

# Next-Generation Networking and Edge Computing for Mixed Reality Real-Time Interactive Systems

Susmit Shannigrahi  
Computer Science Department  
Tennessee Technological University  
sshannigrahi@tntech.edu

Spyridon Mastorakis  
Computer Science Department  
University of Nebraska Omaha  
smastorakis@unomaha.edu

Francisco R. Ortega  
Computer Science Department  
Colorado State University  
fortega@colostate.edu

**Abstract**—With the proliferation of head-mounted displays, cloud computing platforms, and machine learning algorithms, the next-generation of AR/VR applications require research in several directions - more capable hardware, more proficient software and algorithms, and novel network protocols. While the first two problems have received considerable attention, the networking component is the least explored of these three. This paper discusses the networking challenges encountered by the AR/VR community that experiments with novel hardware, software, and computing platforms in a real-world environment.

In this collaborative work, we discuss the current networking challenges both quantitatively (by analyzing AR/VR network interactions of head-mounted displays) and quantitatively (by distributing a targeted community survey among AR/VR researchers). We show that the cloud-provided network services are not ideal for the next-generation AR/VR applications. We then present a Named Data Networking (NDN) based framework that can address these challenges by offering a hybrid edge-cloud model for the execution of AR/VR computational tasks.

## I. INTRODUCTION

Virtual reality (VR) and augmented reality (AR) research has come a long way since Ivan Sutherland's first head-mounted display [1]–[3] and is becoming increasingly popular [4]. In particular, head-mounted displays using optical-see thru, such as the Magic Leap One headset [5] and Microsoft HoloLens 2 [6], [7] have shown a preview of the future of interactive computing. These devices enable users to seamlessly integrate their physical world with the virtual world, redefining how we interact with our surroundings.

Three primary moving parts make these applications possible. The hardware that gathers input for processing, the software (or algorithm) that usually runs on a local/remote site that processes this input and creates an output that is sent back to the device, and the network that connects the device to a remote processing location (when needed). There are several challenges in all three components that we must address to enable the next generation of AR/VR applications. While the first two have received considerable attention, we found networking to be the least explored of these three components. This work tries to define the networking challenges of AR/VR applications along with possible solutions.

Our contribution in this paper is threefold:

- First, we identify the networking needs of AR/VR researchers, both quantitatively and qualitatively. Towards this

goal, we conducted a community survey among AR/VR researchers to understand their networking needs ( $n = 14$ ). This survey provides us with insights into how researchers currently interact with the network and compute platforms, as well as the bottlenecks they encounter. We find that most researchers are disappointed by the latency and lack of quality of service in today's cloud services.

- Second, to complement the survey, we measure and dissect the latency of interaction with the cloud services using Microsoft HoloLens. We find the typical latency to be 250-500ms while the acceptable latency is around 50ms, pointing to an immediate need to reduce it.

- Third, learning from these exercises, we enumerate the networking challenges of the AR/VR community beyond just bandwidth and latency. Though previous works [8]–[10] have pointed out numerous networking challenges, our work points out additional challenges such as the need for an edge-cloud hybrid model. Unlike some of the previous work [9], the networking challenges we point out are not in the context of a specific application and meant to be generic. Finally, we present Low Latency infrastructure for augmented-Reality Interactive Systems (LLRIS), an NDN-based framework that can address these challenges.

The rest of this paper is organized as follows: in Section II, we give some background on mixed reality, in Section III, we present the results of our survey, while, in Section IV, we discuss a set of AR/VR use-cases. Section V discusses networking challenges for AR/VR, Section VI presents LLRIS, and, finally, Section VII concludes our paper.

## II. BACKGROUND

VR and AR are becoming popular [4]. In particular, head-mounted displays with AR software [5], [6] has the potential to change interactive computing. These devices enable users to seamlessly integrate their physical world with the virtual world, redefining how we interact with our surroundings. Current devices include Microsoft HoloLens (version 1 and 2) and Magic Leap One (Figure 1), and Google Glass is an AR device but with a stereoscopic head-mounted display. Other industry players are currently working on various devices - for example, nReal just announced their AR head-mounted displays to be priced at \$499 for consumers (no phone or computer included) and \$1199.00 for developers (with additional components).

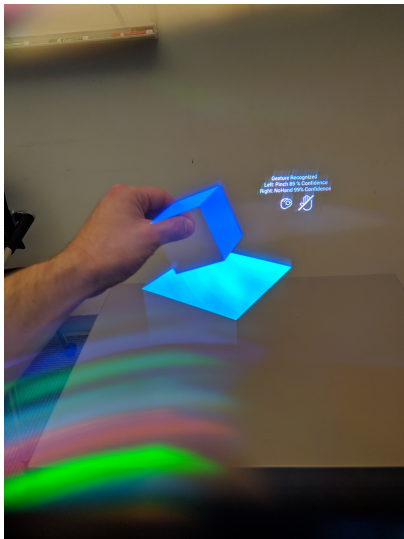


Fig. 1: Virtual object rendered by Magic Leap One

Mixed reality environments is a spectrum known as the VR continuum (Figure 2) and has four components: real-environment (the one humans are in), AR (adding virtual objects to the real environment), augmented virtuality (embedding real-environment objects into a virtual world), and VR (synthetic world). While all of them have specific uses, AR has the most potential to be an everyday device [7], [11], as an extension to the personal computer (desktop, tablet, phone).

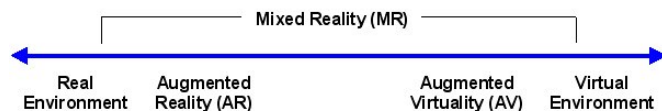


Fig. 2: Virtual reality continuum

An immediate use for AR head-mounted displays is enterprise applications. While Magic Leap and nReal are working to make the device pervasive among users, Microsoft has seen the current market potential to be enterprise software. Microsoft secured a \$480 million contract for HoloLens from the Department of Defense [11]. Microsoft announced further hardware development beyond its upcoming HoloLens 2. It seems that the future of interactive computing is in wearable optical-see thru devices, such as the Magic Leap One [5].

While the current trend for AR head-mounted displays is encouraging, important networking challenges remain to be addressed to improve the real-time interaction of interactive computing using this modality. While each application and problem domain may have different constraints, in general, interactive computing requires to be able to maintain low-latency responses. Low latency has been associated with poor performance [3], [12] in AR/VR systems. A very clear example was shown by Chris Wickens that for flight simulation delays of 50ms can degrade the operation while longer delays can lead to loss of control [3], [13]. Additionally, many other networking questions remain—how do we support the bandwidth requirements as these applications become more

pervasive? How do we offload computation to the edge for reduced latency and the ability to (partially) reuse previous computations? How do we design protocols that can support the performance and Quality of Service (QoS) requirements of AR/VR applications? What are the QoS requirements needed by the AR/VR then community?

Previous works have looked into these challenges but failed to capture the entire landscape. Braud et al. [8] looked into networking challenges but only in the context of mobile AR. While the end-devices will most likely be wireless, a sizable infrastructure component will still exist in the AR/VR ecosystems that connect the devices to computing platforms. Burke [9] provides a comprehensive list of AR/VR application requirements but does not enumerate the networking challenges explicitly. Other works [10], [14] have pointed out the problems either in a narrow context (5G or at the transport layer). Our work, on the contrary, looks at end-to-end networking requirements for AR/VR applications and presents a novel framework for addressing these challenges.

The motivation of our work is to set forth a set of questions and challenges for the networking community that can improve interactive computing. Our vision is to bring the interactive computing community (human-computer interaction, virtual reality, augmented reality, 3D user interfaces, and others) together with the networking community to understand the networking requirements better and adequately support them through novel network protocols. We envision such collaborations will engage the networking, multimedia, vision, and machine-learning communities for solving the critical problems listed in this paper (and other ones to come) and avoid silos. We also envision that these initiatives will be supported by a state-of-the-art testbed, where these communities can work together to develop the next generation AR/VR applications.

### III. A COMMUNITY SURVEY AMONG AR/VR RESEARCHERS

To better understand the problems, needs, and perceptions of networking among AR/VR researchers, we conducted a small scale targeted survey among 14 prominent AR/VR researchers. The use cases were diverse, providing a broad perspective into how the community uses the network for AR/VR research; application ranged from immersive education, immersive collaboration, HCI, Avatars and Embodiment, multimodal interfaces, 3D-UI, Robotics, social VR, VR chat rooms, applied perception, and B2B and consumer market research. The participants were from the USA and Europe.

The survey (somewhat unsurprisingly) pointed out that the latency was the most critical factor in AR/VR research. The researchers pointed out that the average latency for the AR/VR use cases must be below 50 ms, and some researchers pointed out latency requirements to be below 30 ms. Interestingly, one survey participant mentioned that 200 ms would be acceptable for low-quality 2D AR. However, the majority pointed out that this latency is very hard to achieve when the computations do not reside in the local network.

Another critical networking requirement that the survey highlighted was QoS guarantees. Surprisingly, the QoS requirements varied widely—from dedicated bandwidth requirements to low latency, from no (or negligible) packet loss to low jitter. Some participants pointed out that low packet loss would be beneficial but not at the cost of higher latency (we conjecture that TCP loss recovery plays a significant role in higher latency). Similar to the latency, the respondents pointed out the problems of QoS with the current networking stack and services that are available.

For the bandwidth requirements, the responses varied between <1 Gbps, 1-10 Gbps, and 10-40 Gbps, based on the use cases. More interestingly, the participants noted their disappointment with cloud services. To quote one response (and others reflect similar sentiment): "Cloud-based webpages currently are not consistently responsive even with a solid internet connection. If the cloud is not consistent enough for 2D apps, I have a hard time believing the quality of service will be consistently good enough for AR/VR needs for the foreseeable future." They noted that, in the cloud, the QoS parameters they need are impossible to achieve, and as a result, most prefer to install and use devices on local area networks.

Interestingly enough, though the researchers utilize local devices or local area networks, the majority points out that they have not investigated edge computing or an edge-cloud hybrid model for addressing the networking challenges. This observation points towards (a) the lack of suitable frameworks that can make such integration possible, and (b) an opportunity to create such a framework that can address the networking requirements (bandwidth, delay, and other QoS parameters) through an edge-cloud hybrid model.

#### IV. UNDERSTANDING AR/VR NETWORKING CHALLENGES QUANTITATIVELY

**Pilot AR-collaborative application:** To understand the problem further, we developed AR-collaborative and tested it at Colorado State University (CSU). This is an application where two users wearing Microsoft HoloLens 1 have the objective to collaborate in order to build a robot from multiple components (Figure 3). The application was developed using the Microsoft mixed reality toolkit combined with Microsoft Azure Cloud spatial markers and Photon Engine.

The spatial markers are critical for the placement of virtual objects. When using the Microsoft Azure Cloud (without the Photon Engine) with the spatial markers, the latency (in best-case scenarios) fluctuated between 250ms to 500ms. However, in many cases, it was common to see latency around 500ms and sometimes beyond this number. After adding the Photon Engine, which is used to sync additional data, latency marginally increased. The latency was problematic for the collaborative tasks, as users noticed the lag in the movements and actions when trying to move objects together. As stated earlier, anything above 50ms is not acceptable for interactive computing (in most cases). In particular cases, latency of 20ms becomes noticeable.

In order to circumvent this problem, we have started developing our own spatial markers. While they currently only operate for a limited set of scenarios and restricted to a local area network, our latency is between 2ms to 4ms. The next step is to find ways to distribute this open-source solution we are developing into multiple points, including the edge device, edge servers, and the cloud.

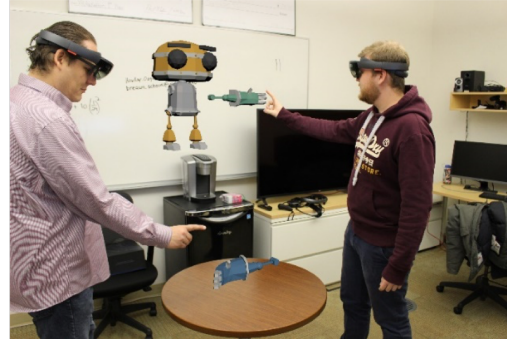


Fig. 3: AR-collaborative pilot application

**Streaming application:** In a second experiment, we conducted a trial with a depth-sensing camera input data. The camera sent the input from the device to a remote site for processing. The system captured and processed a frame every 0.33ms (or 30 fps) and used a window-based protocol to send the frame to the remote site. Note that unlike video streaming [15], where the viewer is passive, having a continuous stream of packets at a fixed interval is not enough for these applications. Instead, each gesture from the user must be processed within a fixed amount of time. When the processing is complete, the remote location sends back data (e.g., the input is recognized as someone playing tennis). For this experiment, we utilized remote servers at CSU, Denver, Sacramento, and Berkeley connected over a dedicated testbed.

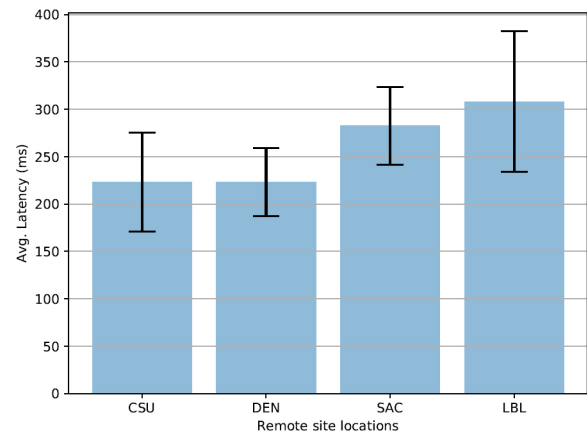


Fig. 4: Latency between a client and remote processing sites

In our experiment, we noticed a per frame delay of 200-300ms for sending the packets and getting data back. Note that we do not count the processing time in this calculation, which can add tens of milliseconds to this delay. The processing software we used had a per-frame processing delay of 30-40ms. The majority of the network delay (99%) was for

sending the input over the network (approximately 35KB per frame) but getting the data back was relatively faster due to the smaller size of the response, only a few bytes. Buffering the input and sending all at once is also not practical since it eventually increases the latency. As Figure 4 shows, even with dedicated servers and paths, the latency numbers are much higher than what AR/VR applications can tolerate. The solution, therefore, is to offload computation towards the edge of the network, where the network latency could be tolerable. However, offloading computation to the edge is severely challenging using TCP/IP, since it requires keeping track of the edge servers and their IP addresses, as well as the services each of these edge servers provides.

## V. NETWORKING CHALLENGES FOR AR/VR APPLICATIONS

Learning from our experiments and the community survey, we summarize the networking challenges and the desired network properties for AR/VR in this section—some of which are well-known (latency and bandwidth), though the underlying causes are not always obvious. The bandwidth challenge for VR/AR is well-known [8], [9], [13] and the ever-increasing amount of bandwidth in both wired (100Gbps fibers) and wireless networks (WiFi AX—11 Gbit/s theoretical maximum, and tens of Gbps with 5G) might be a candidate solution. However, bandwidth challenges might persist as the deployment of AR/VR becomes pervasive and available to users all around the world in the context of several daily tasks.

**Latency:** The most critical aspect of AR/VR is latency. AR/VR applications demand ultra low latency for real-time tasks. Several works [3], [13] have discussed that latency and bandwidth are important for AR/VR applications, though the degree of latency that an application can tolerate depends on the individual use case. For example, if we ask Amazon’s Alexa a question that is not cached in the local system, the response time from Alexa to go to the cloud and back is within a few seconds, an acceptable margin from a user in most cases. On the other hand, when a system needs to recognize multiple gestures, e.g., in the case of a gesture recognizer [16], [17], the latency of a few seconds might be unacceptable. While an argument might be that networking and AR/VR technology will continue to improve, the need to offload certain computational tasks will also continue as tasks become more complex (e.g., deep learning algorithms).

The only way to reliably reduce latency is to place the computation as near to the user as possible (a.k.a, the edge), only going to distant nodes as a last resort. AR/VR researchers are already putting computation near the devices by putting computing devices (from Raspberry PIs to dedicated servers) on the local area network. However, they also use commercial products (such as the Hololens) that are tightly coupled with cloud services. Currently, there is no way to combine these two modes easily. The trade-off between latency and computing capacity needs to be carefully considered; our solution presents an edge-cloud hybrid model that provides the researcher the

ability to use both local and remote computations and maintain a balance between latency and compute power.

**Reuse of previous computation:** Certain physical communication and processing constraints are not possible to overcome. A common way to further reduce latency and processing load is to reuse (partially or fully) already executed computations. When users perform similar gestures, the servers could reuse parts of previously recognized gestures to determine incoming gestures. The reuse can be for a single users (for example, the pre-rendered surroundings in an interactive game) or multiple users (interactive navigation in a city). From a networking point of view, the main challenge is how to forward “similar” tasks (e.g., gestures) to the same edge/cloud server (or set of servers), which could offer reuse of previously executed tasks, given that multiple available servers may exist. Another challenge would be to adequately name and describe these computations, so that can be easily reused—as far as we know, this is not done today in the AR/VR community.

**Mobility:** Nowadays, more mobile devices are sold than PCs [18]. It is imperative that AR users can move around freely without any QoS degradation. Additionally, the massive bandwidth and low latency expectations are going to challenge even the latest 5G technologies that aim to deliver 1ms end-to-end latency and more than 10Gbps to the end-devices [10]. Such a massive performance improvement is easier said than done. Enormous technological challenges such as the ability to properly utilize multiple disparate connections (WiFi, 5G) and the ability to seamlessly move from one endpoint to the other (handover) will need to be addressed. Moreover, though cloud providers have already begun to provision “edge clouds” with their distributed presence [10], the networking infrastructure (both at radio access and transport layers) is not ready to support such a model. What would be a good way to integrate computing and mobility directly into the communication network?

**Discovery of computation resources:** Multiple different servers with different hardware specifications may be available for the execution of application-defined tasks. AR applications may have specific hardware requirements for the servers that execute their tasks (e.g., certain CPU cores, CPU power, or memory requirements). As a result, applications may need to discover which servers can offer the computation resources necessary for their requirements. The network needs to ensure that the discovery process is seamless, adaptive to server failures, and identifies the server closest to the user that can execute the required task(s) most efficiently.

**Discovery of service availability:** As mentioned above, AR users should be able to move freely (e.g., in the context of a smart city) without any QoS degradation. The majority of existing frameworks assumes that applications are *a priori* aware of the service(s) they are looking for (e.g., gesture recognition). However, in a dynamic AR scenario (e.g., as users move in a smart city), it may not be feasible to assume that every single server can offer every single service. Moreover, a server may offer different versions of a service that could satisfy application requirements (e.g., gesture

recognition implemented through different algorithms). To this end, applications may need to discover the service(s) offered at the edge, which provide the functionality they are looking for, as well as the different versions of the offered service(s). This function could be combined with the discovery of computation resources. An alternative solution would be for the AR application itself to send the execution code to a server. However, this would put the burden on the application and the resource-constrained headset, having to be repeated every time a user connects to a different/new server.

**Secure and authenticated communication:** A fundamental challenge that needs to be tackled is how to provide secure communication between users and computation resources located at the edge of the network or on the cloud. Users also need to be able to establish trust and authenticate the servers they communicate with. Security is critical in cloud/edge enabled AR applications. For example, an AR application for visually impaired citizens can provide real-time navigation, text-to-speech translation, and other functions, posing a physical threat for users if compromised.

## VI. LLRIS - A LOW LATENCY INFRASTRUCTURE FOR AUGMENTED REALITY INTERACTIVE SYSTEMS

In this section, we present our preliminary directions towards creating LLRIS, a comprehensive NDN-based framework that addresses the challenges mentioned in the previous section. The overall LLRIS architecture is illustrated in Figure 6, while its components are discussed below. LLRIS will also enable the experimentation of the networking community with various NDN-based edge networking designs.

**Dynamic Edge Service and Resource Discovery:** LLRIS will feature a dynamic discovery mechanism, where devices will be able to seamlessly on-board an edge network and discover the edge services and resources available within this network [19]. To achieve that, edge devices and servers will run instances of a discovery daemon. The device discovery daemon will send Interests under a “/discovery” prefix, specifying the service functionality, a deadline by which the execution results need to be retrieved [20], and/or minimum resources that applications need in the edge network (Figure 5). Discovery Interests will be forwarded by the NDN network towards the closest edge server that runs a discovery daemon. The edge server daemon needs to be aware of the services and computing resources offered by all servers in the edge network. This can be achieved through an NDN distributed dataset synchronization protocol—the design of light-weight synchronization protocols will be explored in the context of LLRIS in order to amortize the cost of synchronization updates and investigate the tradeoff between the frequency of synchronization updates and the freshness of synchronization information.

**Edge service invocation and task offloading:** Once the services and resources that fulfill the application needs are discovered, LLRIS will provide a set of mechanisms that enable devices to: (i) invoke edge services by offloading computational tasks to edge servers; and (ii) pass input parameters required for task execution (e.g., a video frame to be

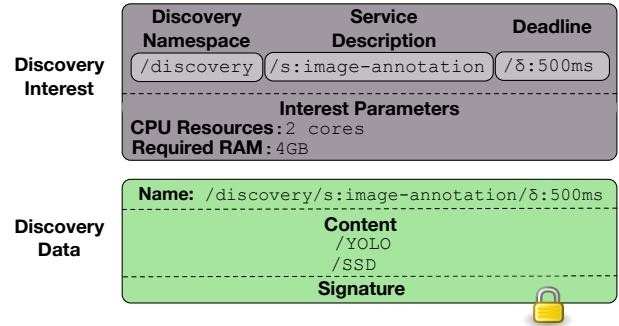


Fig. 5: Service and resource discovery Interest-data exchange

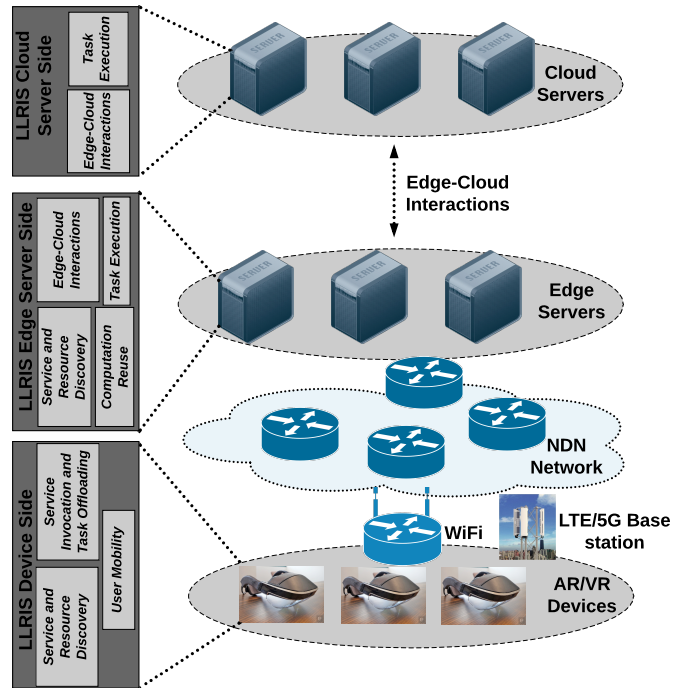


Fig. 6: LLRIS framework architecture

annotated) [21]–[23]. The adaptive NDN network will forward computational tasks towards edge servers that currently have available the resources necessary to execute the tasks. To achieve that, we are currently investigating mechanisms to make the network aware of the current conditions at the edge (e.g., CPU or memory utilization of edge servers).

**Computation reuse and mobility:** LLRIS will enable the deployment and experimentation with various NDN producer and consumer mobility support schemes, such as IPoC [24], MAP-Me [25], and KITE [26]. Furthermore, NDN naming schemes will be explored, enabling the network to forward tasks for potentially “similar” computation to the same edge servers, so that execution results of previous tasks can be (partially) reused. For example, a task “/annotation/CSU/CS/room=312/FOV=<FOV-1>” that requests annotation for an image taken in room 312 of the Computer Science (CS) department at CSU and a task “/annotation/CSU/CS/room=312/FOV=<FOV-2>” for the annotation of an image taken in the same room of the same building will be forwarded to the same edge server(s). This will maximize the chances that parts of the results of the former task will be reused during the execution of the latter

task if the Field of View (FOV) of these tasks overlap [27].

**Security and authentication:** LLRIS will utilize the NDN security principles [28] for mutual authentication between AR/VR devices and edge servers. Signed Interests will be utilized during the discovery and task offloading process, so that servers can authenticate devices, while the signatures of data packets will be utilized by devices to authenticate servers. Furthermore, LLRIS will enable the experimentation of the community with different trust models and secure device onboarding and task offloading designs.

**Hybrid edge-cloud model:** LLRIS will enable interactions between edge and cloud for optimal service placement at the edge based on various criteria and objectives as requested by users (e.g., service popularity, computational task complexity). Applications will send requests for computational tasks—for example, in the form of “/annotation/CSU/CS/room=312/FOV=<FOV-1>/ $\delta=50\text{ms}$ ”, where  $\delta$  is the deadline requested by the application for the reception of the task execution results. This request might be forwarded by the NDN network to an edge server that does not have the necessary computing resources for the execution of the task (e.g., the server is fully utilized at that moment or the task requires excessive resources for its execution). Depending on the value of  $\delta$  (i.e., how delay-tolerant the task might be), the server will further offload this task to another edge server (if an edge server that has the necessary resources is available) or to the cloud, possibly notifying the user application about this decision.

## VII. CONCLUSIONS AND FUTURE WORK

This work demonstrates the need for new networking protocols and edge-computing for AR/VR applications. We show that even with latest cloud platforms and hardware the amount of network latency the current applications face is unacceptable, pointing towards the need for a more suitable framework that aligns well with the requirements of AR/VR applications. This framework must address requirements such as low latency, reuse of computation, an edge-cloud hybrid model, and security.

We present an NDN-based framework, LLRIS, help with these application requirements. The framework provides support for service discovery, task offloading, compute reuse and caching, and an hybrid-edge cloud compute model. The work is but the first step in creating a comprehensive framework for AR/VR applications. We continue to work closely with the AR/VR community to create the appropriate protocols, deploy them for real applications within the framework, and study their performance.

## REFERENCES

- [1] I. E. Sutherland, “The ultimate display,” *Multimedia: From Wagner to virtual reality*, pp. 506–508, 1965.
- [2] —, “A head-mounted three dimensional display,” in *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, ser. AFIPS ’68 (Fall, part I). New York, NY, USA: ACM, 1968, pp. 757–764. [Online]. Available: <http://doi.acm.org/10.1145/1476589.1476686>
- [3] J. J. LaViola Jr et al., *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.
- [4] M. Bernard, “5 Important Augmented And Virtual Reality Trends For 2019 Everyone Should Read,” <https://tinyurl.com/yxwan52h>, 2019.
- [5] M. Leap, “Magic Leap One,” <https://www.magicleap.com/magic-leap-one>, 2019.
- [6] Microsoft, “Hololens 2 Overview,” <https://www.microsoft.com/en-us/hololens/hardware>, 2019.
- [7] T. Bishop, “Airbus and Microsoft team up to sell holographic tech to airlines, defense and aerospace companies,” <https://www.geekwire.com/2019/airbus-microsoft-team-sell-holographic-tech-airlines-defense-aerospace-companies/>, 2019.
- [8] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui, “Future networking challenges: The case of mobile augmented reality,” in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 1796–1807.
- [9] J. Burke, “Browsing an augmented reality with named data networking,” in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, July 2017, pp. 1–9.
- [10] T. Driscoll, S. Farhoud, S. Nowling et al., “Enabling mobile augmented and virtual reality with 5g networks,” Tech. Rep., 2017.
- [11] D. Grossman, “Microsoft Selling 100,000 HoloLens to U.S Army for \$479 Million,” <https://www.popularmechanics.com/military/a25346986/microsoft-hololens-army-deal/>, 2019.
- [12] S. Friston and A. Steed, “Measuring latency in virtual environments,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 4, pp. 616–625, April 2014.
- [13] C. D. Wickens, “The effects of control dynamics on performance.” 1986. Mar 2017, [Online; accessed 24. Jan. 2020]. [Online]. Available: <https://tools.ietf.org/id/draft-han-icrg-arvr-transport-problem-00.txt>
- [15] S. Mastorakis, P. Gusev, A. Afanasyev, and L. Zhang, “Real-time data retrieval in named data networking,” in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 2018, pp. 61–66.
- [16] P. Narayana, R. Beveridge, and B. A. Draper, “Gesture recognition: Focus on the hands,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5235–5244.
- [17] P. Narayana, J. R. Beveridge, and B. Draper, “Continuous gesture recognition through selective temporal fusion,” in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019.
- [18] T. H. Luan et al., “Fog computing: Focusing on mobile users at the edge,” *arXiv preprint arXiv:1502.01815*, 2015.
- [19] S. Mastorakis and A. Mtibaa, “Towards service discovery and invocation in data-centric edge networks,” in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–6.
- [20] S. Shannigrahi, C. Fan, and C. Papadopoulos, “Scari: a strategic caching and reservation protocol for icn,” in *Proceedings of the Asian Internet Engineering Conference*, 2018, pp. 1–8.
- [21] S. Mastorakis, A. Mtibaa, J. Lee, and S. Misra, “ICedge: when edge computing meets information-centric networking,” *IEEE Internet of Things Journal*, 2009.
- [22] M. Król et al., “Rice: Remote method invocation in icn,” in *Proceedings of the 5th ACM Conference on Information-Centric Networking*. ACM, 2018, pp. 1–11.
- [23] M. Król, S. Mastorakis, D. Oran, and D. Kutscher, “Compute first networking: Distributed computing meets icn,” in *Proceedings of the 6th ACM Conference on Information-Centric Networking*. ACM, 2019, pp. 67–77.
- [24] S. Shannigrahi et al., “Bridging the icn deployment gap with ipoc: An ip-over-icn protocol for 5g networks,” in *Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies*, ser. NEAT ’18. New York, NY, USA: ACM, 2018, pp. 1–7.
- [25] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng, “Map-me: Managing anchor-less producer mobility in content-centric networks,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 596–610, 2018.
- [26] Y. Zhang et al., “KITE: producer mobility support in named data networking,” in *ICN’18: 5th ACM Conference on Information-Centric Networking (ICN’18)*, 2018, pp. 1–12.
- [27] J. Lee, A. Mtibaa, and S. Mastorakis, “A case for compute reuse in future edge systems: An empirical study,” in *2019 IEEE Global Communications Conference (GLOBECOM) Workshops*. IEEE, 2019.
- [28] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang, “An overview of security support in named data networking,” *IEEE Communications Magazine*, vol. 56, no. 11, pp. 62–68, 2018.