González Morín, Diego; García Armada, Ana; Pérez, Pablo. Cutting the cord: key performance indicators for the future of wireless virtual reality applications. In: *12th IEEE/IET International Symposium on Communication Systems, Networks and Digital Signal Processing, (CSNDSP), 20-22, July 2020 (Online). IEEE, Nov 2020, 6 pp.*

DOI: https://doi.org/10.1109/CSNDSP49049.2020.9249445

Cutting the Cord: Key Performance Indicators for the Future of Wireless Virtual Reality Applications

Diego González Morín
Nokia Bell Labs Spain
Madrid, Spain
diego.gonzalez_morin@nokia-bell-labs.com

Ana García Armada
Universidad Carlos 3 de Madrid
Madrid, Spain
anagar@uc3m.es

Pablo Pérez
Nokia Bell Labs Spain
Madrid, Spain
pablo.perez@nokia-bell-labs.com

***Abstract*—The new generation of telecommunication networks (5G) are considered a key technology enabler for the future of Virtual Reality (VR) technologies. 5G improved throughput and low latency capabilities open the door for novel distributed VR architectures that will accelerate the development of wireless immersive VR. While the downlink requirements for wireless VR have been previously studied, the uplink transmissions have been completely neglegted. However, for the succesful consecution of a fully immersive wireless VR, the uplink stream becomes as important as the downlink. In this paper, we thoroughly analyze VR thecnologies state of art to extract an initial set of data flow and processing times requirements for a subgroup of identified key algorithms. Besides, we propose the peak throughput and latency requirements Key Performance Indicators (KPIs) estimation as a simple optimization problem, given the data gathered in the state of the art analysis. Finally, we propose a set of valid ranges of peak throughput and latency requirements for a selected group of three different offloading scenarios.**

***Index Terms*—5G, Virtual Reality, Key Performance Indicators, Multi-Access Edge Computing**

I. Introduction

Virtual Reality (VR) is considered one of the most important media technology disruptions of the last decade: VR´s capability of fully immerse the user into a virtual scenario has revolutionized a wide range of fields such as industry, entertainment or education. Furthermore, VR high-end devices have reached unprecedented levels of visual resolution: latest premium devices, such as Varjo XR-1 [1], achieve a visual resolution of 60 pixels per degree or a 20/20 vision, which is comparable to the human eye's resolution. This level of resolution allows the developers to design extremely visually-realistic scenarios which widens the set of VR target applications.

Even though the improvements on VR visual resolution are remarkable the overall user experience is still some steps behind. The sense of embodiment or self awareness is a fundamental factor which drastically affects the user experience on VR applications [2]. However, effective embodiment and immersiveness require computationally demanding state of the art algorithms such as hand tracking, accurate pose estimation or real-time egocentric segmentation which require a continuous feed of sensor data, such as high definition RGB and depth camera frames. Furthermore, the motion to photon latency should remain below 15 ms to avoid motion sickness or discomfort during the VR experience [3]. This strict requirement along with the aforementioned level of resolution demands the usage of high-end GPUs for real-time rendering which are too bulky to fit inside the goggles. As a consequence, state of the art VR devices are still tethered, considerably constraining the user mobility and comfort and reducing the range of possible use cases.

Over the last years, the interest on VR technologies has quickly grown with the introduction of the new generation of telecommunication networks (5G). 5G includes a set of 3 main capabilities, enhanced mobile broadband (eMBB), ultra reliable low latency (URLLC) and massive machine-type communication (mMTC) [4] which have widen the limits of telecommunication technology. Consequently, 5G networks have been considered a key technology enabler for VR technologies: high data rates along with ultra low latency capabilities open the door to process offloading for VR technologies, using 5G multi-access edge computing (MEC). VR offloading might not only enable lighter untethered VR devices, which can drastically improve the user mobility and overall immersiveness, but also increase the computational resources available for the VR application.

VR offloading is considered an important technology disruption as it will enable a fully immersive wireless VR. There are several research proposals of task offloading and resource allocation algorithms for 5G MECs. In [5], [6] and [7] different optimization problem-based task scheduling methods are proposed. In [7] the optimization problem is not only constrained by the currently available resources but also by the measured latency and energy consumption. Similarly, in [8], the authors propose a reinforcement learning-based method in which the goal is to optimize the effective latency and energy consumption. Other research examples are, on the other hand, more focused on the specific application of VR which aim to study the necessary network requirements and architectures for a successful VR offloading. In [9], the authors propose two different network slicing methods to enhance the URLLC and eMBB services and fulfill the intensive requirements of wireless VR. In [3] the authors propose theoretical architectures based on millimeter wave (mmWave) 60 GHz networks

This work has received funding from the European Union (EU) Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie ETN TeamUp5G, grant agreement No. 813391.

operating in an indoor environment. They used statistical models of the network to estimate its performance on different setups, from which they concluded that end to end latencies could be kept below 8 ms for up to 16 users with the aid of 4 mmWave band access points (mmAP) and 8 MECs. In [10] it is studied how different offloading computational resources configurations affect the final immersiveness of the experience. More specifically, they statistically study how the type of external computing resource, edge, fog or cloud servers, and their available resources affect the perceived experience.

The VR-specific research examples are only considering the downlink data rate and latency requirements for their analysis. However, VR immersiveness is not just defined by the visual resolution: the user needs not only to realistically perceive the virtual scenario, but seamlessly interact and move within it. Furthermore, in [11] the authors highlight the importance of the sense of embodiment and sense of presence in VR applications to allow the user to really feel part of the experience rather than just being and observer. State of the art VR research aims to produce novel methods and algorithms to improve the user experience, interaction and self awareness. Most of these methods involve high uplink data rates to transmit the sensor data captured in each frame while the end to end latencies should lay below the sampling period of the device, which is smaller than 16 ms in current VR devices. These tight constraints along with the intensive computational resources favor the idea of VR offloading relying on 5G eMBB and URLLC capabilities. The tight relation between latency and quality of experience in VR enhances the importance of MEC-based solutions to reduce the extra latency added by the backhaul link or other propagation delays.

The goal of this paper is to study the effective latency and throughput requirements for a successful VR offloading, as a function of the different computation times estimated for a set of different VR algorithms or processes. A state of the art based analysis of the downlink and uplink data rates, and time requirements is described for such processes. Later, we propose a simple optimization problem to estimate the less restrictive sets of effective required donwlink and uplink peak throughput and end to end latency values which ensure per frame processing times below the device´s sampling rate. This estimation is performed for a set of different proposed VR offloading scenarios.

II. A Breakdown of Immersive Virtual Reality Applications

As previously introduced, the immersiveness and sense of embodiment requires not only state of the art VR resolution, but also a set of algorithms which aim to enhance the user´s self awareness and interaction with the virtual scene. We are focusing our study on 4 main groups of algorithms which will we briefly dissected to extract their data flow and timing requirements, which will be the baseline for our proposed approach to estimate an initial set of peak throughput and latency requirements.

For our analysis, we use the Varjo XR-1 [1] as the reference device to estimate the different sensors' resolution. We chose this device as it combines a 20/20 vision resolution with cutting-edge sensorization. The device includes a set of high-resolution stereo cameras for video pass-through applications and a active IR-based depth sensor. These stereo cameras can be used to apply some real-time egocentric segmentation techniques to allow the user to see himself/herself, which can considerably increase the user's self awareness and sense of embodiment [12]. In the proposed algorithms, the frames are analyzed individually. Consequently, we considered each frame to be sent individually rather than as a video feed. We assume a compression ratio of 20, estimating a pixel weight of 0.9 to 1.3 bits. Besides, according to the reference device, the capture stereo RGB data has a resolution of 1080p, which is translated to a frame size of 3.32 to 5 Mbits. The rendered frame is composed of two 4K images, one for each eye, adding up to a total frame size of 14.92 to 19.9 Mbit. Finally, the estimated depth frame size according to the device specifications can be bounded within 0.5 and 1.66 Mbits. These estimated frame sizes are used along the numerical analysis proposed in this paper.

A. Real-time Pose Estimation

5G offloading of VR applications might lead to a new generation of wireless high-end VR devices. Consequently, the user will have the chance to freely move within the virtual scene rather than just acting as a static observer. In this new scenario it is even more crucial to accurately track the device's pose in real-time as pose inaccuracies can provoke motion sickness. On mobile scenarios, the device translations are not constraint and the tracking errors can critically increase, which might completely ruin the experience. Current devices use simultaneous localization and mapping algorithms (SLAM) for real-time 6DoF device pose estimation using the RGB feed and the embedded inertial measurement units (IMUs) [13]. Besides, the overall robustness and accuracy can be improved by adding depth information from the available depth sensors. The output of SLAM algorithms can be reduced to a set of pose estimations. In general, these algorithms run in real-time at frequencies of 30 to 60Hz. In general, high update rates usually improve the accuracy, while demanding more computational power [14]. See Table I for an itemized summary of the data flow and timing requirements.

B. Hand Tracking

Hand tracking is a key feature which has been introduced in most of the newest devices over the recent years, allowing the user to interact with the virtual content with his/her own hands instead of using a remote control device [15] [16] [17]. However, the majority of these algorithms are not robust enough yet as their performance strongly depends on the surrounding environment and the used training data, while they still demand high computation resources. Incorporating a very robust failure-safe hand tracking system is a key factor to considerably increase the sense of embodiment and user experience in VR. Besides, a robust and precise hand tracking might enable the usage of VR based training for real situations in fields such as medical or military, as it permits a more realistic interaction with the virtual objects. In [15], the authors propose a generative adversarial network (GAN) based solution to solve the hand tracking problem from a single RGB sensor. On the contrary, in [17], the authors just use a single depth sensor as the only input to their proposed method. Finally, in [16], both the RGB and depth feeds are used for real-time hand tracking estimation. We consider this last setup as our reference scenario to numerically analyze hand tracking algorithms as it is the most demanding one. Regarding the update rate, in [18] the authors achieve an update rate of 50Hz on a GPU-less device. We assume update frequency rates of 60 to 90Hz for hand tracking state of the art examples running on high-end hardware, with processing times between 8 and 12 ms (see Table I)

C. Egocentric Segmentation

The way the user perceives his/her own hands can drastically affect the sense of embodiment and the immersiveness [12]. There is a new research path which explores methods to allow the user see himself/herself within the virtual scenario without the use of avatars. The so called egocentric segmentation techniques rely on computer vision algorithms to segment the hands or other human body parts from a real-time egocentric video feed. The segmented feed is included and seamlessly blended with the virtual content. Some research examples of such techniques are explained in [19] [20] [21].The main goal of these approaches is to improve the sense of embodiment, the sense of presence (SoP) and the interaction with the virtual or even real objects as in [20]. In general, these methods rely on dual stereo camera feed for realistic binocular representation of the segmented video feed. Egocentric segmentation should run at the same data rate as the device´s update rate to avoid any flicker or miss alignment which can ruin the experience. Besides, the output is also a stereo feed to be blended with the stereo rendered scene and displayed in each eye. The summary of the data flow and timing requirements for offloaded egocentric segmentation algorithms is shown in Table I.

D. Scene Rendering

Rendering stereo 20/20 vision with a density of 60 pixels per degree per eye demands intense computational resources. The devices which provide this level of resolution require high-end GPUs: the Varjo XR-1 specifications require the use of a NVIDIA GeForce® RTX 2080 Ti GPU. Besides, the data transmission rate between the device and the tethered computing platform is outstanding. Two different frames have to be rendered, one for each eye. In premium devices, these resolutions reach at least 4K per eye which, assuming a compression ratio of 20, can sum up above 15 Mbits per frame. The input to the render process is the updated position of the virtual objects and the device, given by the VR engine, such as Unity or Unreal. The engine is also in charge of using the data produced by the aforementioned algorithms. State of the art devices are optimized to deliver rendering times below 5 ms.

TABLE I
SUMMARY OF INPUT AND OUTPUT SIZE PER FRAME, UPDATE FREQUENCIES AND ESTIMATED PROCESSING TIMES FOR THE ANALYZED VR PROCESSES

Algorithm	Processing Times (ms)	Frequency Hz	Input Rate Mb/frame	Output Rate Mb/frame
SLAM	7-10	30-60	3.82-6.33	$\ll 1$
Hand Tracking	8-12	60-90	3.82-6.33	$\ll 1$
Egocentric Segmentation	8-12	60-90	3.82-6.33	3.32-5
Rendering	< 5	60-90	$\ll 1$	14.92-19.9

III. Key Performance Indicators: Numerical Analysis

The goal of this section is to offer a numerical initial estimation of the required network key performance indicators (KPI) necessary for a successful wireless VR while preserving or even improving the immersiveness. Over the previous analysis, we highlighted the importance of the sensor data transmission in VR applications. Consequently, our target is to go one step further from the state of the art VR research an include the uplink and processing times in our study.

The maximum admissible motion to photon latency of 15 ms is the time deadline which the entire update procedure, including sensor acquisition, data processing, frame rendering and data transmission has to fulfill. This time deadline is even more constraint in devices running at 90 Hz or higher. We consider a very simple approach to estimate the transmission time of a single frame:

$$t_f = \frac{S_f}{T}, \tag{1}$$

where t_f is the transmission time of frame f in seconds, S_f the size in bits of f and T the effective throughput in bits per second. This throughput can be understood as the necessary peak throughput that needs to be sustained during t_f. After the frame is transmitted, the link is considered to be idle until the next frame is sent. As a consequence, our analysis is focused on the peak throughput which the network should be able to provide rather than an estimation of the necessary mean throughput. For simplification, we consider that T already includes all the possible sources of throughput loss such as packet loss, re-transmission or packet duplication, among others. For simplification, we consider a total latency L which involves both the uplink and downlink packet transmission latencies. The total network transmission time t_N can be written as:

$$t_N = \frac{S_u}{T_u} + \frac{S_d}{T_d} + L \tag{2}$$

where S_u and S_d are the frame size on the uplink and downlink sides respectively, and T_u and T_d the effective peak throughput. If we consider the processing time t_p the aggregated time of all the non-networked processes involved in the VR application, we can then express the total update time t, which has to be smaller than the device´s update period $1/f_d$, as:

$$t = \frac{S_u}{T_u} + \frac{S_d}{T_d} + L + t_p < \frac{1}{f_d}, \tag{3}$$

The processing times for the different possible VR offloading scenarios are hard to be accurately estimated in a general case: they depend not only on the characteristics of the specific set of algorithms used, but also on the hardware and software implementations. Consequently, our goal is to estimate the less restrictive sets of uplink and downlink peak throughput and latencies which satisfy Eq. 3, given different processing times. To achieve this, we propose a simple optimization problem which aims to maximize a reward function:

$$T_u, T_d, L = \arg \max_{T_u, T_d, L} R(T_u, T_d, L, t_p, S_u, S_d). \tag{4}$$

As we want to find the less restrictive set of peak throughput and latency KPIs, the reward function is designed to try to maximize the latency and reduce the peak throughput while increasing the total update time. The reward function is expressed as:

$$\begin{aligned} R(T_u, T_d, L, t_p, S_u, S_d) = {} & w_L \left(1 - \frac{L^{\text{max}} - L}{L^{\text{max}} - L^{\text{min}}}\right) + \\ & w_{T_u} \frac{T_u^{\text{max}} - T_u}{T_u^{\text{max}} - T_u^{\text{min}}} + \\ & w_{T_d} \frac{T_d^{\text{max}} - T_d}{T_d^{\text{max}} - T_d^{\text{min}}} + \\ & w_t \frac{t_N + t_p}{\frac{1}{f_p}}, \end{aligned} \tag{5}$$

where L^{max}, L^{min}, T_u^{max}, T_u^{min}, T_d^{max} and T_d^{min} are the maximum latency and peak throughput values and w_L, w_{T_u}, w_{T_d}, w_t weights to modify the reward function. These weights are used to adjust the importance during the optimization of each of the target parameters: T_u, T_d, L or t. We defined a set of constraints to solve the proposed optimization problem which can be summarized as:

$$\frac{S_u}{T_u} + \frac{S_d}{T_d} + L + t_p < \frac{1}{f_d} \tag{6}$$

$$kT_u^{\text{max}} \leq T_d^{\text{max}} \tag{7}$$

$$T_u \leq T_u^{\text{max}} \tag{8}$$

$$T_d \leq T_d^{\text{max}} \tag{9}$$

$$T_u^{\text{max}} + T_d^{\text{max}} <= T^{\text{max}} \tag{10}$$

$$L \geq L^{\text{min}} \tag{11}$$

For our analysis, we set the minimum latency L^{min} to 1 ms and T^{max} to 10 Gbps according to the initial capabilities of ultra-dense 5G networks. The initial deployments of 5G netowrks implement Time Division Duplex (TDD) method to separate the inward and outward signals. According to 3GPP TS 38.213 V.16.1.0 [22] standard, we can assume 4 times more slots allocated for the downlink signaling. As a consequence,

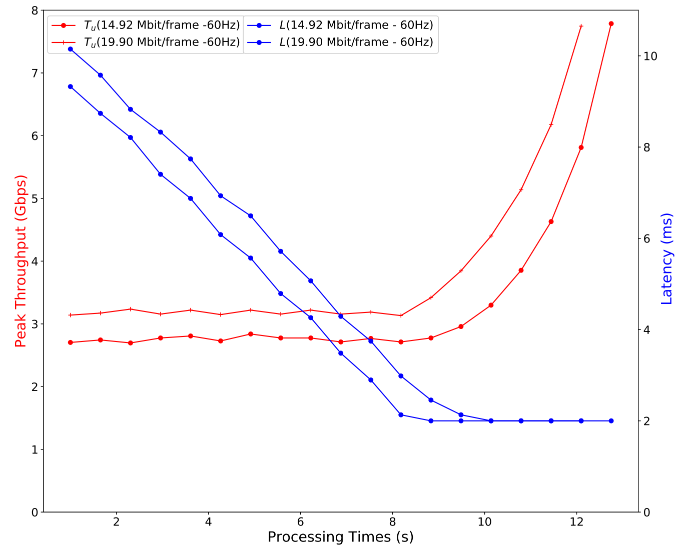

Fig. 1. Peak throughput and latency results for different values of processing times t_p for the rendering offloading scenario.

we assign 2 Gbps for T_u^{max} and 8 Gbps for T_u^{max}. Therefore, the value of k in Eq. 7 is set to 4. We solve the optimization problem for a set of t_p ranging from 1 ms to $(1/f_p - 2)$ ms, obtaining groups of peak throughput and latencies for each considered processing time.

IV. Experiments and Results

We use the previously described optimization problem to extract useful peak throughput and latency KPIs for different offloading scenarios.

A. VR Rendering Offloading

In this scenario we consider that the only offloaded process is the rendering step. In high resolution VR applications this is the most demanding step both in terms of data transmission and computational requirements, and can be considered our KPIs baseline. In this scenario, the uplink data rate is neglected as no media is transmitted, only virtual object updates and other metadata. On the donwlink side, the transmitted data is the rendered frame, which consists of two 4K frames, one for each eye. According to Table I, the downlink frame size lays between 14.92 and 19.9 Mbit. The update rate used for this first experiment is set to 60 Hz. For this first experiment we used $w_{T_u} = 0.25$, $w_{T_d} = 0.25$, $w_L = 0.25$ and $w_t = 0.25$ so no target parameter is prioritize during the optimization.

In Fig. 1 we can observe the output from the proposed experiment. As we described in Section II, the rendering times for high-end devices usually require less than 5 ms. For processing times below this value, we estimate round-trip transmission latencies above 5 ms. On the other hand, the downlink peak throughput estimated requirements are already quite severe, oscillating between 2.8 and 3.2 Gbps.

The 3GPP TS 38.306 V15.8.0 [23] standard includes a reference formula to calculate the supported maximum data rate by a radio access with a specific configuration. We can use this formula along with different standardized setups reflected in

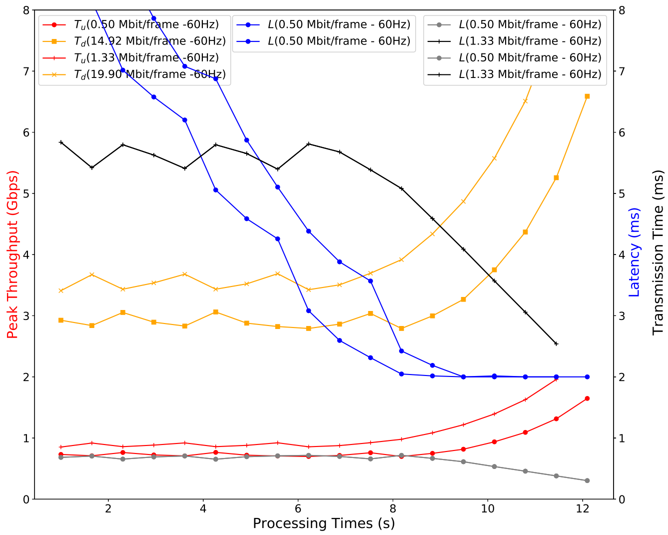

Fig. 2. Peak throughput and latency results for different values of processing times t_p for the rendering offloading scenario with uplink transmission of depth sensor data for real-time processing.

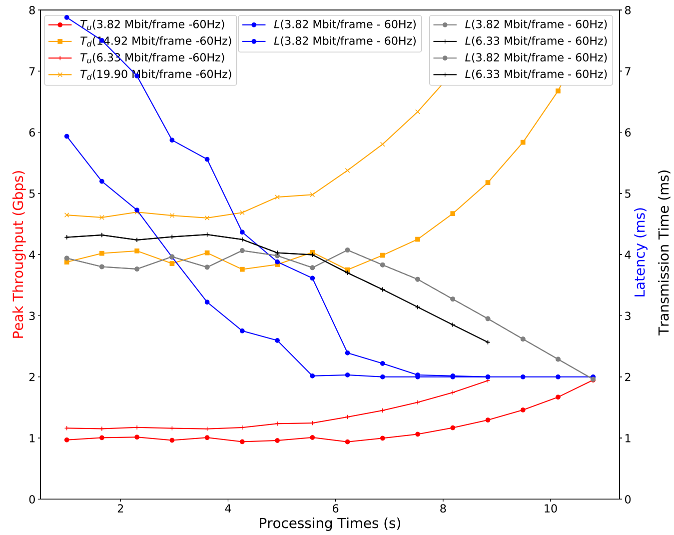

Fig. 3. Peak throughput and latency results for different values of processing times t_p for the full VR offloading scenario. Update rate: 60 Hz.

3GPP TS 38.101-1 V15.2.0 [24] and 3GPP TS 38.213 V.16.1.0 [22] to estimate the radio access configuration which can theoretically fulfill the estimated requirements. We propose a single component carrier network with a 11DL:1GP:2UL TDD configuration. Using numerology 3 or subcarrier spacing of 120 KHz , 400 MHz as the channel size and a 256 Quadrature Amplitude Modulation (QAM) the network can theoretically provide a downlink throughput above 3.5 Gbps if 2 multiple input multiple output (MIMO) layers are used. Besides, numerology 3 subcarrier spacing allows theoretical latencies below the millisecond for short range air transmissions.

B. VR Rendering Offloading and Depth Sensor Data Uplink Transmission

In this experiment, we consider the case in which the rendering is offloaded from the device along with another depth-based algorithm, such as hand tracking. We chose this scenario as it requires high uplink data rates without being as demanding as the full offloading scenario. Consequently, the downlink frame size is identical to the one described in the first scenario. On the contrary, in this case the uplink frame size is estimated to be between 0.5 and 1.33 Mbits. The update data rate is again considered to be 60 Hz. In this case, we used $w_{T_u} = 0.25$, $w_{T_d} = 0.25$, $w_L = 1$ and $w_t = 0.25$ to try to prioritize the latency optimization and target values some milliseconds above the low threshold.

Fig. 2 depicts the obtained values for this second experiment. From the analysis in previous sections we can expect processing values below 10 ms. We can observe in Fig. 2 how the required round-trip latency decreases below 3 ms, with minimum values above 9 ms. Besides, the peak downlink throughput remains stable around 3.2 Gbps for processing times below 9 ms. On the uplink side, we estimated peak throughput values around 0.9 Gbps. In both the downlink and uplink, the peak throughput starts exponentially increasing for processing values above 9 ms. In this experiment, we can also observe the black and gray curves which determine the time in ms for which the network must sustain the uplink and downlink peak throughputs respectively. Following the same procedure as in the first experiment, we propose a radio access configuration that can support the estimated requirements. Even though we now require 1 Gbps of uplink peak throughtput, the most suitable network configuration is the one described for the first experiment.

C. Full VR Offloading

In this last experiment, we propose the full VR offloading scenario in which the entire processing pipeline is offloaded from the device. In this scenario, the downlink is the same as in the previous experiments: the high-definition rendered frame. However, the uplink is transmitting now the full sensor data. We will only consider the stereo RGB and depth frames: other data, such as audio or IMU measurements can be neglected. In this scenario, we have an uplink frame size that lays between 3.82 and 6.33 Mbit. This is the most interesting scenario as it demands the maximum from the network at both the downlink and uplink sides. The used update rate is kept at 60 Hz. We also used the same weight values as in the previous experiment.

Fig. 3 shows the results from this last experiment. We can observe how the uplink peak data rate has similar values to the ones obtained in the previous values. However, the uplink transmission time is 4 times higher. On the other hand, the downlink peak throughput requirements have considerably increased with values around 4.5 Gbps. On the contrary, there are no considerable changes on the latency as the optimization problem is configured to keep the latency values high. In this scenario, the radio access configuration suitable for the previous results is not valid anymore. We exceed both the uplink and downlink peak throughput supported by the network. In this case, the radio access requires a 4 layer MIMO along with

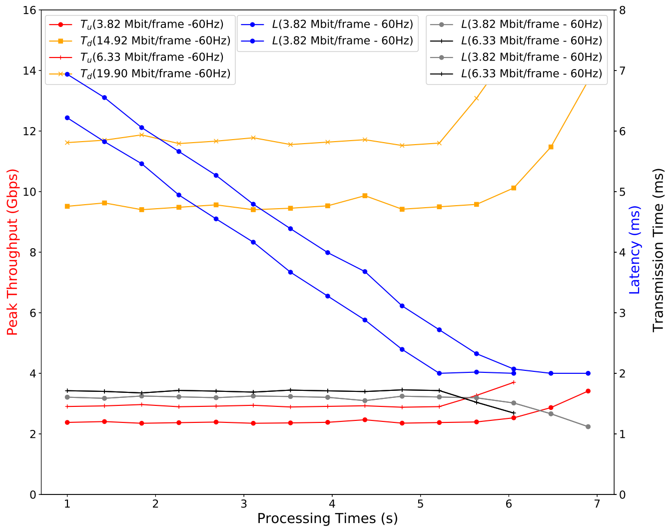

Fig. 4. Peak throughput and latency results for different values of processing times t_p for the full VR offloading scenario. Update rate: 90 Hz.

the same numerology, channel size and modulation as in the previous experiments.

Finally, we repeated the same experiment but setting the update rate at 90 Hz. We can observe in Fig. 4 that in this situation, the peak throughput requirements are too high to be fulfilled with the previously described configuration. In order to satisfy these requirements, it is necessary to try different carrier aggregation configurations or explore the usage of mmWave high-frequency bands.

V. Conclusions

In this paper, we have highlighted how 5G network improved capabilities can enable the successful development of unthethered VR technologies which can unleash VR's full immersive potential. To further analyze the actual requirements for VR offloading, we described the current state of the art of the most relevant VR algorithms, indicating their data flow and processing times estimation. With the purpose of extracting valuable peak throughput and latency key performance indicators for such processes, we proposed a simple optimization problem, which we used to extract a set of network requirements for 3 different VR offloading scenarios. Finally, we proposed the valid network configurations which according to the standards will fullfil the estimated requirements. In general, we can conclude that VR offloading is a demanding task which impose extreme requirements and even further development of 5G networks.

References

[1] Varjo-XR1 Pro product specifications. [Online]. Available: https://varjo.com/products/xr-1/

[2] K. Kilteni, R. Groten, and M. Slater, "The sense of embodiment in virtual reality," *Presence: Teleoperators and Virtual Environments*, vol. 21, no. 4, pp. 373–387, 2012.

[3] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Network*, vol. 32, no. 2, pp. 78–84, 2018.

[4] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5g wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[5] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[6] A. A. Al-Habob, A. Ibrahim, O. A. Dobre, and A. G. Armada, "Collision-free sequential task offloading for mobile edge computing," *IEEE Communications Letters*, vol. 24, no. 1, pp. 71–75, 2020.

[7] C. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *2017 IEEE Globecom Workshops (GC Wkshps)*, 2017, pp. 1–7.

[8] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for mec," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[9] J. Park and M. Bennis, "Urllc-embb slicing to support vr multimodal perceptions over wireless cellular systems," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.

[10] E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.

[11] E. A. McManus, B. Bodenheimer, S. Streuber, S. de la Rosa, H. H. Bülthoff, and B. J. Mohler, "The influence of avatar (self and character) animations on distance estimation, object interaction and locomotion in immersive virtual environments," in *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, ser. APGV '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 37–44.

[12] F. Argelaguet, L. Hoyet, M. Trico, and A. Lecuyer, "The role of interaction in virtual embodiment: Effects of the virtual hand representation," in *2016 IEEE Virtual Reality (VR)*, 2016, pp. 3–10.

[13] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2502–2509.

[14] M. Z. Zia *et al.*, "Comparative design space exploration of dense and semi-dense slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1292–1299.

[15] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric rgb-d sensor," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[17] J. Taylor *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016.

[18] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3213–3221.

[19] E. Gonzalez-Sosa, P. Perez, R. Tolosana, R. Kachach, and A. Villegas, "Enhanced self-perception in mixed reality: Egocentric arm segmentation and database with automatic labelling," 03 2020.

[20] P. Perez, E. Gonzalez-Sosa, R. Kachach, J. Ruiz, I. Benito, F. Pereira, and A. Villegas, "Immersive gastronomic experience with distributed reality," in *2019 IEEE 5th Workshop on Everyday Virtual Reality (WEVR)*, 2019, pp. 1–6.

[21] G. Bruder, F. Steinicke, K. Rothaus, and K. Hinrichs, "Enhancing presence in head-mounted display environments by visual body feedback using head-mounted cameras," in *2009 International Conference on CyberWorlds*, 2009, pp. 43–50.

[22] 3GPP, "NR; Physical layer procedures for control," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.213, 04 2020, version 16.1.0.

[23] 3GPP, "NR; User Equipment (UE) radio access capabilities," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.306, 01 2020, version 15.8.0.

[24] 3GPP, "NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.101, 07 2018, version 15.2.0.