Mobile Networks and Applications (2020) 25:1114–1121
https://doi.org/10.1007/s11036-020-01528-0

Addressing the Bandwidth Demand of Immersive Applications Through NFV in a 5G Network

Armir Bujari[1] · Ombretta Gaggi[1] · Michele Luglio[2] · Claudio E. Palazzi[1] · Giacomo Quadrio[1] · Cesare Roseti[2] · Francesco Zampognaro[2]

Published online: 27 March 2020

Abstract

Immersive technology is expected to become a key player in the next wave of consumer electronics with potential applications spanning a wide variety of sectors. Yet, there are a number of technological hurdles which need to be addressed. Without speed and computation capability, interactivity cannot be ensured. More than often the constrained nature of the mobile device demands is complemented by cloud ones through the use of cloud offloading techniques. With cloud offloading, battery consumption and processing performance problems are translated into sensor data selection and network latency problems. In this context, we focus our study on some functional building blocks that can be used in conjunction and deployed in a 5G virtualized architecture to alleviate bandwidth and latency requirements of immersive application scenarios. To this end, we have created an experimental scenario exploiting real data to test our combined solution based on the field of view cut and optimized transport protocols, discussing the tradeoffs that emerge.

Keywords Immersive technology · Transport layer · 5G · NFV

1 Introduction

The fifth generation (5G) mobile networks are rapidly becoming a reality and they will enable a lot of use-cases to be available in a mobile scenario [1, 2]. Among them, immersive technology such as, for instance, Augmented Reality (AR) and Virtual Reality (VR) represent some of the most interesting and challenging use-cases, embodying stringent latency and bandwidth requirements, outlining the need for the so-called optimized network slices aimed at supporting them [3–5].

The proliferation of accessories used to enjoy immersive realms and the spread of software libraries used to build customized user experiences [6] are evidence of the general interest in this sector. Indeed, different areas such as entertainment, education, industry or medicine have already seen the first applications. One of the most famous examples, in particular for AR applications, is represented by the game Pokémon Go, whose success demonstrated the technical and commercial potentiality of this technology [7]. Also, it gave birth to several copycats, involving even big companies of the game industry such as Microsoft with its Minecraft Earth.

The same domain also hosts dedicated VR devices whereby users, thanks to headsets, can experience immersive gameplay, e.g., [8–10]. Clearly, education is another field that can benefit from immersive contents by allowing users to explore remote areas of the world, discover cultural heritage, and attend lectures in where they explore the human body or practice with virtual surgery [11–13]. In the industry domain, we can have remote diagnostics, repair and design [14]. Applications are basically countless.

Despite a lot of emerging scenarios of employment, we are only at the sunrise of this new technology era because of the numerous technological issues that have still to be addressed. We are far behind the point where users may interact with dynamic scenes in total mobility. The algorithms that have to implement such type of scenarios must be developed by taking into account a lot of elements. They have to provide on-demand localization, graphics

This work is partially funded by ESA through the VIBeS project (Artes AT program), ESA contract number 4000122991/18/UK/ND, and by the Department of Mathematics of the University of Padua through the BIRD191227 project. Responsibility for the presented content resides with the authors.

✉ Claudio E. Palazzi
cpalazzi@math.unipd.it

Extended author information available on the last page of the article.

rendering, user interaction support, and device-to-cloud communications while remaining lightweight and guarantee high performances as well as low energy consumption [15, 16].

A fully immersive virtual environment can even exacerbate these issues and, as an example, let us consider the streaming of 360-degree videos. To comply with stringent Quality of Experience (QoE) requirements of human perception is not a simple task: in [17] the authors consider a human fovea of 200 dots, hence 200 pixel per degree; with this value, a user field of view (FoV) of 150x120 degrees, a future compression factor of 600 and a full motion video at 60 fps could require a 5.2 Gbps of network throughput. These scenarios will be hard to support for the 5G in a static scenario and it will be even harder in case of wireless artifacts involved.

In order to reach the required efficiency in network resource usage, a viable solution could be represented by application-driven-networking. This emerging networking paradigm contemplates the interconnection and chaining of networking functions that together can constitute a so-called service function chain, embodying several optimization's tailored to the dynamics of the scenario at hand [18, 19]. These series of optimizations can contribute to alleviating network resource demand in scenarios that involve media that are particularly hungry of bandwidth without hindering latency requirements:

- *FoV-Cut* function: it performs a cut of the 360-degree scene on the basis of the user(s) FoV. The network function receives the user FoV as an input to compute the resulting scene that will encompass it, not airing the full video but only parts of it.
- *FoV-Prediction* function: it predicts a users' FoV on the basis of historical inputs; it then uses the outcome to pre-fetch those pieces of the scene that are not yet aired. The network function can be employed when the network path is idle or when it is not fully utilized, i.e., the user is relatively static.
- *Optimized Transport*: in order to support timely heterogeneous data flows, lightweight protocols for fast delivery of data (e.g., UDP) and reliable delivery protocols (e.g., legacy or a TCP variant) could be necessary. State-of-the-art proposals such as VoAP and TCP-Wave could be suitable candidates achieving the targeted objective [20–22].

The aforementioned optimizations can be provisioned as virtualized network functions. They can be orchestrated and managed by an architectural framework that has end-to-end visibility on the network resources. The Network Function Virtualization (NFV) can come into help: it is a 5G ingredient designed as a logically centralized brain, capable of unified resource management [23]. In this article, we discuss two functional blocks, i.e., (*i*) a $FoV-Cut$ function and (*ii*) an optimized transport layer function that are used jointly in different scenarios of deployment, besides discussing the various tradeoffs that emerge. We analyze a scenario in where a virtual augmented environment is provided to users, that are subject to the same dynamics and with the possibility to customize several features of the world in where they are immersed in and/or to interact with it.

The rest of this paper is organized as follows. In Section 2 we provide the background of some 5G technologies enablers. Section 3 offers details of the dataset under scrutiny while Section 4 discusses a FoV delimiting function. In Section 5 we analyze the requirements of an optimized transport layer and possible solutions to guarantee co-existence among heterogeneous flows. Section 6 presents the experimental scenario and tradeoffs that emerge. Finally, Section 7 concludes the article.

2 Background

Differently from the previous generations of mobile networks, the 5G architecture is a service-based architecture (SBA) in which the system functional blocks are implemented as Network Functions (NFs) that expose a set of standardized interfaces to other NFs authorized to interact [24]. NFs are stored in a NF repository in order to discover the services offered by other NFs. This NF based architectural approach supports the concepts of modularity, reusability and self-containment fostered by current network softwarization approaches in order to exploit the advantages of state-of-the-art virtualization technologies.

From a high-level point of view, a 5G network is thus a set of logical network functions, deployed on top of a set of end-to-end physical resources, that need to be dynamically instantiated, interconnected, orchestrated, scaled, migrated and terminated. The main enabling technologies in this vision are:

1. Network Function Virtualization (NFV), i.e., the set of technologies meant to replace physical implementation of network devices and other network functional blocks (routers, firewall, ACL, etc.) with (virtualized) software implementation running on legacy multipurpose hardware [23];
2. Software Defined Networking (SDN), i.e., the set of forwarding abstraction, node architectures, and configuration protocols to dynamically re-program high speed forwarding devices through external software controllers [25].

In particular, NFV (Fig. 1) is a reference network architecture used to virtualize entire classes of network

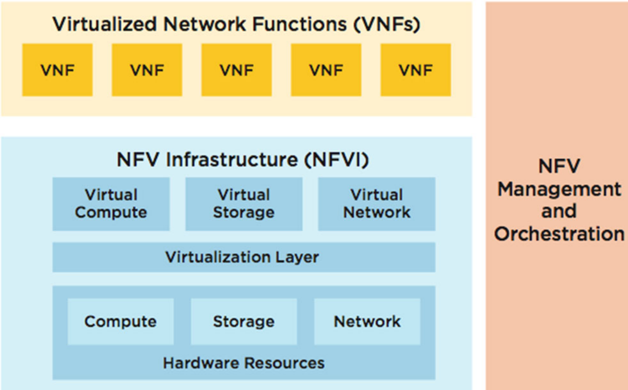


Fig. 1 High level NFV reference architecture

node functions into building blocks that may connect, or chain together, to create specialized communication services. These resources are deployed as Virtualized Network Functions (VNFs) on top of Network Function Virtualization Infrastructure (NFVI) whose lifecycle is managed by a logically centralized brain called the Management and Orchestration (MANO) plane. In this context, several integration patterns have been proposed, integrating SDN technology into the NFV reference architecture, providing the flexibility to enable VNFs across multiple domains and technologies in order to create a service-specific network, and this is referred to as Network Slicing [26].

While current deployments of 5G are not there yet, the design and provisioning of specialized services e.g., one or more slices of immersive environments, through the aid of a standardized NFV/SDN architecture and reference points has the positive impact of being 5G-ready.

In the following, we delve discuss the network functions under scrutiny, whose objective is that of alleviating bandwidth and latency requirements of immersive applications.

3 The dataset

Wen-Chih Lo et al. in [27] report a dataset containing user-centric metadata captured while subjects were watching 360-degree scenes through an Oculus Rift. In specifics, the dataset contains data on 50 people watching ten 360-degree videos. All the videos were taken from YouTube, and they have been cut and modified to have:

- one minute fixed duration;
- a resolution of 3840x1920;
- a frame rate of 30fps.

Hence, totalling 1800 frames per video. The data includes, but is not limited to, the user FoV captured on each video frame. Therefore, for each user, there is a log containing the FoV coordinates for every frame, represented with 3 fields: (i) yaw [-180,180] representing the up and down direction, (ii) pitch [-90,90] representing the left and right direction and, (iii) roll [-180,180] representing the rotation. We also have a summary of the area the person is watching, represented using tiles (later on).

Before delving into further details, it is noteworthy to point out that Youtube employs the equirectangular projection technique to map a 360-degree video in a rectangular-shaped frame. Referring to Fig. 2, the 2D rectangular is divided into three parts comprised as follows (i) a quarter of the rectangular is the half back semi-sphere, (ii) two quarters is the front semi-sphere, and (iii) the last comprising another quarter belonging to the half back semi-sphere. Along the equator of the sphere and plane the projection does not make distortions on the image, but as we approach the poles the distortion becomes greater. A horizontal line near the poles in the rectangular is a semi-circular line in the semi-sphere.

4 Field of view delimiting function

The objective is to asses different scene cut approaches in terms of the amount of saved bandwidth while considering the aggregate FoV of the users. In the envisioned scenario multiple users experience the same virtual experience scenario, i.e., are subject to the same scene dynamics. In the following we briefly discuss the computational methods evidencing their goodput.

In the first approach, referred to as the tile method, the scene is divided into portions of NxM size where both M and N are quantities denoting the number of pixels. This quantity represents a portion of a frame and in the original dataset, a tile has a size of 192x192 pixels for a total of 200 tiles per frame. The tiles are numbered incrementally starting from the upper-left to lower-right. Moreover, for each person there is a list in the dataset of which tiles per frame have been viewed by the user during video playback.

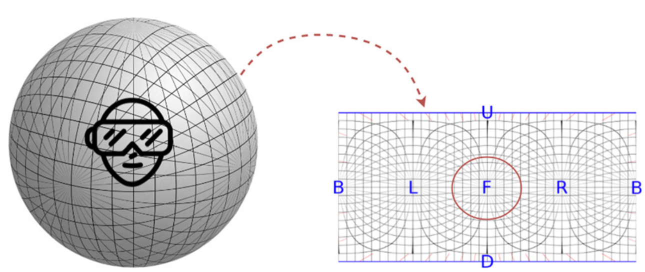


Fig. 2 Equirectangular projection. On the left is shown is the spherical representation of the scene while on the right the equirectangular projection. The letters in the right figure denote the Back, Left, Front, Up, Down and Right direction, respectively

The tile method simply counts for each frame all the tiles that all users have viewed, removing any duplicates. From this basis, it is possible to plot a graph like in Fig. 3, representing the amount of pixels viewed by the users altogether. With these data, we can even draw considerations on the behaviour of specific users while viewing a particular video scene.

This computational method is based on the assumption that the scene could be segmented into chunks of an *a-priori* known size which are then glued together and aired. Moreover, a tile of size 192x192 pixels might represent an over estimation of whats an individual user actually views. An alternate method, performing the cut dynamically based on the users' FoV barycenter, and discussed in a prior work [28], yields better results.

5 An optimized transport layer

Transport layer optimization is one of the main challenges when trying to satisfy immersive application requirements [29]: i) support of different classes of traffic, ii) fairness, iii) low latency, and, iv) flexibility of use in distributed or hybrid architectures. Interestingly, recent transport layer research focuses its attention on application requirements. This is the case of those protocols that concentrate on the session layer, aiming to improve end-to-end application performance through a better and more efficient scheduling of segments (e.g., SPDY/HTTP2.0, QUIC, MP-QUIC, TCP WAVE, MP-TCP) [30, 31]. Such solutions may be integrated as part of the Transport Layer Optimization as a VNF (TLO-VNF) to improve transport efficiency over specific network segments where the management of data flows can be beneficial to increase the overall scenario experience (Table 1).

As an add-on enhancement at the transport layer, we explore the adoption of novel TCP variants, namely TCP Wave and BBR protocol, designed with the aim of optimizing performance over links spanning on a large set of bandwidth and delay values, while having as a target objective the limitation of the end-to-end delay close to its minimum value [31–33]. These characteristics are expected to positively affect performance in scenarios where mobility and dynamic link configurations are required, as well as in those scenarios where different classes of traffic are supposed to co-exist.

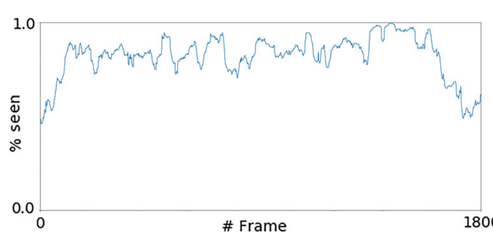


Fig. 3 Example reporting the percentage of pixels seen for each frame in the Roller Coaster scenario

6 A case study

For a clearer understanding, we discuss a practical example of a realtime application modelling, identifying the elements of the transmission chain. In the targeted scenario, discussed in the introduction, different data flows can be identified, among which: (i) multimedia data, e.g., an immersive video flow from a server to the users, (ii) sensing data, e.g., the users' FoV, data from the user(s) to the server, and (iii) unicast data from the user(s) to a server transporting personalized content. In particular, the video flow is bandwidth hungry and has to fulfill stringent latency requirements adhering to QoE specifications [34].

The chain includes a set of operations which take a given amount of time. The first case, herein referred to as regular, refers to a standard immersive video streaming service, where the source video might be pre-encoded. The operations involved, in sequence, are as follows:

$$\begin{aligned} regular\_chain &= VE && : \text{video encoding} \\ &+ TX && : \text{send frame to RTP} \\ &+ net && : \text{network} \\ &+ RX && : \text{frame reception} \\ &+ DJ && : \text{de-jitter buffer} \\ &+ VD && : \text{video decoding} \end{aligned} \tag{1}$$

The VE time can vary depending on whether the video is already recorded and encoded in the proper streaming format in the user device or into a local streaming server. Nonetheless, even in this case a packetization time shall be considered. Otherwise, if the video is generated in realtime (i.e., augmented reality applications, or streaming of a live content), such time must be considered. We can refer to the delay budget as the sum of the latency components introduced by each of the above mentioned elements.

The quantity $TX + net + RX$ refers to the delay contribution of the network transport element. In case of non realtime applications, such time is not a problem. Instead, this time is critical for realtime immersive applications and also in the proposed approach where the video shall be encoded and delivered to the users according to their feedback. Therefore, in introducing the optimized approach, we have to include the time required to collect the feedback from the users, i.e., FoV used to identify the relevant portion

Table 1 Methods comparison. More details on the online computational method can be found in [28]

Videos	Tiles (% seen)	Ad-Hoc Square (% seen)	Ad-Hoc Oculus Rift (% seen)
Mega Coaster	0.7854	0.7990	0.6275
Roller Coaster	0.82132	0.8415	0.6609
Driving with	0.8960	0.8617	0.6768
Shark Shipwreck	0.9670	0.9656	0.7584
Perils Panel	0.8441	0.8152	0.6403
Kangaroo Island	0.8645	0.8359	0.6565
SFR Sport	0.7698	0.7993	0.6278
Hog Rider	0.7607	0.7870	0.6181
Pac-Man	0.6812	0.7466	0.5864
Chariot Race	0.8111	0.8263	0.6490

of the video to send, through a dedicated video encoding function which shall work in realtime.

In this second case, the video encoding time is always present, because the transmitted content reflects the users' feedback. As well, the network component shall be carefully considered, to allow for a proper exploitation of the service. Nonetheless, we can consider the feedback component negligible in the delay budget: in fact, the head movement speed is limited (i.e., the user cannot turn front to back in 0-time), so a given amount of frames are processed before a significant change to the field of view. In other terms, the encoding process can be performed independently from the users' feedback. Therefore, we can assume that the delay budget is the same as in the regular case also for the proposed application, but a VE time must be considered. Finally, the overall time to complete the transmission of a frame, considering the requirements of live applications, shall be less than a reference value, representing the upper bound for the delay budget. In regular application scenarios, such time can be assumed as in [29] to be 75 ms.

6.1 The scene

As an optimization, alleviating network bandwidth demand, the *FoV-CUT* network function could be employed, performing a cut of the scene, airing only parts of it falling into the aggregate users FoVs. The targeted virtualized network function has to rely on its computation on the FoV coordinates, yaw and pitch, which are easily projected into a 2D plane, using the equirectangular projection. Assuming a user FoV of 100x100 degrees, to compute the bandwidth expenditure, we check whether a tile falls into a user's FoV. If this is the case, we consider the tile as part of the scene, otherwise it is omitted from it and not accounted for in the bandwidth requirement.

For the following analysis and without loss of generality, we chose a particular scene pertaining to a roller coaster scenario where the amount of bandwidth saved amounts to 22%.

6.2 Experimental settings

We decided to implement a part of the system into a real network, to evaluate the effective timing for realtime video distribution. We developed an RTP-like application using UDP traffic, which is suitable for multicast transmissions, guaranteeing the lowest latency with respect to TCP-based video protocols [20]. The developed application takes into account actual frame sizes evaluated theoretically from the dataset. A given group of packets is generated for every video frame each 33 ms, representing a framerate of 30 fps. Only if all such packets are received, the frame is sent to the de-jitter buffer, otherwise we have a frame loss.

All the tests are preformed in a controlled environment; both source and receiver clocks are synchronized using the NTP protocol, with an accuracy granularity of less than 1 ms. Therefore, the time difference between the packet timestamp and the reception time, denoting the overall network delivery time can be measured.

We assume two possible configurations of the service: (i) a streaming server is located in the private cloud, core network of the network operator, assuming a network delay of 10 ms and (ii) a local network where the Mobile Edge Computing (MEC) video source functionality resides, directly delivering the video to the end users, with a network delay of 2 ms. The available bitrate is 35 Mbit/s.

In the evaluations, we did not consider the video encoding time, but rather just send bulk data. Therefore, from the delay budget and according to the results obtained, this value can be identified as a specification to develop such functions in a future real scenario.

In this context we run a reference test with no background traffic, for both configurations. This allows us to identify the nominal transmission values for the proposed application. Then, in addition, we run other tests were a TCP connection is activated sharing the bottleneck link of the system. The TCP-based flows represent user personalized content, augmenting the experience, which ranges from an embedded video stream(s) and/or objects used to interact with the environment.

6.3 Results

In all the performed tests we measured the following quantities: (i) jitter, (ii) min and max values for inter-arrival of packets, (iii) network delivery time and (iv) frame losses. We report that in all the considered scenarios no frame loss was experienced. Concerning the jitter measure, it is inferred by RTP protocol to provide a short-term indication to the sender of network congestion.

In Fig. 4 we report the resulting jitter values. In the case of an unloaded network, that is only UDP video traffic is present in the bottleneck, or when using BBR, we obtain a jitter value of about 1 ms. In the other cases, TCP Wave and CUBIC, this value is higher and oscillates periodically. In the case of Wave the jitter goes from 1 ms to 5 ms, while for Cubic we have larger periods where the value is high, reaching peaks of 12 ms in the private cloud configuration, otherwise at lower values in the MEC one. Therefore, especially in the case of TCP concurrent traffic, additional analysis is required to identify the size of a possible de-jitter buffer.

In fact, when receiving a real time video in the presence of jitter, a de-jitter buffer is required, which ultimately affects the overall delivery time. The buffer allows to smooth out the received frames and gives tolerance to possible delay variations in reception. The size of the de-jitter buffer is a trade-off between making it small (giving the advantage of introducing less delay, but at a risk of discarding those packets not arriving before the buffer becomes empty) or making it large (introducing a larger delay, but reducing frame losses). Since the jitter function used in practice is an exponential weighted average with a reduction factor of 16, its absolute value might not be enough to adequately tailor the de-jitter buffer [35]. In fact, we may still experience some frames reaching the receiver too late with a single-frame de-jitter buffer.

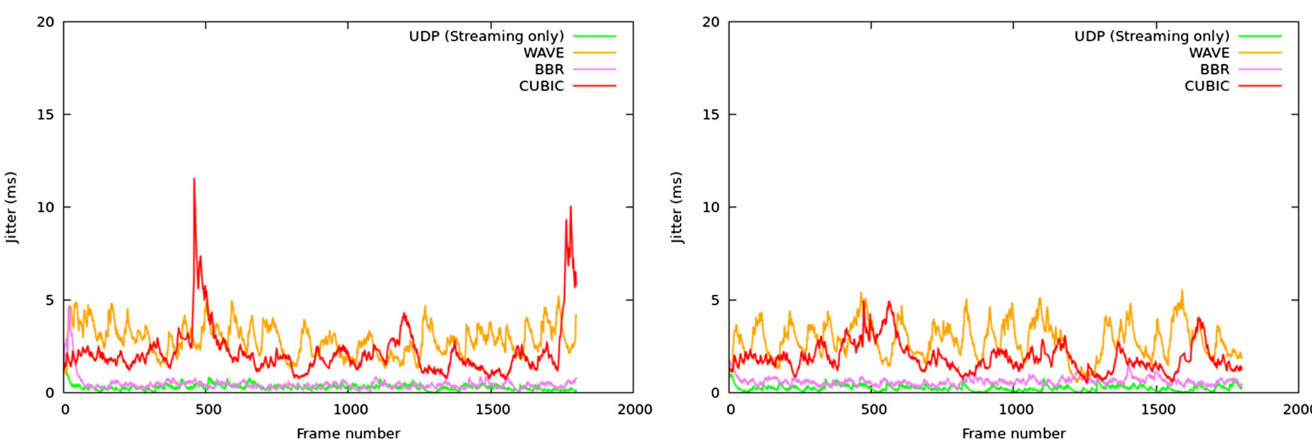
As stated in [35], a reasonable size for the de-jitter buffer shall take into account the minimum and maximum delay in receiving the packet with respect to the timing reference of 33 ms. Note that complex dynamic approaches for the configuration of the de-jitter buffer exist, but due to the nature of the service they are not applicable, since they may introduce variations in user perception of the scenario. In Table 2, we report the measures obtained for the inter-arrival timing.

Table 2 Inter-arrival timing for the unloaded network scenario

Private cloud	MEC
RTP: min 28 ms, max 38 ms	RTP: min 28 ms, max 38 ms
Wave: min 20 ms, max 56 ms	Wave: min 18 ms, max 58 ms
BBR: min 12 ms, max 53 ms	BBR: min 25 ms, max 45 ms
CUBIC: min 14 ms, max 332 ms	CUBIC: min 20 ms, max 467 ms

While on the unloaded network scenario we can consider a relatively small de-jitter buffer of 10 ms, such value increases in the presence of concurrent TCP traffic (Fig. 4). In this context, both Wave and BBR can benefit from a buffer in the range of 1 frame (33 ms) while Cubic requires a much larger buffer of about 10 frames. We can observe that such times, as expected, are almost independent from the network latency.

Next, we consider the overall network delivery time, intended as the time since the frame is available at the source (output of the encoder) to be sent as UDP packets until it is fully received. This value represents the $TX + net + RX$ component, and must be as low as possible, to make this



(a) Private cloud (b) MEC

Fig. 4 Jitter measured in the different configuration scenarios

Table 3 Measured delivery time

Private cloud	MEC
RTP: 25 ms	RTP: 19.8 ms
Wave: 32 ms	Wave: 29.4 ms
BBR: 26 ms	BBR: 22 ms
Cubic: 569 ms	Cubic: 608 ms

application feasible. From Table 3, we can see that both BBR and Wave guarantee lower delivery times in both the private cloud and MEC scenario. We also note that for the MEC scenario, we also have some significant gains (in the range of 3-5 ms). Instead, with Cubic, the delivery time is significantly higher in both the MEC and Private Cloud setup, due to the protocols' attitude in filling the buffer to exploit the optimal transmission point.

Starting from the delivery time values measured on our testbed and the previous discussion concerning buffering, we can compare all such contributions with regard to an hypothetical delay budget limit of 75 ms. It is noteworthy to point out that a possible co-existence with Cubic is not possible, due to its delivery profile, so it will not be considered. The optimal case is BBR, where $TX + net + RX = 22ms$ and DJ in the presence of TCP Wave or BBR $= 33ms$. This gives a $VE + VD$ time of $(75ms - 55ms) = 20ms$. Therefore, if the encoding ad playout times are compatible with this value, the application is practically feasible into a real network.

7 Conclusion

It is difficult to delimit the application potential of immersive technologies, as their possible uses are virtually uncountable. In fact, basically all major IT players are part of the long list of companies that are researching and interested in this area. Nevertheless, the so-called fourth wave of consumer electronics has not arrived yet and specific business models are needed [36]. This is even worse when considering innovative and highly demanding (in terms of performance and required resources) applications such as the immersive ones, as the fundamental technological problems that have been discussed in this work are still far from being solved. However, as the first step in this direction, we have here proposed an architectural and transport layer combined solution that may successfully lead the way to answer the latency and (distributed) coherence challenges posed by immersive applications. Such an approach may serve as a reference for the developer communities of immersive platforms, namely next-gen mobile networks as well as hardware designers and developers.

References

1. Sexton C, Kaminski NJ, Marquez-Barja JM, Marchetti N, DaSilvah LA (2017) 5G: Adaptable networks enabled by versatile radio access technologies. IEEE Communications Surveys & Tutorials 19(2):688–720
2. Anbalagan S, Kumar D, Ghosal D, Raja G, Muthuvalliammai V (2017) SDN-assisted learning approach for data offloading in 5G HetNets. Mobile Netw Appl 22(4):771–782
3. Bastug E, Bennis M, Medard M, Debbah M (2017) Toward interconnected virtual reality: opportunities, challenges, and enablers. IEEE Commun Mag 55(6):110–117
4. Melis A, Mirri S, Prandi C, Prandini M, Salomoni P, Callegati F (2018) Integrating personalized and accessible itineraries in MaaS ecosystems through microservices. Mobile Netw Appl 23(1):167–176
5. Bujari A, De Giovanni L, Palazzi CE (2018) Optimal configuration of active and backup servers for augmented reality cooperative games. Concurrency and Computation: Practice and Experience 30(20):e4454
6. Digicapital For AR/VR 2.0 to live, AR/VR 1.0 must die. Available: https://www.digi-capital.com/news/2019/01/for-ar-vr-2-0-to-live-ar-vr-1-0-must-die. Last accessed: April 10 2019
7. Pokémon Go Available from: https://www.pokemongo.com. Last accessed: April 10 2019
8. Maggiorini D, Quadri C, Ripamonti LA (2012) On the feasibility of opportunistic collaborative mixed reality games in a real urban scenario. In: Proceedings of the International Conference on Computer Communications and Networks (ICCCN), Munich, Germany
9. ThinkMobiles Virtual Reality in Gaming Report. Available: https://thinkmobiles.com/blog/virtual-reality-gaming. Last accessed: April 10 2019
10. Roccetti M, Marfia G, Amoroso A, Caraceni S, Varni A (2012) Augmenting augmented reality with pairwise interactions: the case of Count Luigi Ferdinando Marsili Shooting Game. In: Prof. of IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA
11. Prat P, Ives M, Lawton G, Simmons J, Radev N, Spyropoulou L, Amiras D (2018) Through the HoloLens™ looking glass: augmented reality for extremity reconstruction surgery using 3D vascular models with perforating vessels. European Radiology Experimental 2(1)
12. Polyakova IR, Maglieri G, Mirri S, Salomoni P, Mazzeo R (2019) Art scene investigation: discovering and supporting cultural heritage conservation through mobile AR. In: Proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France
13. Donatiello L, Morotti E, Marfia G, Di Vaio S (2018) Exploiting immersive virtual reality for fashion gamification. In: Proc. of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy
14. Palmarini R, Erkoyuncu JA, Roy R, Torabmostaedi H (2018) A systematic review of augmented reality applications in maintenance. Robot Comput Integr Manuf 49:215–228
15. Ciman M, Gaggi O (2014) Evaluating impact of cross-platform frameworks in energy consumption of mobile applications. In: Proc. of the International Conference on Web Information Systems and Technologies (WEBIST), Barcelona, Spain, pp 423–431
16. Ciman M, Gaggi O (2017) An empirical analysis of energy consumption of cross-platform frameworks for mobile development. Pervasive and Mobile Computing 39:214–230

17. Begole B (2017) Responsive media: Media experiences in the age of thinking machines. APSIPA Transactions on Signal and Information Processing 6:E4
18. Medhat AM, Taleb T, Elmangoush A, Carella GA, Covaci S, Magedanz T (2017) Service function chaining in next generation networks: state of the art and research challenges. IEEE Commun Mag 55(2):216–223
19. De Giovanni L, Della Croce F, Tadei R (2004) On the impact of the solution representation for the internet protocol network design problem with max-hop constraints. Networks 44(2):73–83
20. Bujari A, Massaro M, Palazzi CE (2015) Vegas over access point: making room for thin client game systems in a wireless home. IEEE Trans Circuits Sys Vid Technol 25(12):2002–2012
21. Abdelsalam A, Luglio M, Roseti C, Zampognaro F (2016) TCP wave resilience to link changes: a new transport layer approach towards dynamic communication environments. In: Proc. of the International Conference on Data Communication Networking (DCNET 2016), Lisbon, Portugal
22. Abdelsalam A, Luglio M, Roseti C, Zampognaro F (2017) TCP wave: a new reliable transport approach for future internet. Comput Netw 112:122–143
23. Mijumbi R, Serrat J, Gorricho JL, Bouten N, De Turck F, Boutaba R (2016) Network function virtualization: state-of-the-art and research challenges. IEEE Communications Surveys & Tutorials 18(1):236–262
24. Carella GA, Pauls M, Magedanz T, Cilloni M, Bellavista P, Foschini L (2017) Prototyping NFV-based multi-access edge computing in 5G ready networks with Open Baton. In: Proc. of the IEEE conference on network softwarization (NetSoft), Bologna, Italy, pp 1–4
25. RFC 7426 Software-Defined Networking (SDN): Layers and architecture terminology. Available from: https://datatracker.ietf.org/doc/rfc7426. Last accessed April 10, 2019
26. Bujari A, Palazzi CE, Polonio D, Zanella M (2019) Service function chaining: a lightweight container-based management and orchestration plane. In: Proc. IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, pp 1–4
27. Lo W, Fan C, Lee J, Huang C, Chen K, Hsu C (2017) 360° video viewing dataset in head-mounted virtual reality. In: Proc. of the ACM Multimedia Systems Conference (MMSys), Taipei, Taiwan
28. Bujari A, Palazzi CE, Perale D (2019) Addressing bandwidth demand in full-immersive virtual reality. In: Proc. of EAI international conference on smart objects and technologies for social good, ACM, Valencia, Spain
29. Braud T, Bijarbooneh FH, Chatzopoulos D, Hui P (2017) Future networking challenges: the case of mobile augmented reality
30. De Coninck Q, Bonaventure O (2017) Multipath QUIC: design and evaluation. In: Proc. of the ACM international conference on emerging networking EXperiments and Technologies (CoNEXT 2017), pp 160–166
31. Abdelsalam A, Luglio M, Roseti C, Zampognaro F (2017) TCP wave: a new reliable transport approach for future internet. Computer Networks, 112
32. Cardwell N, Cheng Y, Gunn CS, Yeganeh SH, Jacobson V (2016) BBR: congestion-based congestion control. ACM Queue 14:20–53
33. De Giovanni L, Massi G, Pezzella F, Pfetsch ME, Rinaldi G, Ventura P (2013) A heuristic and an exact method for the gate matrix connection cost minimization problem. Int Trans Oper Res 20(5):627–643
34. Begole B (2017) Responsive media: media experiences in the age of thinking machines. APSIPA Transactions on Signal and Information Processing, 6
35. Morton A, Claise B (2009) Packet delay variation applicability statement. RFC 5481
36. Bujari A, Furini M, Mandreoli F, Martoglia R, Montangero M, Ronzani D (2018) Standards, security and business models: key challenges for the IoT scenario. Mobile Netw Appl 23(1):147–154

Affiliations

Armir Bujari[1] · Ombretta Gaggi[1] · Michele Luglio[2] · Claudio E. Palazzi[1] · Giacomo Quadrio[1] · Cesare Roseti[2] · Francesco Zampognaro[2]

Armir Bujari
abujari@math.unipd.it

Ombretta Gaggi
gaggi@math.unipd.it

Michele Luglio
luglio@ing.uniroma2.it

Giacomo Quadrio
gquadrio@math.unipd.it

Cesare Roseti
roseti@ing.uniroma2.it

Francesco Zampognaro
zampognaro@ing.uniroma2.it

[1] Department of Mathematics, University of Padua, Via Trieste 63, Padua, Italy

[2] Department of Electronics Engineering, University of Roma Tor Vergata, Rome, Italy