and reliability to the objects in the viewport greedily.

```
Input  : $F_n$, the frame range for which the context is received
         $t$, time since last playout
         $objects$, point cloud objects
         $bandwidth$, perceived bandwidth by the previous iteration [b/s]
         $f_c$, context frequency
         $n_q$, the number of quality representations
         $delivery$, the available delivery mechanisms, $0, 1$
         $x, y, z$, the user's coordinates
         $d_x, d_y, d_z$, the user's focus
         $scene$, location and translation of objects in a scene
         $A_v$, the visible area of each object as $(x, y, z,\ d_x, d_y, d_z,\ scene)$
Output: $qualities$, specifying the selected quality representation
         $deliveries$, specifying the selected mode of delivery for the objects

$qualities \leftarrow [1, \forall pc \in objects]$
$deliveries \leftarrow [1, \forall pc \in objects]$
Initialize at lowest quality and reliable delivery
if $t = 0$ then
    return $qualities, deliveries$
end
$budget \leftarrow bandwidth.1/f_C$
$bits \leftarrow \sum_{pc \in objects}$ Size$(F_n, pc, 1)$
if $bits \geq budget$ then
    return $qualities, deliveries$
end
for $pc \in objects$ do
    if $A_v = 0$ then
        $deliveries[pc] = 0$
    end
end
for $pc \in$ Sort(objects, dist) do
    for $q \in [2; n_q]$ do
        $cost \leftarrow$ Size$(F_n, pc, q)$ - Size$(F_n, pc, q-1)$
        if $bits + cost > budget$ then
            break
        end
        $bits \leftarrow bits + cost$
        $qualities[pc] \leftarrow q$

    end
end
return $qualities, deliveries$
```

Table 2: Context information sent by the client.

Information	Source
Last played frame	player
Time since last playout	player
Scene	player
Bandwidth	network
Position	user
Focus	user
Visible area of each object in a scene	user

3.3.3 Prediction

Given the dynamic nature of the mmWave networks, having periodic context information from the client may not be sufficient for delivering the right content in the right way. This is because the most recently received context information may already be stale when taking a decision. There can be situations where the context information is not available on time or remains useless by the time it reaches the server. In such a scenario, it is necessary to incorporate an artificial intelligence (AI)-based prediction component that uses historical context data and state-of-the-art AI algorithms to predict how user actions will change over time. In addition, prediction of the networking conditions is equally essential to take the right delivery/quality decisions on time. In the current work, we have incorporated a case when there is no perfect bandwidth prediction and also a case where we assume that this module predicts the available bandwidth perfectly in order to highlight the importance of this module.

3.4 QAR-Enabled Delivery

This component provides an abstraction of the transport layer and implements the decisions taken by the application layer. The primary functions of this module are as follows:

- Handling reliable streams and unreliable datagrams: The module is responsible for managing the lifecycle of a stream. Based on the decisions taken, it dynamically creates and terminates reliable and unreliable streams and manages their corresponding flow control.
- Congestion control: This module takes care of the congestion control based on the negotiated congestion control algorithm of the client.
- User Datagram protocol (UDP) optimizations: The module manages optimizations such as generic segmentation offloading (GSO) for UDP that enables applications to bundle and transfer multiple UDP segments between user space and the kernel at the cost of one. This reduces the computational overhead incurred due to the user-space deployment of QUIC.
- The module generates corresponding QUIC packets that will be sent to the UDP socket for delivery.

3.5 Client

The client is responsible for playing out the content for the user and collecting context information, which is delivered periodically to the server. The main components of the client are as follows:

- As discussed earlier, the QAR-enabled delivery module carries out the same functionalities in the case of the client as it does in the case of the server. In addition, it is responsible for initiating the connection with the server and negotiating the transport layer parameters needed for adaptive reliable delivery. Furthermore, it retrieves the content sent by the server and fills up the buffer. It also shares the statistics related to the connection with the context collection component. If the server deploys FEC, it sends the retrieved content to be decoded before filling up the buffer.
- Given the dynamic nature of mmWave networks, the buffer plays an important role in making sure that there is always content available for the uninterrupted video-playing experience. However, having a large buffer size does not go well with interactive immersive media, particularly where 6DoF content is involved. Since the user keeps interacting with objects in the scene dynamically, there is a possibility of data in the buffer turning obsolete. Hence, having an optimal buffer size is essential by trading off the requirements of the network and the application. The current work considers a buffer of one second, given the variability of the link capacity of mmWave networks.
- Since the decision-making is done on the server, the player is responsible for decoding and rendering the content. It initiates the connection with the server on the application level. It is responsible for negotiating the application-level settings with the server. Furthermore, it tracks the user's viewport and position in order to share it with the context collection component.
- The context collection component collects data from other components such as the freeze-related information from the buffer, network-related information from the delivery module, and user-related information from the player. It generates an `HTTP/3_POST` request with the context information embedded in the body and delivers it periodically to the server.

3.6 Context

The context comprises the key information collected by the client from different sources and sent to the server for decision-making. The client collects information about the user, network, and video player periodically and shares it reliably using an `HTTP/3_POST` request. The user information usually involves pose, gaze, and hand tracking. The pose information comprises the position and orientation of the user in a scene. This is crucial for providing an immersive experience to the user, as the scene should adapt rapidly and accurately to any rotational and translational motion performed by them. Next, the user's gaze information is essential for rendering those elements of the environment that are in their viewport. Current HMDs allow gaze tracking, and a few support hand-tracking with their on-device cameras and gloves.

Table 3: Observed bitrates (Mb/s) of four point cloud objects.

Quality	#1	#2	#3	#4
1	2.40 ± 0.01	3.50 ± 0.03	4.48 ± 0.01	5.00 ± 0.03
2	3.62 ± 0.01	5.03 ± 0.04	7.14 ± 0.01	8.65 ± 0.05
3	5.81 ± 0.03	7.83 ± 0.05	12.08 ± 0.03	15.86 ± 0.11
4	10.00 ± 0.07	13.63 ± 0.09	21.95 ± 0.07	30.16 ± 0.21
5	18.00 ± 0.13	24.72 ± 0.24	40.35 ± 0.14	53.51 ± 0.37

As discussed earlier, the decision-making module relies on the pose and gaze information to prioritize the content and adapt the reliability and quality. Depending upon the use case, hand-tracking information is useful in creating an increased sense of immersion and presence. Then, the client collects information from the player, such as the position and translation of objects in a scene, the last played frame, and the number of playout interruptions. This enables the decision-making module to learn about the impact of its decisions and adapt them accordingly. Finally, the client collects the network-related characteristics as experienced by the transport layer such as the packet loss, number of retransmissions, throughput, and RTT delay.

4 Experimental Evaluation

To evaluate the proposed approach, we consider an immersive point cloud video streaming use case. Below, we first discuss the evaluation scenario, the implementation, and the evaluation setup. Then, the obtained results are presented and discussed.

4.1 Evaluation Scenario

Volumetric video streaming allows evaluation of the performance of the proposed approach for applications with high-bandwidth and low-latency requirements. Here, the content is delivered on demand, i.e., the content is pre-captured and pre-encoded, and delivered when the client demands it.

The 8i data set was used to generate a volumetric video scene using point clouds [24]. The four objects are first compressed using MPEG's reference V-PCC encoder [25]. Each object is encoded at five quality representations, resulting in the bit rates presented in Table 3. Since the original dynamic objects only possess 300 frames (or ten seconds of video at 30 FPS) the resulting footage is played out 12 times, alternating between forward and backward movement as to smoothly change the location of the objects. This results in a more realistic video length of 120 seconds, or 2 minutes. The sequences are temporally split into one-second-long segments and made available on the server for the client to request on demand. As suggested in [23], to generate the scene, the four point cloud objects are placed on a 60° arc, looking inward. The camera is initially positioned at the center of the corresponding circle, moving closer to and back farther away from a point cloud object, before moving left or right toward the next one.

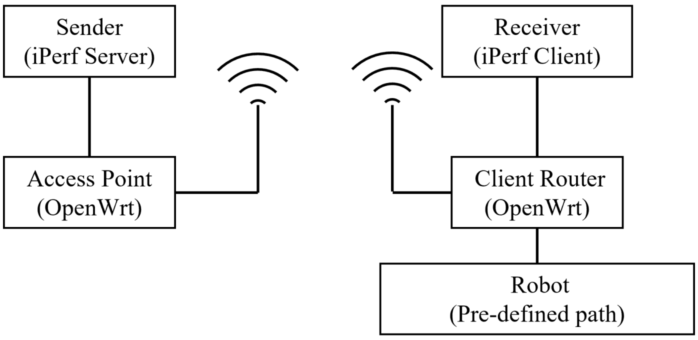


Fig. 7: An illustration of different components of the testbed used in trace collection.

4.2 Implementation

Here, we provide the details of the implementation of the client and server for the sake of evaluation. We have implemented a headless player, which can retrieve the immersive video content from the data sent by the server and simulate playback for the user. In order to emulate the viewport of the user, we have provided the player with a programmatically-generated unique camera trace of the user's FoV [26]. The trace consists of a user position and focus for each frame, based on the considered scene. To enable interactivity, the buffer size is fixed to one second. The client is configured to send the user context to the server multiple times per second. The context information in Table 2 is presented to the server. After each iteration of download, the player logs different parameters related to the downloaded content, such as the perceived throughput, download time, playout interruptions (freezes), and total freeze duration. The client is implemented as such that, should the buffer be empty, playout is stalled; in this case, the user will see the objects standing still rather than moving. The server is implemented as described in Section 3. The algorithm presented previously has been deployed to make decisions.

The transport layer of the client and the server depends on QPR [10] in order to realize the parallel existence of reliable and unreliable delivery mechanisms under the same connection. The application layer decisions are mapped to semantics of QPR in order to have the envisioned QAR on the transport layer. The client and the server are built using Cloudflare's QUIC (QUICHE) [27], which is an implementation of QUIC and provides support for `HTTP/3`. For the sake of comparison, we also added the support for a reliable variant of QUIC and QPR without any adaptation for both the player and the server.

4.3 mmWave Trace Collection

For the sake of the evaluation and to emulate mmWave networks, we collected throughput traces using commercial-off-the-shelf mmWave routers and a configurable robot. In this direction, we designed a testbed as illustrated in Figure 7. Firstly, the sender is a laptop hosting an iPerf [1] server. The sender is connected to a TP-Link Talon

[1] https://iperf.fr/

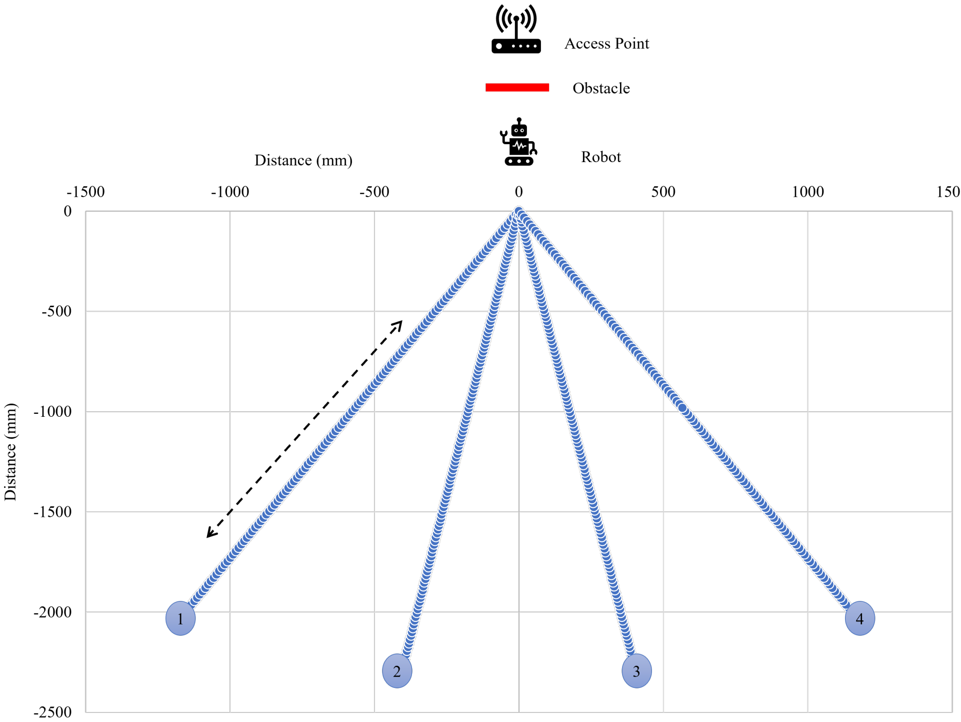


Fig. 8: An illustration of the path followed by the router and the placement of various components in testbed.

AD7200 mmWave router [2], which acts as an access point. On the client side, a second router is placed on top of a Rover Robotics 4WD Rover Pro robot [3]. Next, this router is connected to a laptop that hosts the iPerf client that acts as the receiver. The sender and receiver are connected to their respective routers using a gigabit Ethernet cable. Although routers are capable of achieving throughput beyond 1 Gbps, it is limited by the cable capacity on the current testbed. The routers are connected in a master-client network using the wireless link and run OpenWrt [1]. This version offers complete support for this router by improving performance in terms of CPU usage.

We have collected the throughput traces for two cases when there is a clear LoS and when it is broken using an obstacle. The traces were collected in the following manner:

- The path of the robot is designed in such a way that it emulates the scene considered while placing point cloud objects. As discussed previously, the four-point cloud objects are placed on a 60° arc, looking inward. As presented in Figure 8, 1, 2, 3, and 4 represent the point cloud placements in the scene. The same dimensions are

[2] https://www.tp-link.com/us/home-networking/wifi-router/ad7200/
[3] https://roverrobotics.com/products/4wd-rover-pro

considered while constructing the scene and the placement of the objects in this testbed.
- Next, the robot is programmed to follow the path illustrated in Figure 8. The robot moves close to the point cloud object and away from it at an approximate speed of 0.8 m/s.
- As mentioned previously, the robot hosts the client router, and the access point is placed at a distance of 1 m from the initial starting point of the robot. In order to induce obstruction to the LoS of the routers, cabinets are placed between the robot and the access point.
- As the robot traces the given path, the throughput experienced by the client while receiving data using the iPerf tool is logged every second.

As shown in Figure 9a, the throughput fluctuates even when there is a clear LoS between the receiver and the access point. This can be attributed to the beam misalignment caused by the constant motion of the robot. While this is resolved through beam forming, it is important to understand that the transmit beams of the router while being directional are imperfectly shaped. In such a case, they can also be affected by the way the room is designed. In the second case, the throughput deteriorates badly as the mobility is accompanied by loss of LoS. These two scenarios can perfectly depict a real-world 6DoF immersive media scenario which involves both mobility and the disturbance of LoS.

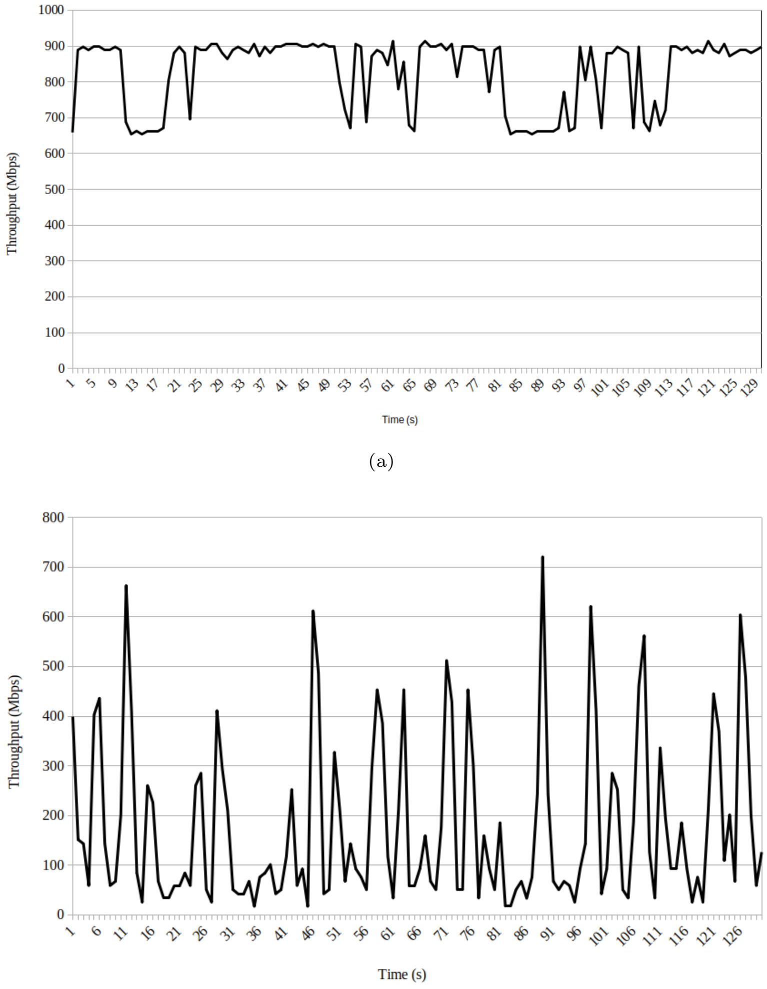
4.4 Evaluation Setup

To evaluate the impact of network impairments on the performance of the proposed framework, a network setup is emulated using MiniNet6 [28], where the player and server are implemented on individual virtual hosts (Figure 10). We have deployed traffic control (tc) to vary the link bandwidth, using the traces presented before. We have implemented the Mininet-based setup on a node (Hexacore Intel E5645 (2.4 GHz) CPU and 24 GB of RAM) hosted on imec's Virtual Wall infrastructure [29] as seen in Figure 10.

For every experiment, we have logged the throughput, the quality of each object per sending iteration, the number of freezes, and the total freeze duration. Each experiment for a combination of use case, delivery mechanism, and network configuration has been run for ten iterations, which is enough for statistical significance. We have compared the performance of the proposed framework against the reliable flavor of QUIC and the partially reliable flavor of QUIC. Henceforth, QPR refers to partially reliable QUIC, where half the data (top two ranked point cloud objects) are sent reliably and the rest unreliably.

4.5 Evaluation Results

First, we present the results for the scenario where there is no obstacle between the access point and the client's router, i.e., using trace 1. As presented in Figure 11, both QAR and QPR outperform the reliable variant of QUIC by almost achieving 28% more throughput. This performance of QPR is in line with what we observed in our previous work. However, despite sending more content reliably (see Figure 12), QAR's



(a)

(b)

Fig. 9: mmWave throughput traces collected when (a) there is a clear LoS, and (b) there is an obstacle.

performance is on par with QPR. In addition, all the content in all three cases is sent at the highest quality without any playout interruptions. This can be attributed to the availability of sufficient bandwidth to stream the content. These results confirm that all three protocols are suitable for streaming immersive multimedia on mmWave networks when there is a clear LoS. Depending upon the bandwidth and the reliability requirement of the application, an appropriate protocol can be deployed.

Next, we present the results for the case where trace 2 is deployed. Figure 13a and Table 4 present the delivered qualities and observed freezes when the context frequency is 1 Hz. As seen in the previous case, both QPR and QAR deliver higher-quality

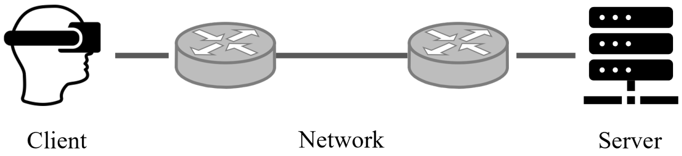


Fig. 10: Experimental setup, using Mininet to host a virtual network within a node on the Virtual Wall infrastructure.

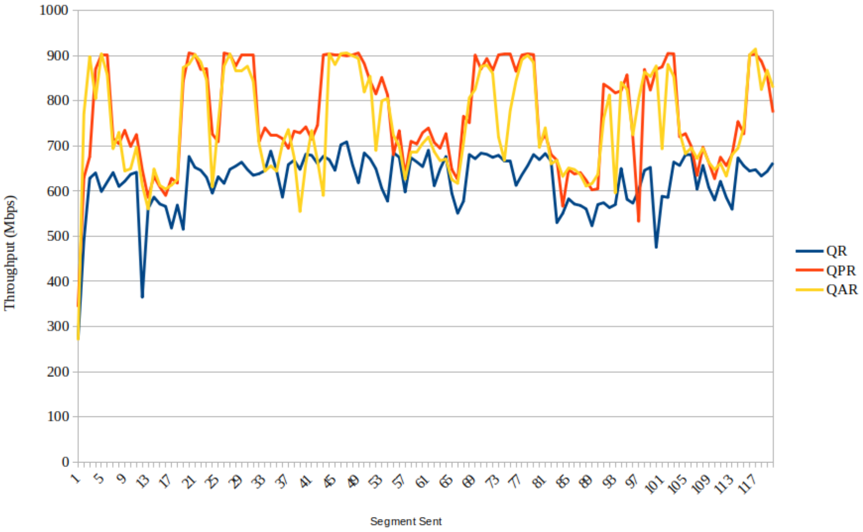


Fig. 11: Throughput for different variants of QUIC over sending iterations when trace 1 is deployed, when the context frequency equals 1 Hz (one out of ten iterations shown).

content compared to the reliable protocol. However, in all three cases, a high number of freezes are observed with a very high freeze duration. This can be attributed to the recency of the information available for the heuristic to take quality decisions. As the qualities are allotted based on the perceived throughput of the previous iteration and due to the high variations in the available throughput, the server could send more data than the available capacity, leading to additional freezes. However, in terms of the number of freezes and total freeze duration, QAR performs better than QPR, followed by reliable QUIC (QR).

Next, we have increased the context frequency to observe its impact on the quality, the number of freezes, and the freeze duration. Comparing Table 5 to Table 4, increasing the context frequency leads to a reduced number of freezes. For example, in the case of QAR, the number of freezes and the total freeze duration have reduced by more than 30% (Figure 13b). However, the content is delivered at a lower quality setting owing to informed decision-making of the algorithm. Moreover, in the case of QAR, 5% less objects in the viewport are sent at the highest quality (Figure 13a). The

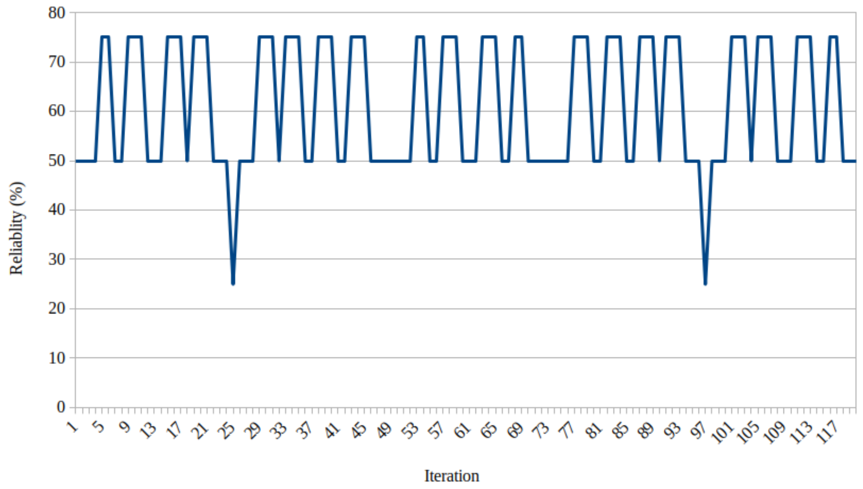


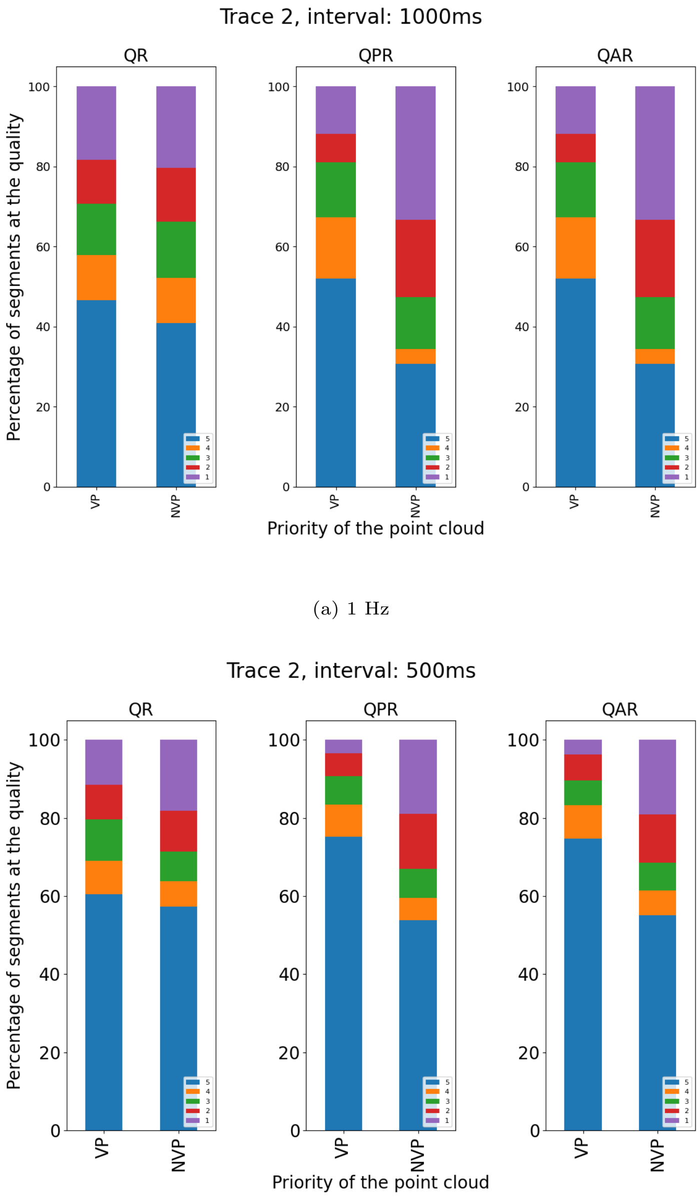
Fig. 12: Percentage reliability as adapted by QAR per iteration when the context frequency is 1 Hz (one out of then iterations shown)

Table 4: Observed freezes and total freeze duration (TFD) for trace 2 when the context frequency is 1 Hz.

Protocol	#Freezes (avg)	TFD (s)
QR	48±0.7	32.5±0.1
QPR	42±0.6	30.8±0.1
QAR	40±0.8	29.1±0.2

increased context frequency allows the algorithm to more accurately estimate that the network is saturated, and it thus needs to lower the quality to avoid freezes. However, given the reactive approach, this may not do away with the freezes completely. The most important gain using QAR can be seen in Figure 14a, where all the objects in the viewport are delivered reliably by QAR. This translates to 22.5% more objects being sent reliably compared to QPR. This becomes particularly important in scenarios like mmWave networks where deploying FEC to protect data is not feasible due to additional processing latency and very high overhead.

As presented in Figure 14b, as the context frequency increases, there is a clear improvement in terms of the number and the duration of freezes observed by the client. While QAR performs slightly better compared to other approaches, the number of freezes has reduced considerably for all the protocols. This underlines the importance of having more recent information for the server to take decisions. Hence, sending the context information more frequently forms an important part of the framework. While we see a clear improvement, there are still a considerable number of freezes. This is because the bandwidth information available in the context is that which is



(a) 1 Hz

(b) 2 Hz

Fig. 13: Observed qualities of point cloud objects per iteration when the context frequency equals, (a) 1 Hz, (b) 2 Hz; VP refers to “in viewport” and NVP refers to “not in viewport”.

Table 5: Observed freezes and total freeze duration (TFD) for trace 2 when the context frequency is 2 Hz.

Protocol	#Freezes (avg)	TFD (s)
QR	32±0.7	21.5±0.1
QPR	27±0.6	20.1±0.1
QAR	26±0.8	20±0.2

Table 6: Observed freezes and total freeze duration (TFD) for trace 2 when the context frequency is 1 Hz and there is perfect bandwidth prediction.

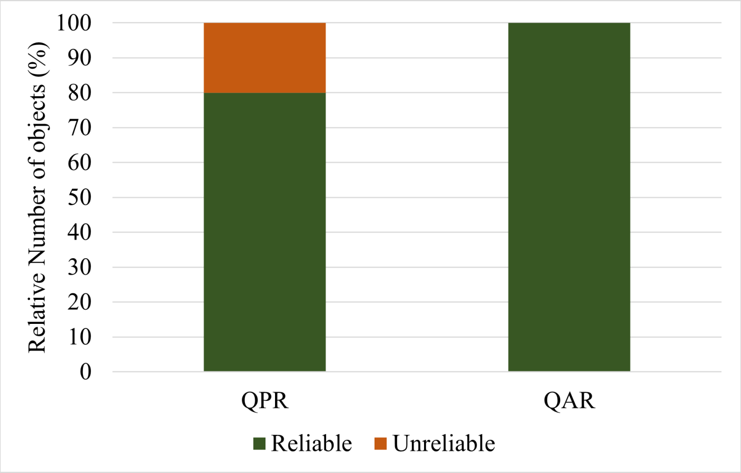
Protocol	#Freezes (avg)	TFD (s)
QR	8±0.63	1.054±0.01
QPR	2±0.00	0.176±0.00
QAR	2±0.00	0.160±0.00

experienced by the client in the previous iteration. Despite an increase in the frequency, the curve flattens for all protocols, because of highly dynamic changes in the available bandwidth which the decision-maker is not completely aware of. This is where the prediction component comes into the picture.

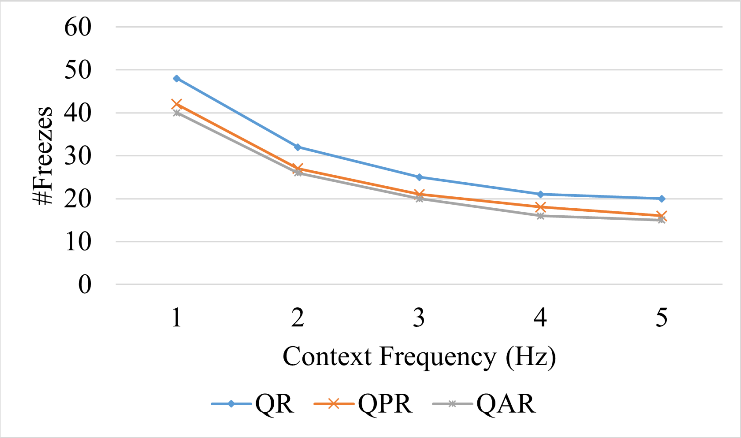
To understand the impact of the prediction component, we assume that the server is able to predict the available bandwidth and take decisions based on the predicted value instead of the value provided in the context information. To this end, we have provided the server with the throughput trace used in the experiments. As shown in Table 6, the number of freezes and the total freeze duration has come down drastically for both QPR and QAR. In the case of QR there is still a higher number of freezes which is not tolerable for 6DoF immersive media use cases. In addition, the freeze duration has also come down drastically, for instance, in the case of QAR it is 0.08 seconds per freeze which can go unnoticed in most situations.

4.6 Discussion and Use Cases

Our experimental evaluation has shown that, in the considered scenarios, mmWave networks support interactive VR applications without any issues as long as there is no blockage. Furthermore, the proposed mechanism where every object in the viewport mechanism is delivered reliably outperforms the reliable QUIC in terms of throughput. As long as the user viewport prediction is accurate, we argue that it is sufficient to deliver that part of the data reliably at higher quality. Even if the user changes their viewport, the client can reconstruct the scene outside of the viewport, as no initial data is lost. Next, when there is a blockage, it is essential to deliver the context as frequently as possible to reduce the number of freezes. When the context information is supplemented by the prediction mechanisms on the server, there will be almost no



(a) Number of objects in the viewport and their respective delivery mechanisms,



(b) Impact of context frequency on the number of freezes

Fig. 14: Number of objects in the viewport in comparison to its delivery type (a) and the impact of the frequency of the context of the number of freezes (b).

playout interruptions. The system adapts the quality based on the predicted bandwidth, which can be lowered when there is a blockage. It should be noted that the system makes the best use of the available bandwidth by prioritizing the content in the viewport. However, having a resilient link without interruptions is an open research challenge that can only be handled at the lower layers. As discussed previously, the approaches like multi-connectivity, beamforming and relay-aided communications can further complement the proposed system. A cross-layer approach can be deployed, where the network-level approaches aid the upper layer with relevant information to take higher-level decisions. Given that the proposed approach is implemented in user space, it is flexible to integrate the approach with lower layers. In addition, the upper

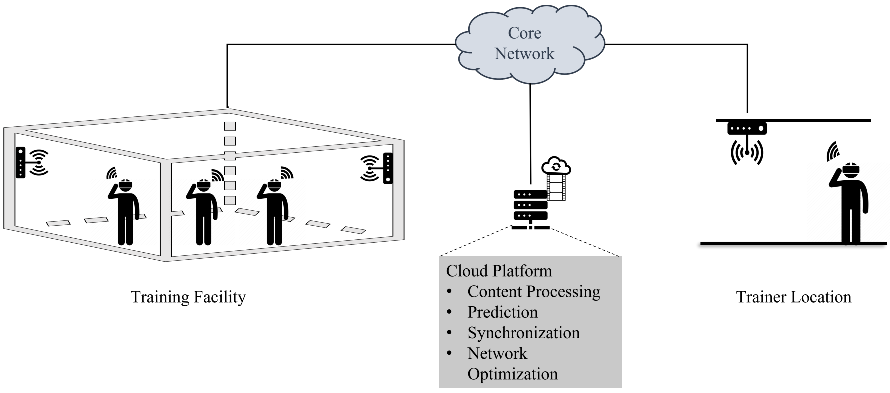


Fig. 15: Illustration of immersive training use case.

layer information like the user's viewport can be beneficial for the lower layers in case of optimizations like beamforming where the beam can be steered in the direction of predicted user movement. Finally, the historical context information can be used to develop and train the prediction models.

The proposed mechanism opens an avenue for various interactive VR use cases. They range from one-to-one to one-to-many to many-to-many. One-to-one refers to interactive communication between two users, such as an immersive teleconference. One-to-many refers to an interaction between two sites where there are multiple people on one side and one user on the other. An example of such a scenario is immersive training when an instructor trains multiple trainees across the network. Finally, many-to-many can be an immersive gaming scenario where the state needs to be synchronized across many users over the network. Figure 15, presents an example of an immersive training use case where different trainees work together at a training facility to learn a sequence of actions. Since there are all connected wirelessly, they can freely roam around and interact with the virtual environment and other users. This is expected to increase immersion and training efficiency. Furthermore, the trainees can be helped by an expert trainer from a remote location. In order to ensure that the state is synchronized amongst all the users and the right content is delivered to the user, the processing can be offloaded onto a cloud facility on the core network. The proposed framework can be deployed here to collect the relevant context information and take reliability and quality decisions accordingly.

5 Conclusion

In this paper, we propose a context-aware and reliable transport layer framework that delivers immersive media content on millimeter wave (mmWave) networks. To this end, we propose a client-server-based framework that is equipped with an adaptively reliable variant of QUIC at the transport layer. The client which plays the immersive media content for the user also collects the information related to the user, network, and scene under consideration and delivers it to the server periodically as context. The server, which is equipped with a decision-making heuristic along with a bandwidth prediction mechanism, uses the context information to adapt the delivery mechanism along with the quality of the content. We have implemented the client and server as per the defined mechanism and generated a scene using point cloud objects for the sake of evaluation. Furthermore, to emulate the mmWave networks, we have collected throughput traces using commercial-off-the-shelf mmWave routers and a robot to induce mobility. Our evaluation has shown that the mmWave networks support the seamless delivery of immersive media when there is no blockage. In the presence of blockage, the frequency of context information plays a key role in reducing the playout interruptions. Our proposed mechanism is able to deliver 22.5% more content in the viewport reliably without additional playout interruptions or quality changes. The playout interruptions have been reduced to almost zero when the server is equipped with perfect bandwidth prediction. This transport layer-based immersive media delivery framework can further enhance the delivery when complemented with lower-layer optimizations.

As part of future work, we will extend the case for multiple users and perform further analysis. Moreover, we will develop an algorithm to predict bandwidth in real time and integrate it with the prediction module. Finally, we will use the cross-layer approach discussed earlier, where the upper and lower layers exchange information to take collective decisions.

Acknowledgments. Part of this research was funded by the ICON project INTERACT, realized in collaboration with imec, with project support from Flanders Innovation and Entepreneurship (VLAIO). Project partners are imec, Rhinox, Pharrowtech, Dekimo and TEO. This research is partially funded by the FWO WaveVR project (Grant number: G034322N).

References

[1] Struye, J., Ravuri, H.K., Assasa, H., Fiandrino, C., Lemic, F., Widmer, J., Famaey, J., Torres Vega, M.: Opportunities and challenges for virtual reality streaming over millimeter-wave: An experimental analysis. In: 2022 13th International Conference on Network of the Future (NoF), pp. 1–5 (2022). IEEE

[2] Clemm, A., Torres Vega, M., Ravuri, H.K., Wauters, T., De Turck, F.: Toward truly immersive holographic-type communication: challenges and solutions. IEEE Communications Magazine **58**(1), 93–99 (2020)

[3] Chaccour, C., Soorki, M.N., Saad, W., Bennis, M., Popovski, P.: Can terahertz provide high-rate reliable low-latency communications for wireless vr? IEEE Internet of Things Journal **9**(12), 9712–9729 (2022) https://doi.org/10.1109/JIOT.2022.3142674

[4] Uwaechia, A.N., Mahyuddin, N.M.: A comprehensive survey on millimeter wave communications for fifth-generation wireless networks: Feasibility and challenges. IEEE Access **8**, 62367–62414 (2020)

[5] Nor, A.M., Halunga, S., Fratu, O.: Survey on positioning information assisted mmwave beamforming training. Ad Hoc Networks **135**, 102947 (2022)

[6] Ravuri, H.K., Struye, J., Torres Vega, M., Hooft, J., Famaey, J., Wauters, T., De Turck, F.: Streaming 8k video over millimeter wave networks: An experimental demonstrator. In: 2022 13th International Conference on Network of the Future (NoF), pp. 1–3 (2022). IEEE

[7] Ren, Y., Yang, W., Zhou, X., Chen, H., Liu, B.: A survey on TCP over mmWave. Computer Communications **171**, 80–88 (2021)

[8] Tarafder, P., Choi, W.: Mac protocols for mmwave communication: A comparative survey. Sensors **22**(10), 3853 (2022)

[9] Jung, J., An, D.: Access latency reduction in the quic protocol based on communication history. Electronics **8**(10), 1204 (2019)

[10] Ravuri, H.K., Torres Vega, M., van der Hooft, J., Wauters, T., De Turck, F.: Partially reliable transport layer for QUICker interactive immersive media delivery. In: Proceedings of the 1st Workshop on Interactive eXtended Reality, pp. 41–49 (2022)

[11] Iyengar, J., Thomson, M.: RFC 9000 QUIC: A UDP-based multiplexed and secure transport. Omtermet Emgomeeromg Task Force (2021)

[12] Li, J., Niu, Y., Wu, H., Ai, B., Chen, S., Feng, Z., Zhong, Z., Wang, N.: Mobility support for millimeter wave communications: Opportunities and challenges. IEEE Communications Surveys & Tutorials (2022)

[13] Cao, Y., Lv, T., Ni, W.: Intelligent reflecting surface aided multi-user mmwave communications for coverage enhancement. In: 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–6 (2020). https://doi.org/10.1109/PIMRC48278.2020.9217160

[14] Hayes, D.A., Ros, D., Alay, Ö., Teymoori, P., Vister, T.M.: Investigating predictive model-based control to achieve reliable consistent multipath mmwave communication. Computer Communications **194**, 29–43 (2022)

[15] Chakareski, J., Khan, M., Ropitault, T., Blandino, S.: Millimeter wave and free-space-optics for future dual-connectivity 6dof mobile multi-user vr streaming. ACM Transactions on Multimedia Computing, Communications and Applications **19**(2), 1–25 (2023)

[16] Zhang, D., Han, B., Pathak, P., Wang, H.: Innovating multi-user volumetric video streaming through cross-layer design. In: Proceedings of the Twentieth ACM Workshop on Hot Topics in Networks, pp. 16–22 (2021)

[17] Struye, J., Lemic, F., Famaey, J.: Covrage: Millimeter-wave beamforming for mobile interactive virtual reality. IEEE Transactions on Wireless Communications (2022)

[18] Wu, H., Caso, G., Ferlin, S., Alay, Ö., Brunstrom, A.: Multipath scheduling for 5g networks: Evaluation and outlook. IEEE Communications Magazine **59**(4), 44–50 (2021)

[19] Haile, H., Grinnemo, K.-J., Hurtig, P., Brunstrom, A.: Rbbr: a receiver-driven bbr in quic for low-latency in cellular networks. IEEE Access **10**, 18707–18719 (2022)

[20] Michel, F., Cohen, A., Malak, D., De Coninck, Q., Médard, M., Bonaventure, O.: FlEC: enhancing QUIC with application-tailored reliability mechanisms. IEEE/ACM Transactions on Networking (2022)

[21] Li, J., Zhang, C., Liu, Z., Hong, R., Hu, H.: Optimal volumetric video streaming with hybrid saliency based tiling. IEEE Transactions on Multimedia **early access**, 1–1 (2022)

[22] Zverev, M., Garrido, P., Fernandez, F., Bilbao, J., Alay, Ö., Ferlin, S., Brunstrom, A., Agüero, R.: Robust QUIC: integrating practical coding in a low latency transport protocol. IEEE Access **9**, 138225–138244 (2021)

[23] van der Hooft, J., Wauters, T., De Turck, F., Timmerer, C., Hellwagner, H.: Towards 6DOF HTTP adaptive streaming through point cloud compression. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 2405–2413 (2019)

[24] d'Eon, E., Harrison, B., Myers, T., Chou, P.A.: 8i voxelized full bodies-a voxelized point cloud dataset. ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006 **7**, 8 (2017)

[25] Schwarz, S., Preda, M., Baroncini, V., Budagavi, M., Cesar, P., Chou, P.A., Cohen, R.A., Krivokuća, M., Lasserre, S., Li, Z., *et al.*: Emerging MPEG standards for point cloud compression. IEEE Journal on Emerging and Selected Topics in Circuits and Systems **9**(1), 133–148 (2018)

[26] Ravuri, H.K., Torres Vega, M., van der Hooft, J.D., Wauters, T., De Turck, F.: Adaptive partially reliable delivery of immersive media over quic-http/3. IEEE Access **11**, 38094–38111 (2023) https://doi.org/10.1109/ACCESS.2023.3268008

[27] Marx, R., Herbots, J., Lamotte, W., Quax, P.: Same standards, different decisions: a study of QUIC and HTTP/3 implementation diversity. In: Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC, pp. 14–20 (2020)

[28] Kaur, K., Singh, J., Ghumman, N.S.: Mininet as software defined networking testing platform. In: International Conference on Communication, Computing & Systems (ICCCS), pp. 139–42 (2014)

[29] IDLabt ilab.t: https://doc.ilabt.imec.be/ilabt/virtualwall/ (2023). https://doc.ilabt.imec.be/ilabt/virtualwall/