An Innovative Algorithm for Improved Quality Multipath Delivery of Virtual Reality Content

Fabio Silva
Performance Engineering Lab
School of Electronic Engineering
Dublin City University
Dublin, Ireland
fabio.silva3@mail.dcu.ie

Mohammed Amine Togou
Performance Engineering Lab
School of Electronic Engineering
Dublin City University
Dublin, Ireland
mohammedamine.togou@dcu.ie

Gabriel-Miro Muntean
Performance Engineering Lab
School of Electronic Engineering
Dublin City University
Dublin, Ireland
gabriel.muntean@dcu.ie

***Abstract*—This paper describes and evaluates an Innovative Algorithm for Improved Quality Multipath Delivery of Virtual Reality Content (QM4VR) that addresses the stringent communication requirements of Virtual Reality (VR) applications. Making use of the Multipath TCP (MPTCP) built-in multipath delivery features (subflows), QM4VR explores the subflows' characteristics, evaluates their performance (e.g., delay, throughput or loss) and proposes a new management scheme to improve the Quality of Service (QoS) of the VR applications. glsqm4vr adopts a Machine Learning (ML)-based approach to evaluate the subflows' performance which is implemented in two steps: 1) a linear regression scheme to forecast the subflow's performance for a given feature; and 2) a linear classification scheme to arrange the results obtained in step 1. Based on these results QM4VR selects the most appropriate subflows for data delivery in order to achieve improvement of VR QoS levels.**

***Index Terms*—Multimedia networking, VR, MPTCP, QoS, prioritised content delivery**

I. Introduction

Today, VR technology has expanded its spectrum of applicability from the gaming and entertainment industries to include areas such as education [1], medical training [2], military [3] and its integration with real world applications [4]. As a result, its market value has increased significantly. [5] suggests an international market of US$150 billion by 2020 while [6] shows the evolution in the number of users in the US market from 21.9 million in 2017 to 45.3 million in 2019 and is expected to reach 71.3 million by 2022. For the European market alone, [7] estimates a total production expansion between €15 and €34 billion by 2020.

Yet, as the significance of the VR technology grows, so does the concerns about its applications' increasing demands, e.g., ultra-high quality video [8] and sophisticated 3D audio technologies [9]. For instance, [10], [11] describe the bandwidth and Round-Trip Time (RTT) demands that far exceed the limits of upcoming 5th generation (5G) networks [12]. Additionally,

This work was supported, in part, by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 688503 and, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

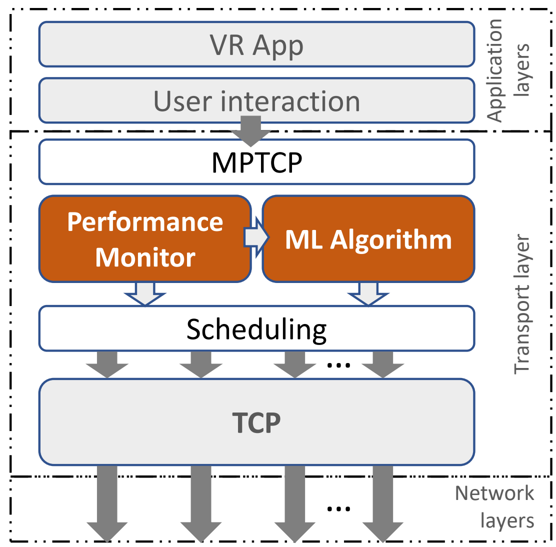

Figure 1: QM4VR architecture

a Cisco® report [13] estimates a 12-fold growth in network traffic of VR content between 2017 and 2022. Although such stringent demands can be overestimated, at least in the short-term, it is important to emphasise how challenging it will be for the infrastructure to support the VR operation.

This paper proposes an Innovative Algorithm for Improved Quality Multipath Delivery of Virtual Reality Content (QM4VR) to enhance the transmission of VR content through exploring the multipath inherent characteristics of the MPTCP subflows [14]. MPTCP goes beyond the Transmission Control Protocol (TCP) and offers the capability of establishing a set of TCP connections amongst peers using multiple paths simultaneously. MPTCP operates transparently between the application and network layers and offers the same type of services as the regular TCP. By introducing a new approach to the management of MPTCP subflows, QM4VR offers a content-aware distribution of prioritised packets to improve the QoS of VR applications. For example, VR services sensitive to delay (e.g., motion tracking information) should be redirected to subflows with shorter delays while other services such as movement detection video which require higher throughput

978-1-7281-2150-5/19/$31.00

should be redirected to subflows with higher throughput. To achieve this, QM4VR keeps track of the subflows operations and performance, and alters the load-balance/scheduling scheme to improve the QoS parameters. Figure 1 shows the algorithm's basic architecture. A full description is available in Section III. The rest of this paper is structured as follows. Section II surveys some related work. Section III describes the QM4VR's architecture. Section IV outlines the testbed scenario along with the simulation results. Finally, Section V concludes the paper.

II. Related works

The goal of QM4VR is to evaluate the subflows available on MPTCP subflow pool and prioritise the traffic of specific VR services according to their distinct demands. In this context, we first survey the schemes that were proposed to enhance the QoS of VR services. Then, we describe few schemes that deployed MPTCP to improve the performance of various applications.

A. VR Services

Improving QoS of typical multimedia services (i.e., audio and video) has been extensively studied in the literature. However, very few studies have focused on improving the QoS of VR services solely. In [8], the authors focused on bandwidth to improve VR video services delivery while the authors in [11] examined ways to reduce the transmission delay to promote the VR experience. Nevertheless, these efforts suggest that VR requirements far exceed the expected specification of the next-generation 5G networks [12]. Although network technologies have shown that *"less-than-stellar"* configurations were enough to support other technologies, VR unquestionably presents a challenge to the network infrastructure.

Given the relevance and projected growth attributed to the VR technology, standardisation initiatives carried out by renowned and accredited technical associations started addressing the problem. In May 2017, the Institute for Electrical and Electronic Engineers (IEEE) [15] and Video Electronics Standards Association (VESA) [16] formed special groups to focus on the development of VR standards. Still, solutions to address these demands on the infrastructure level have been proposed. Specific edge computing architectures, caching techniques and proper resource allocation [8], [17] can distribute VR content to reduce latency and improve bandwidth content delivery. Besides, the network performance rises concerns about the eventual side effects of Cybersickness [18] which can be aggravated by poor network performance and jeopardise the whole VR experience [18], [19]. Consequently, the work presented in this paper aims at improving the VR content delivery at both the network and transport layers in order to mitigate these adverse challenges.

B. MPTCP

MPTCP enables data transport over multiple concurrent paths, called subflows [14], behaving as regular TCP sessions at the transport layer of the Open Systems Interconnection (OSI) and, therefore, keeping network compatibility. By not altering how the communication between OSI layers is established, MPTCP can be seamlessly integrated into the OSI model.

The fact that MPTCP complies with the premise of transparency to the OSI upper layers (application, presentation and session) and the lower layers (network, data link and physical) makes it adherent to the TCP/IP standard and grants it a *virtual standardisation* when it comes to its use and implementation. In addition, the Internet Engineering Task Force (IETF) [14] developed a comprehensive work addressing MPTCP. These works cover architectural guidelines for MPTCP development, congestion control and use cases in real networks.

Since then, several studies have examined the use of MPTCP to improve the performance of a variety of applications. For instance, the authors in [20] presented an MPTCP multimedia content transport that explores the interaction between the transport and application layers to improve video delivery. In [21], a network performance enhancement using a Redundant MPTCP (rMPTCP) packet scheduler targeting optimal failover time and smoother latency variations was developed. The authors in [22] demonstrated the benefits achieved by improving the management of the subflow communication while. the work in [23] demonstrates the benefits of an improved MPTCP congestion control. Finally, the work in [24] introduces the concept of computer-generated congestion control capable of adjusting itself to the network environment - instead of a fixed policy. This approach is similar to the QM4VR algorithm, although they work at different levels. Whilst [24] proposal works at a lower level generating congestion control algorithms at endpoints, the QM4VR works at a higher level evaluating the subflows' performance and adapting the load balance according to the needs of VR services.

III. QM4VR Solution

The QM4VR algorithm provides an enhanced management of the MPTCP subflow pool and increases the transmission efficiency of VR content. By doing so, QM4VR is able to redirect or prioritise VR components based on specific demands and/or relative sizes. Figure 2 illustrates the main VR components.

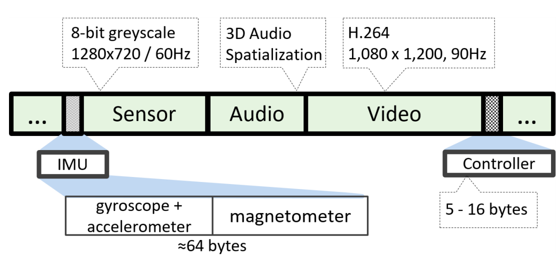

Figure 2: VR data streaming

The Sensor component information are images or videos mapping the Head-mounted Display (HMD) and controllers/joysticks through the infrared light used to track their position and movement. The Inertial Measuring Unit (IMU) component is responsible for the "fine-grained" movement detection and composed by 3 types of devices: accelerometers ($\frac{m}{s^2}.10^{-4}$), gyroscopes ($\frac{\theta}{s}.10^{-4}$ in radians) and magnetometers (10^{-4} gauss) [25]. These components are responsible for VR tracking with high accuracy. The controllers contain also user interaction information (pressed buttons, touchpads, etc). Finally, the video streaming and the increasingly complex audio features [9] summarises the most common VR components.

The interdependency between these components is responsible for the *motion-to-photon* latency (i.e., the time between user interaction and its results in the VR experience). Figure 3 illustrates how the relationship between these components works [26]. Camera samples and IMU data are combined to establish a pose estimation (motion) necessary for the human interaction in a VR environment. This information - in combination with the controller data - are then sent to be rendered by the application, which returns a group of frames to be shown (photon) in the display of the Head-Mounted Display (HMD).

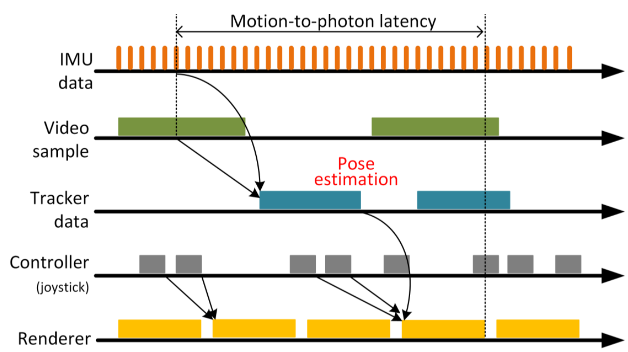

Figure 3: Motion-to-photon latency

This *latency budget* [8] is of utmost importance for the VR experience and it should be addressed to minimise *motion-to-photon* latency and improve the *impression of presence*, i.e., the sensation of being immersed in the experience.

A. Algorithm modules

To address the *motion-to-photon* latency, the QM4VR modules are presented in Figure 4 and are organised in two groups: the *Content-aware Arbiter* and the *Subflow Monitor*. The algorithm performs its functions in the transport layer of the OSI model and sits on the top of the MPTCP protocol.

The modules in the Subflow Monitor group are:

- **Feature extraction**: gathers information about subflows' operation by monitoring a packet transmission (e.g., latency value, packet loss, etc).
- **Performance History Buffer (PHB)**: tracks subflows' operations and executes forecast performance-related calculations for each subflow feature.
- **Feature classifier**: classifies the information available in the performance history buffer based on the linear classifier described in subsection III-B2.

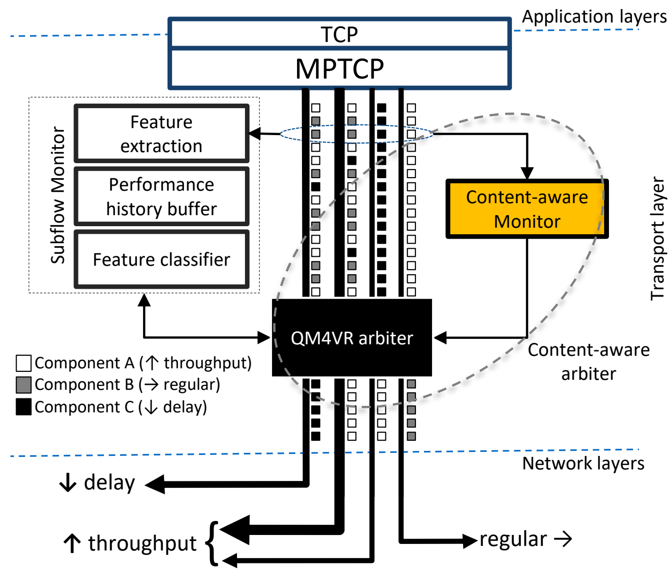

Figure 4: QM4VR diagram

The modules in the Content-aware Arbiter group are as follows :

- **Content-aware monitor**: once a specific packet content is found (e.g., checking the packet header for a IMU data), this monitor activates the arbiter module.
- **QM4VR arbiter**: fetches the performance report from Feature classifier module and, based on the feature-classified report, the arbiter chooses the best subflow to transport a specific component.

B. Machine learning approach

QM4VR performs two types of analysis to *forecast-like* performance (linear regression) and classify the subflows (linear classifier) according to its intended usage.

1) Linear regression: For the sake of a light-weight algorithm, this work extends a previous research [22], [27] and simplifies the linear regression applied to PHB report to extract *only* the slope of the linear regression for any given subflow feature performance (delay, throughout, etc.).

Equation 1 describes how the slope is obtained (i.e., in this case, it is the covariance of x and y divided by the sum of squares (variance) of x). For this project, the x variable is a timestamp for the exact moment a packet is transmitted and the y variable is the value of a specific network feature (e.g., latency or throughput) for that transmission. These variables compose the dataset used in the linear regression forecasts.

$$slope = \left(\sum_{i=1}^{n} (x_i - \overline{x}).(y_i - \overline{y}) \right) / \sum_{i=1}^{n} (x_i - \overline{x})^2 \quad (1)$$

Figure 5 shows two hypothetical examples and the expected results represent the significance of the linear relationship between x and y. Algorithm 1 calculates the slope for a specific subflow feature from the history buffer.

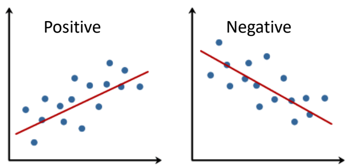

Figure 5: Hypothetical slopes

Algorithm 1: Linear regression/slope calculation.

```
Result: Returns the linear regression slope of a given
        subflow feature.
Input:  V_h ← subflow feature history performance
slope ← NaN;
if V_h.size > 0 then
    while V_h.size > limit do
        V_h.removeFirst;
    end
    slope, n, num, den = 0;
    sum_x, sum_y, x̄, ȳ = 0;
    foreach feature in V_h do
        n++;
        sum_x += feature.time;
        sum_y += feature.value;
    end
    x̄ = (sum_x / n);
    ȳ = (sum_y / n);
    foreach obj in V_h do
        num =+ (feature.time - x̄) . (feature.value - ȳ);
        den =+ pow(feature.time - x̄));
    end
    slope = (num / den);
end
return slope;
```

2) Linear Classifier: Once each subflow has its features properly evaluated by the linear regression, a classification scheme must address the subflow pool and identify the most suitable subflows for a specific usage.

To accomplish this task, QM4VR analyses each subflow by applying a linear classification approach commonly used for neural networks. Figure 6 presents the subflow as a neuron in a neural network.

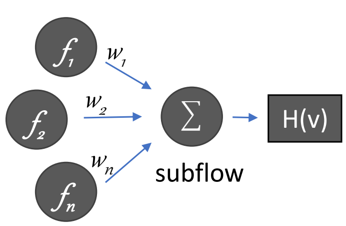

Figure 6: Subflow as a neuron

In this computational model, the features of the subflow (i.e. delay, jitter) are represented by the vector $\vec{f}$. Each feature can have a specific *relevance* represented by the weight vector $\vec{w}$. These vectors are combined by the sum of the bias b and the *dot product* (weighted sum) of the inputs $\vec{f}$ and weights $\vec{w}$, as described in Equation 2:

$$\vec{w}.\vec{f} + b = (\sum_{i=1}^{n} w_i.f_i) + b \tag{2}$$

Finally, an activation function, $H(v)$ (Equation 3), computes the weighted sum into a single output and defines if the neuron is fired or not.

$$H(v) = H(\vec{w}.\vec{f} + b) \tag{3}$$

Usually, for this application, $H(v)$ is a Heaviside step function as shown in Figure 7.

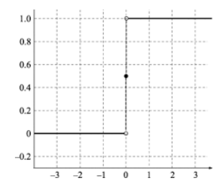

Figure 7: Hypothetical Heaviside step function

Equation 4 defines when the function $H(\vec{w} \cdot \vec{f} + b)$ fires:

$$H(v) = \begin{cases} 1, & \text{if } v >= 0 \\ 0, & \text{if } v < 0 \end{cases} \tag{4}$$

This Heaviside *behaviour* is obtained by the *Sigmoid* function shown in Equation 5.

$$H(v) = \frac{1}{1 + e^{-v}} \tag{5}$$

This computational model is usually identified as a **linear classifier** for the trigger boundary and is based on a linear combination of inputs. Algorithm 2 details the weighted sum calculation and the Heaviside/Sigmoid function usage.

3) Sorting process: The sorting process uses the results of the linear regression and linear classification to identify the best subflow for a specific requirement following the functional rules (conditions to be observed) and classificatory rule (network feature-classified). Equation 6 shows how the functional and classificatory rules are defined.

$$S_i = \begin{cases} -1, & \text{for } S_{cw} < P_s \\ -1, & \text{for } \theta_S > 0 \\ \min\limits_{s_1 \ldots s_n} (\vec{S}_{pool}), & Otherwise \end{cases} \tag{6}$$

Where S_{pool} is the subflow pool, S_i is the subflow ID to be used, S_{cw} is the subflow available congestion window, P_s is the packet size and θ_S is the average time for packet transmission in a given subflow.

There are two functional rules presented in Equation 6:

- $S_{cw} < P_s$ checks for the available congestion window for the transmission of a given packet.

Algorithm 2: Linear classifier.

```
Result: Returns the weighted sum (ws) of the inputs f⃗
        and weights w⃗
SetKwRepeatDodowhileInput: inputs ← f⃗;
weights ← w⃗;
b ← bias;
result ← NaN;
if inputs.size > 0 && weights > 0 then
    ws = 0;
    limit = inputs.size;
    while index > limit do
        ws += inputs[i] . weights[i];
    end
    ws += ws + b;
    result = 1 / (1 + (1/pow(e, ws)));
end
return result;
```

- $\theta_S > 0$ checks if a subflow has been operational before considering its historical performance (assuming that an operational subflow presents average package transmission time bigger than zero).

The classificatory rule is based on the results of the linear classifier (weighted sum) and considers the result of the Sigmoid function to sort/order the subflow properly.

IV. Testbed

QM4VR is assessed using Network Simulator 3 (NS-3) and MPTCP [28] implementation of the IETF Request for Comments (RFC) 6824 [14].

The testbed settings cover two common scenarios for VR: Long Term Evolution (LTE) and a IEEE 802.11 (Wi-Fi Alliance) (Wi-Fi) implementations. These scenarios take into consideration conditions that will represent "stress test" situations for the QM4VR algorithm (e.g., high noise networks, different subflow delay performance, uneven subflow throughput and packet loss). A modified MpTcpBulkSender application is set in order to generate the packets necessary for the tests. These packets are modified to represent the expected simulated traffic for each VR component under scrutiny. Its counterpart, MpTcpPacketSink application is also used and both applications are extensions of the NS-3 original applications [28]. Table I summarises the testbed environment setup.

A. QM4VR *Algorithm Assessment*

The aforementioned simulation environment is implemented to have high fluctuation in the background noise/traffic to validate the adaptation capabilities of QM4VR. Table II outlines the different scenarios and configurations used to assess QM4VR. These scenarios consider the combination of different weights for network features under scrutiny to define the relevance and impact of these features on the QoS of VR applications. The motivation behind the definition of these weights sought to explore the impact of QM4VR

Table I: Simulation setup summary

Parameter	Value
Environment	NS-3 open source MPTCP [28]
Simulation length	2000 seconds
Number of nodes	8 Nodes
Delay	adjustable to stress conditions
Prioritisation ratio	1/500
Sender app	MpTcpBulkSender [28]
Receiver app	MpTcpPacketSink [28]
Subflows Data Rate	1Mbps (LTE)
	512Kbps (Wi-Fi)
	512Kbps (Wi-Fi)

Table II: Scenarios and weighted features

Scenario	latency	throughput
t1	20%	80%
t2	40%	60%
t3	60%	40%

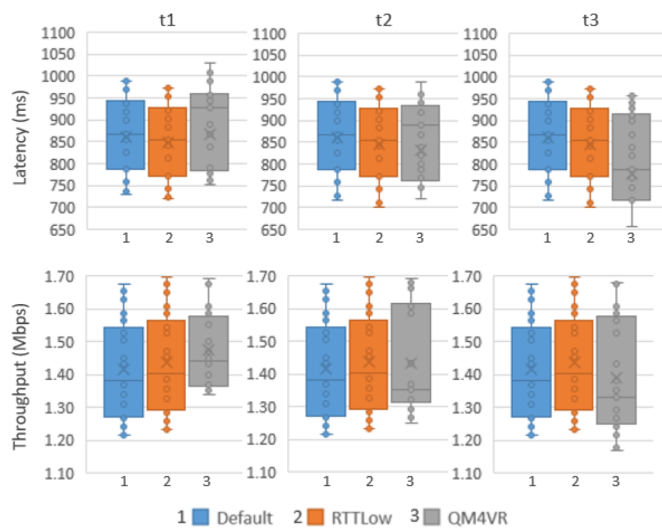

Figure 8: Results for use cases t1, t2, t3.

when improving specific VR dataflows (e.g., tracking and interaction) in the VR environment and, for that matter, QM4VR targets the improvement of the latency (decreasing delay) and throughput (increasing overall combined MPTCP performance).

For assessment purposes, QM4VR algorithm is compared to the following two algorithms: Default - a round-robin approach based on MPTCP (employs the default MPTCP NS-3 implementation) and RTTLow - a RTT-aware MPTCP enhancement (focuses on achieving a low RTT average value [22]). The graphs in Figure 8 depict the results for use cases *t1*, *t2* and *t3* configured as described in Table II.

In use case *t1* (20% focus on latency and 80% focus on throughput), QM4VR offered no significant reductions of latency when compared to the Default MPTCP algorithm and it is outperformed the RTTLow with about 2%. Referring to throughput, QM4VR outperformed the Default algorithm with 4.6% and the RTTLow with 2.8%. In use case *t2* (40% focus on latency and 60% focus on throughput), QM4VR outperformed the Default algorithm with 3.4% and the RTTLow with roughly 1.7%. In terms of throughput, QM4VR outperformed the Default algorithm with 1.3% and the RTTLow with roughly the same value. Finally, in use case *t3* (60% focus on latency and 40% focus on throughput), QM4VR outperformed the Default MPTCP algorithm with 9.6% and the RTTLow with roughly 8.0%. In terms of throughput, QM4VR offered no significant increase when compared to Default algorithm, but outperformed by RTTLow with approximately 3.3%.

In conclusion it can be noted how when considering both latency and throughput, the proposed solution has a balanced behaviour in comparison with the other solutions which favour one or the other of the performance metrics.

V. Conclusions

The increasing use and growing popularity of VR technology fuel a significant VR adoption and corresponding market growth. Nevertheless, due to VR stringent demands, this technology will pose serious challenges to the telecom infrastructure. To address these challenges, we propose the Innovative Algorithm for Improved Quality Multipath Delivery of Virtual Reality Content (QM4VR), a prioritisation algorithm to improve the use and management of the Multipath TCP (MPTCP) protocol and its intrinsic multipath characteristics (subflows).

QM4VR is capable to adapt to the network conditions' fluctuation, predict the trend of each network feature of the different subflows and offer better performance than the "*static*" load-balancing/scheduling schemes tested. The results show that a combination of features - as the latency and throughput explored in the simulation - can be improved as QM4VR offered latency reductions up to 9.6% and throughput increments up to 4.3%.

As future work, we will propose a correlation scheme associating the QoS improvements offered by QM4VR to the Quality of Experience (QoE) of VR applications and an adaptive scheme to apply the QM4VR algorithm to automatically balance MPTCP's network features to specific application needs.

References

[1] D. Bogusevschi, M. Bratu, I. Ghergulescu, C. H. Muntean, and G.-M. Muntean, "Primary School STEM Education: Using 3D Computer-based Virtual Reality and Experimental Laboratory Simulation in a Physics Case Study," *Innovative Pedagogies for Effective Technology-Enhanced Learning (IPETeL) Workshop*, p. 5, 2018. [Online]. Available: http://www.virtlab.com/

[2] T. Mazuryk and M. Gervautz, "Virtual Reality History, Applications, Technology and Future," *Digital Outcasts*, vol. 63, no. ISlE, pp. 92–98, 2013. [Online]. Available: http://bit.ly/2T6pBI8

[3] R. T. Azuma, "A survey of augmented reality," *Presence by MIT*, vol. 6, no. 4, pp. 355–385, 1997. [Online]. Available: http://bit.ly/2NBMcKJ

[4] A. A. Simiscuka and G. M. Muntean, "Synchronisation between Real and Virtual-World Devices in a VR-IoT Environment," *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB*, vol. 2018-June, pp. 1–5, 2018.

[5] H. Bellini, W. Chen, M. Sugiyama, M. Shin, S. Alam, and D. Takayama, "Virtual & Augmented Reality - Understanding the race for the next computing platform," Goldman Sachs Global Investiment Research, USA, Tech. Rep., 2016.

[6] M. Boland, "VR Usage & Consumer Attitudes," 2019. [Online]. Available: http://bit.ly/2Hsx54m

[7] E. Bezegova, M. A. Ledgard, R.-J. Molemaker, B. P. Oberc, and A. Vigkos, "Virtual Reality and its Potential for Europe," Ecorys, Tech. Rep., 2017.

[8] L. Han and K. Smith, "Transport Support for Augmented and Virtual Reality Applications," pp. 1–21, 2017.

[9] M. Johansson and E. Guy, "VR for Your Ears," *IEEE Spectrum*, vol. 02, pp. 24–29, 2019.

[10] B. Begole, "Why The Internet Pipes Will Burst When Virtual Reality Takes Off," pp. 1–4, 2016. [Online]. Available: https://goo.gl/tA8Qn7

[11] C. Westphal, "Challenges in Networking to Support Augmented Reality and Virtual Reality," in *IEEE ICNC*. San Jose, CA: IEEE, 2017. [Online]. Available: https://goo.gl/1x5DKa

[12] NGMN Alliance, "NGMN 5G White Paper," *Next Generation Mobile Networks, White paper*, pp. 1–125, 2015. [Online]. Available: http://bit.ly/2tkosF7

[13] Cisco VNI, "Cisco Visual Networking Index : Forecast and Trends, 2017 – 2022," Cisco Systems, Inc., Tech. Rep., 2015. [Online]. Available: http://bit.ly/2XdLc3H

[14] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses Abstract TCP/IP - RFC6824," Internet Engineering Task Force (IETF), Tech. Rep., 2013.

[15] IEEE-SA, "IEEE Announces Virtual Reality (VR) and Augmented Reality (AR) Standards Projects In Advance of Participation at Augmented World Expo," 2017. [Online]. Available: https://goo.gl/9zRvGG

[16] VESA, "VESA Forms Special Interest Group Focused on Emergent Virtual and Augmented Reality Markets," 2017. [Online]. Available: https://goo.gl/q2JSZ4

[17] P. Rost, V. Jungnicke, and 5G PPP Architecture Working Group, "View on 5G Architecture," 5G PPP Architecture Working Group, Tech. Rep. July, 2016.

[18] J. Barrett, "Side effects of virtual environments: A review of the literature," *Information Sciences Laboratory*, p. 60, 2004. [Online]. Available: http://bit.ly/2DjPEWC

[19] Oculus, "Health & safety," Facebook Oculus VR, Tech. Rep., 2017.

[20] X. Corbillon, R. Aparicio-Pardo, N. Kuhn, G. Texier, and G. Simon, "Cross-layer scheduler for video streaming over MPTCP," *Proceedings of the 7th International Conference on Multimedia Systems - MMSys '16*, pp. 1–12, 2016. [Online]. Available: http://bit.ly/2VZfZ3y

[21] A. Hunger and P. A. Klein, "Equalizing latency peaks using a redundant multipath-TCP scheme," *International Conference on Information Networking*, vol. 2016-March, pp. 184–189, 2016.

[22] F. Silva, D. Bogusevschi, and G.-M. Muntean, "An MPTCP-based RTT-aware Packet Delivery Prioritisation Algorithm in AR/VR Scenarios," in *IEEE IWCMC 2018 - 14th International Wireless Communications and Mobile Computing Conference*. Limassol, Cyprus: IEEE Xplore, 2018.

[23] C. Xu, S. Member, J. Zhao, and G.-m. Muntean, "Congestion Control Design for Multipath Transport Protocols : A Survey," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, pp. 2948–2969, 2016.

[24] K. Winstein and H. Balakrishnan, "TCP ex Machina: Computer-Generated Congestion Control," *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM - SIGCOMM '13*, 2013.

[25] N. Patel, "DK2 Firmware Specification," Facebook Oculus VR, Tech. Rep. April, 2014.

[26] D. Wagner, "Motion-to-photon Latency in Mobile AR and VR," 2018. [Online]. Available: https://daqri.com/blog/motion-to-photon-latency/

[27] F. Silva, D. Bogusevschi, and G.-M. Muntean, "Innovative Algorithms for Prioritised AR/VR Content Delivery," in *IEEE BMSB 2018 - 13th International Symposium on Broadband Multimedia Systems and Broadcasting 2018*. Valencia, Spain: IEEE Xplore, 2018.

[28] M. Kheirkhah, I. Wakeman, and G. Parisis, "Multipath-TCP in ns-3," in *Cornell University Library*, 2015, pp. 3–4. [Online]. Available: https://bit.ly/2MfiElc