Future Networking Challenges: The Case of Mobile Augmented Reality

Tristan BRAUD , Farshid HASSANI BIJARBOONEH, Dimitris CHATZOPOULOS, and Pan HUI
System and Media Laboratory,
Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
Email: braudt@ust.hk, farshidhss@ust.hk, dcab@cse.ust.hk, panhui@cse.ust.hk

Abstract—**Mobile augmented reality (MAR) applications are gaining popularity due to the wide adoption of mobile and especially wearable devices. Such devices often present limited hardware capabilities while MAR applications often rely on computationally intensive computer vision algorithms with extreme latency requirements. To compensate for the lack of computing power, offloading data processing to a distant machine is often desired. However, if this process introduces new constrains in the application, especially in terms of latency and bandwidth. If current network infrastructures are not ready for such traffic, we envision that future wireless networks such as 5G will rapidly be saturated by resource hungry MAR applications. Moreover, due to the high variance of wireless networks, MAR applications should not rely only on the evolution of infrastructures. In this article we analyze MAR applications and justify their need for accessing external infrastructure. After a review of the existing network infrastructures and protocols, we define guidelines for future real-time and multimedia transport protocols, with a focus on MAR offloading.**

I. Introduction

Mobile Augmented reality (MAR) is the research area that deals with integrating the physical environment with virtual objects for mobile devices [1], [2], [3]. Virtual objects are aligned over the physical world for the users to perceive the augmented information as part of their surrounding environment. Depending on the application, the number of virtual objects to be augmented varies. The hardware capabilities plays a major role in choosing which information to augment. Indeed, MAR-appropriate devices range from the low CPU power - small screen smart glasses such as Google Glasses or MadGaze glasses to high end holographic displays (Microsoft Hololens), including a wide range of smartphones and tablet PCs with disparate screen sizes and computational power.

Most of these devices employ a variety of sensors, enabling them to perceive orientation, acceleration, location, temperature, as well as recording audio and video. More importantly, they can connect to remote services and share collected data and exploit resources offered by cloud services. They can also serendipitously work together and exchange context-aware data. Device-to-cloud [4] and device-to-device [5] communication can be made by exploiting a wide range of communication channels: 3G, 4G, WiFi, Bluetooth, and even in some cases, NFC [6], each one presenting its own characteristics, neither of them meeting the MAR constraints.

As in many real-time applications, AR offloading is bandwidth and especially delay constrained. Connecting to external services can become extremely costly if not handled appropriately. In this article, we focus on how the communication channels can be exploited in order to offload AR computation to a distant machine, and provide guidelines towards transport protocol design and implementation in order for offloaded applications to meet the bandwidth and delay constraints in current and future networks.

This paper is organized as follows: In Section II we provide several definitions and general concepts regarding MAR and VR. We then focus on MAR applications and their requirements in terms of computation and resources. In Section III we summarize the requirements of MAR applications and give details on computation costs. In Section IV we cover the current and future Network infrastructures and shed light on several problematics that will strongly affect MAR applications in the following years. Section V lists available multimedia related network protocols and their limitations. Finally, in Section VI we propose our approach to tackling the networking challenges introduced by MAR.

II. Mobile Augmented Reality and Virtual Reality

Several definitions of Augmented reality (AR) and Virtual reality (VR) have been proposed [7], [8]. In this article, we will consider VR and AR to be defined as follows:

Virtual reality is an immersive technology which places the user into a virtual environment controlled by a set of given rules. The user is therefore partially to completely isolated from the physical world as the perception from one or several of its senses is replaced by synthetic stimulations.

Augmented reality has usually been defined in opposition to virtual reality. If VR isolates the user in a synthetic world, AR aims at supplementing the physical world through a virtual layer. The physical and virtual objects are therefore synchronously coexisting in an intermediate plane, at the intersection between virtual reality and telepresence [9].

Although both AR and VR can theoretically interact with all five human senses, most applications only consider audio and video, which are also the most resource and bandwidth-hungry components. MAR can be seen as an extension of AR

Platform	Computing power	Storage	Battery life	Network access	Portability
Smart glasses	very low	4-16 GB	2-3h	Bluetooth	high
Smartphone	low	16-128 GB	6-8h	Cellular/WiFi	high
Tablet PC	medium	32-256 GB	6-8h	Cellular/WiFi	medium
Laptop PC	medium - high	128GB - 2TB	2-8h	Cellular/WiFi/Ethernet	medium to high
Desktop PC	high	512GB - 2TB	unlimited	WiFi/Ethernet	none/dependent on network access
Cloud computing	unlimited	unlimited	unlimited	Ethernet/Fiber Optic	only dependent on network access

TABLE I: The basic characteristic of the devices that participate in a MAR ecosystem.

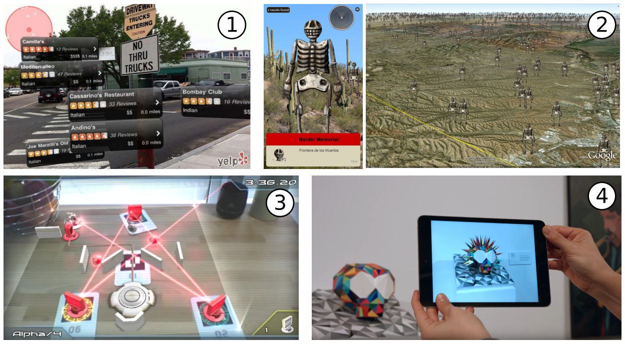


Fig. 1: Several usages of MAR. 1. Orientation (`Yelp`), 2. Virtual memorial (`Frontera de los Muertos`), 3. Video gaming (`pulzAR`), 4. Art (`Yunuene`)[1].

with the following additional requirements: *(i)* the application should run and be displayed on a mobile or a wearable device, and *(ii)* be interacted with in real-time.

MAR applications can encompass a wide range of use cases, from the very basic display of information or movies on smart glasses [10], to adding a complex virtual layer for gaming purpose [11]. In between those extreme cases, other applications may include artistic displays, work helpers, orientation-related operations, etc., each of them with specific requirements; several examples are given in Figure 1. MAR seems to be the most promising and most widely spread application of Augmented Reality. On the other hand, due to VR's immersive nature, the usage has to be limited to specific locations where the user does not put his or her life in danger. Given this sedentary constrain, network access may not pose as much a problem as for ubiquitous MAR. For the aforementioned reasons, even if the approach followed in this article may be applied to any real-time and multimedia application, including VR, our main focus lies in most challenging part of providing seamless service for resource-hungry AR applications in non-optimal network conditions.

III. Computation Aspects of MAR Applications

Mobility in an integral part of any MAR application. This constrains most of the MAR applications to devices that are easily and readily available to the user on the go, such as smartphones, smart glasses, and in general many wearable devices. MAR applications can be classified in various ways. Here we categorize MAR based on their resource needs. In this Section, we first describe the basic characteristics of the MAR applications, we then focus on their requirements and the limitations of the hosting devices.

A. *Characteristics of MAR applications*

In terms of data flow, we consider that a mobile application can be categorized as MAR if it has the following characteristics:

- **Input**: It utilizes data from various sensors of the device (camera, gyroscope, microphone, GPS), as well as of any companion device [12].
- **Processing**: It determines the information that is going to render on the screen of the mobile device. In order to do that it may require to access stored information locally in the device or in a remote database.
- **Output**: It projects its output to the screen of the mobile device together with the current view of the user (i.e., it augments the reality of the user).

AR glasses are the best option for ubiquitous MAR as the projected information is directly and permanently superimposed to the physical world. Yet, their computing power is so limited that most applications remain rather basic. Smartphones are also a suitable option, due to their higher computing power and portability, but require the user to point and hold the device to benefit from AR capabilities. Tablets, PC and laptops get cumbersome considering their larger size and limited mobility. The respective characteristics of those pieces of hardware are summarized in Table I.

B. *Requirements of MAR applications*

The most widely adapted devices for AR are also the least powerful due to their high portability. Depending on the generality of an application (storage, rendering capabilities, and connectivity to the Internet), parts of it may be executed in a cloud surrogate [13]. Vision-based applications [14] are almost impossible to run on wearables, and very challenging on smartphones [15] since they require capable GPUs. Some operations can run flawlessly only on a desktop computer or on a dedicated servers. Therefore, came the idea of executing

[1] https://www.yelp.com, https://www.layar.com/layers/bordermemorial
https://www.playstation.com/en-au/games/pulzar-psvita/
http://www.yunuene.com/wp/

the heavy computations to a distant but more powerful device, *i.e.,* computation offloading. A smartphone may works as a companion device to a pair of smart glasses, while in the end of the spectrum, it can be a virtual machine with almost infinite processing, memory, and storage resources. In between there are FoG [16] and device-to-device (D2D) solutions.

Other important requirements are the memory and the storage of the mobile applications. We formulate the resources of the mobile device with $\mathcal{R}_m$ and the ones of the cloud surrogate with $\mathcal{R}_c$. For example, MAR browsers are projecting virtual objects in the view of the mobile users [17]. The virtual objects need to be aligned with the objects in the 3D real world, which helps the user to perceive the virtual objects as if they are placed in the real world. This process involves matching the feature points of the environment against the ones with a perfectly aligned image of the objects detected in the camera view, namely homography [18]. In practice, in order to compute homography, a large database of real world images are collected and used for feature matching. In such cases, the MAR application cannot store all possible images of the objects to be detected due to limited storage on the device. Hence, the capabilities of any device become only limited by the network access due to its core importance in both computation offloading as well as in content retrieval. We use n_{mc} to describe the link between the mobile device and the cloud surrogate. Due to the potentially large amount of physical and virtual objects to process, offloaded MAR applications may require large amounts of bandwidth. Similarly, due to the real-time and interactive requirements of MAR, latency becomes an integral requirement. We denote the bandwidth of n_{mc} with b_{mc} and the latency with l_{mc}.

Several studies suggest that the human eye transmits around 6 to 10 Mb/s to the brain by taking into account that *accurate data is available only for the central region of the retina (a circle whose diameter is 2 degrees in the visual field)*. However, there is no way to isolate this area on video frames and full frames have to be processed. If we consider that the field of view of a smartphone's camera is between 60 to 70 degrees, a rough estimate of the amount of data to transmit is around 9 to 12 Gb/s. This estimate represents the upper limit of raw data that could be generated per second. In practice, an uncompressed 60FPS, 12 bits per pixel video with a resolution of 3840 x 2160 (4k) presents a bitrate of 711 Mb/s, which can drop to 20-30 Mb/s when compressed with lossy algorithms. We estimate the minimal bandwidth to be in the order of 10 Mb/s for a video feed with enough information to perform advanced AR operations. In the future, we can expect to observe higher resolution video flows and additional video feeds such as stereoscopic or infrared video, pushing this estimate to several hundreds of Mbps.

Regarding latency, real-time applications usually require one way delays around 100ms – between 75ms for online gaming and 250ms for telemetry [19]. However, due to several complications such as the alignment of the virtual layer on the physical world, a seamless experiences is characterized by notably lower latencies. Michael Abrash, Chief Scientist at Occulus VR, even argues that augmented reality (AR) and virtual reality (VR) games should rely on latencies under 20ms [20], with a holy grail" around 7ms in order to preserve the integrity of the virtual environment and prevent phenomenons such as motion sickness. Another study [21] displays the need for latencies below 10ms for a single HD streaming, while pointing out that some latency can be hidden behind mechanisms such as motion prediction. In the rest of this paper, we will therefore consider a maximal tolerable round trip latency of 75ms, while trying to minimize it as much as possible, especially since current studies show latencies greatly higher than those values [22], [23], [24].

Several specifically designed platforms perform offloading for AR operations. These platform focus on offloading the most computation intensive operations as it would save more time on the device and seemingly decrease the latency. For instance, `CloudRidAR` [13] locally performs feature extraction from the video flow. Only those features are then transmitted to the server. More recently, `Glimpse` [25] improves network efficiency by performing local tracking of objects and only offload a selected number of frames to the server.

Considering a MAR application a with $f^{(a)}$ frames per second generation and $p^{(a)}$ processing requirements per frame, it is required that enough resources on the mobile device are available to keep the execution delay of a:

$$P^{(a)}_{local}(\mathcal{R}_m, f^{(a)}, p^{(a)}) < \delta_a. \tag{1}$$

Where δ_a is a execution time constraint that guarantees in-time execution. Although, it depends on the application type, we can think about $\frac{1}{\delta_a}$ as a minimum frame generation rate. If we consider the case that a needs to access an external database with request rate $d^{(a)}$ and virtual object size $o^{(a)}$, Equation

$$P^{(a)}_{local+externalDB}(\mathcal{R}_m, f^{(a)}, p^{(a)}, d^{(a)}, o^{(a)}, b_{mc}, l_{mc}, x) < \delta_a.$$

Where x determines the subset of $o^{(a)}$ that is stored locally while the rest is downloaded from the remote cloud server. Of course caching and prefetching mechanisms can reduce the network overhead of $P^{(a)}_{local+externalDB}$. Given that $\mathcal{R}_m$ is not enough to guarantee in-time execution for a, and by considering computation offloading solutions, Equation 1 changes to:

$$P^{(a)}_{offloading}(\mathcal{R}_m, \mathcal{R}_c, f^{(a)}, p^{(a)}, d^{(a)}, o^{(a)}, b_{mc}, l_{mc}, x, y) < \delta_a.$$

Where x is the splitting parameter that determines which parts of $p^{(a)}$ will be executed locally. In case where the data are not located to the same surrogate as the one that is used for computation offloading the $P^{(a)}_{offloading}$ increases since it requires access to two different servers. It is easy to see that the parameters that affect the Equation 1 are b_{mc}, l_{mc}, x, y. x and y depend totally from the application type and the application developer while b_{mc}, l_{mc} depend only on the network access and the network protocols.

IV. Network access

In the previous section, we have justified that MAR applications are **(1)** computationally intensive, thus the need to offload heavy operations and **(2)** need to access data in remote databases. In order for the experience to be considered as seamless, the latency caused by communication with the remote servers has to be insignificant to the user. The bandwidth between the mobile user and the remote server depends on the networking infrastructure. Regarding latency, we consider a maximum round trip delay of 75ms. If we consider a 30 Frames per second video, the maximum tolerable jitter is 30ms in order not to skip a frame. Achieving these extreme requirements requires an improvement on the current network infrastructures and designing protocols specifically for AR and multimedia traffic. In this section, we analyze the state of present network architectures and the challenges faced by future wireless networks. We also shed light on several well known networking problems that seriously impact latency and bandwidth constrained communication.

A. Wireless Networks

Due to the high requirements in bandwidth and latency of MAR, we only focus on 3G, 4G, and WiFi networks. In order to reflect the everyday characteristics of those networks, most of the presented measurements come from companies such as SpeedTest or OpenSignal[2], which benefit from a large user database (several million users with several billion measurements). On the other hand, as the experimental conditions are often unexplained, those results will be complemented by academic peer-reviewed analysis where available.

1) HSPA+ (High Speed Packet Access): provides theoretically high throughput, between 84 and 168 Mb/s downstream, and 22 Mb/s on the uplink. However, the most common implementations in consumer market are limited to 21 to 42 Mb/s. Recent measurements in the United States [26] show an average download throughputs between 0.66 and 3.48 Mb/s with latencies between 109.94ms and 131.22ms. A study performed over 3 ISPs in Singapore [27] corroborates those results. The maximal downlink throughput revolves around 7 Mb/s while the upload is constrained around 1.5 Mb/s. Those throughputs exhibit large variations over time, with abrupt changes of several orders of magnitude. The latency reaches up to 800ms. In those conditions, HSPA+ is improper for any real-time multimedia application, especially not MAR offloading. Yet, it could be enough to transmit some lower priority data when no other network is available (e.g., connection metadata).

2) LTE (Long Term Evolution): was designed to improve the throughput and latency of HSPA, with expected downlink speeds up to 326 Mb/s, uploads around 75 Mb/s, and delays 50% smaller than HSPA+ [28]. The most optimistic sources give latency around 10ms [29]. In practice, if the improvement in throughput is clearly noticeable, with average downlink bandwidth between 6.56 and 12.26 Mb/s in the United States, the measured latency gain is not as strong as expected – between 66.06 and 85.03ms [26]. Speedtest displays average throughputs around 19.61 Mb/s on the downlink and 7.94 Mb/s on the uplink [30]. LTE is also widely deployed, partly thanks to the usage of lower frequencies which permits a higher range in rural areas. Currently, more than 98% of the United States population is covered [31]. Even if those results are not as high as advertised, the LTE upgrade is noticeable enough to enable the possibility for some real-time applications, including gaming and MAR.

3) LTE-direct: Another interesting feature of LTE is the presence of Device-to-device (D2D) communication through LTE Direct [32]. This so called **in bound D2D** allows mobile devices to intercommunicate in the licensed spectrum without the need for a cellular tower. The coverage radius is approximately 1Km with the data rate of 1 GB/s [33] and theoretically lower latencies. This solution could be particularly interesting for some delay-constrained operations. However, the technology is still young, and has not yet been deployed to the best of our knowledge.

4) WiFi: In parallel to mobile broadband networks, a user can exploit WiFi access points for a theoretically faster and more reliable Internet access. The two most widespread versions of the protocol are 802.11n and 802.11ac, with bandwidth respectively up to 600 and 1300 Mb/s. Yet, OpenSignal measurements present download speeds around 6.7 Mb/s for 802.11n and 33.4 Mb/s for 802.11ac [34]. This disparity can be explained by several factors. First, the theoretical maximum values are made to be reached in specific configuration cases, in a noise free environment, where the measured values correspond to an average over all users. Second, even if an access point (AP) may reach the gigabit per second, it may not be the case of the broadband network it is connected to.

The average reported latency of 802.11 remains around 150ms [35]. However, in a controlled environment (personal access point), delays can drop to a few milliseconds. The mobility aspect of WiFi is dramatically limited by several considerations: Open access points are too sparse to support a continuous transmission of data even during a small trip. Besides, handover is limited and can cause several seconds gaps without connection when changing access point. In 2012, a study [36] performed in a medium sized city in France showed that even if WiFi connectivity was available 98.9% of the time (99.23% for 3G), an Internet connection was only accessible 53.8% of the time due to the aforementioned problems. This study also confirmed the lower throughput values, around 55Kb/s for WiFi and 90Kb/s for 3G.

Another common issue in WiFi networks is the performance anomaly problem [37], [38], where the presence of users far away from an access point can dramatically degrade the throughput of users located closer to the AP as shown in Figure 2. When User A and B are both in the 54Mb/s area, they communicate with the AP and share the bandwidth equally. If User B moves in the 18Mb/s area, User A's throughput will fall around the same as User B due to the fact that User B occupies the channel longer to transmit the same amount of data, preventing A to transmit.

[2] http://speedtest.net/, https://opensignal.com/

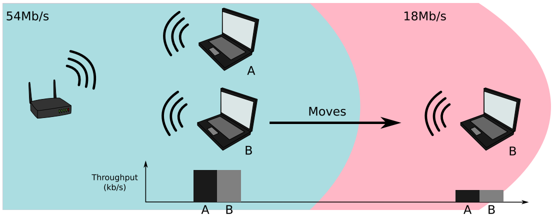


Fig. 2: The Performance anomaly problem.

Platform	Local Server	Cloud Server	University Server	Cloud Server
Connection	WiFi	WiFi	WiFi	LTE
Link RTT	8 ms	36 ms	72 ms	120 ms

TABLE II: The basic characteristic of the devices that participate in a MAR ecosystem.

5) WiFi-direct: Similarly to LTE, WiFi provides D2D communication through Wifi Direct, also called WiFi peer-topeer [39] and D2D outbound, as it is not functioning in the licensed spectrum. With WiFi direct devices communicate with each other without the need of an access point.

LTE-Direct is able to provide the most energy efficient communication scheme when the number of user is relatively high and it also has better detection of nearby devices, but in contrast, WiFi-direct presents a better energy efficiency in case of small amount of data [40]. WiFi-direct transmission range is 200 meters with a bandwidth of 500 Mb/s. However, as the authors of [41] have shown experimentally, the bandwidth depends strongly on the mobility of the users. On the other hand, WiFi-direct is a cheaper solution and it is available to almost every mobile device, while LTE-direct is not yet available to the end customers.

B. Measurements on Real MAR Infrastructure

In order to validate the wireless networks considerations, we measure the average latency of the `CloudRidAR` platform [13] in four different scenarios. Table II presents these results. In the first scenario, the server was placed in the same room as the user, with a direct WiFi connection. The latency is therefore very low – under 10ms. We then used the Google Cloud service with the nearest servers located in Taiwan and connected through the Eduroam network [3]. As the only accessible APs belong to Eduroam and the server is geographically close to our experiment platform, we can consider this situation as one of the most realistic scenarios for offloading to a cloud provider. The average link latency is approx. 36ms, which is enough to send more than 20 frames par seconds. We also tried to connect to a server located inside our university. Surprisingly, even if the distance between the server and the client is drastically reduced, the latency is almost doubled, which is enough for online games, but starts to get problematic for MAR. This phenomenon is probably caused by the infrastructure setup in the interconnection between Eduoram network and the university local network. Several equipments such as firewalls can introduce non-negligible delays in the network. Another lead would be the presence of congestion somewhere on the university network. Finally, we measure the latency of the application offloaded to the Google cloud through a LTE connection. The 120ms latency observed is even higher than the values reported in Section IV-A2, and definitely not suitable for AR applications.

C. The Challenges of 5G and Future Wireless Infrastructures

Considering the limitations of current wireless infrastructures, many hopes have been put into the future 5G infrastructures with Mobile AR being one of the concerns raised by the 5G White Paper [42]. AR is considered one of the future *pervasive and part of everyday life* applications and should be provided as a stable and uninterrupted service in densely populated areas. This article recommends the following general settings: a user should experience a minimum of 50 Mb/s at least 95% of the time in at least 95% of the locations. Data rates should be able to reach 1 Gb/s while end-to-end latency should be approx. 10ms in general, and 1ms for applications which require extremely low latency. In the specific use case of *Augmented Reality*, the 5G White Paper recommends the following Key Performance Indicators: 300 Mb/s on the downlink, 50 Mb/s on the uplink with 10ms end-to-end latency and seamless service between 0 and 100km/h.

Given these propositions, we add some requirements on the variance of those values. Indeed, in the case of MAR offloading most uplink data consists of pictures or videos, with a fixed constant rate. Even if it is always possible to adapt the transmission rate to the network status, no congestion control algorithm is prompt enough to accommodate the abrupt changes in throughput inherent to present wireless networks. Therefore, the round trip latency should not vary more than a few dozen milliseconds. Regarding the infrastructure, offloading servers should be positioned as close as possible to the mobile device to reduce latency, which is already a common practice for major cloud operators[4].

Even if those requirements are met, we can forecast that similarly to 4G, usage will quickly catch up with the capabilities of 5G, especially since the AR market is expected to boom in 2020. If nowadays, sending a 4K video through mobile networks seems delusive, we estimated in Section III-B that uncompressed flows could reach several Gb/s, and technologies such as holograms may put even more stress on mobile networks. Handling such large delay constrained traffic will be the challenge for further generations of mobile networks.

D. The Delicate Problem of Upload to Download Ratio

Offloading MAR to the cloud will often result in an uncommon network usage. Large amount of data need to be transmitted from the client to the server (images, videos,

[3] https://www.eduroam.org

[4] https://cloud.google.com/about/locations/
https://aws.amazon.com/about-aws/global-infrastructure

various sensor data, and points of interests), while the server mostly transmits the computation results and meta data. The resulting traffic thus requires higher bandwidth on the uplink than on the downlink. However, most asymmetric links present the exact opposite profile, with the downlink bandwidth several times bigger than the uplink bandwidth.

1) Past, present and future of access networks: Consumer-level Internet access started around 1992 [43] with dial-up modem access. However, wide adoption started with the apparition of broadband Internet offers (ADSL and Cable). These technologies provide higher bandwidth and lower latencies, but with asymmetric capacities, *i.e.,* a higher downlink bandwidth than on the uplink. In 2005, the number of Americans connecting to the Internet through broadband access exceeded the amount of dialup users and has been constantly growing until 2014 [44]. Even if optical fiber is progressively replacing the ADSL/Cable infrastructure, in some countries, it is not so uncommon to encounter asymmetrical offers. For instance, at the date of writing this article, the French ISP *Orange* was still providing asymmetrical (500 Mp/s down, 200 Mb/s up) optical fiber access to its customers [45]. In the US, many operators propose fiber-like cable access, with similar downlink bandwidth, but lower uplink bandwidth.

The current situation for Mobile networks is quite similar: 3G and 4G networks enabled a quick convergence of usages between mobile Internet and desktop Internet. This convergence became so important that in 2015, 56% of web traffic on the top 10,000 US websites was reported to originate from mobile devices [46]. Mobile broadband networks are characterized by a strong asymmetry, similar to broadband Internet access. Although 5G networks could provide more symmetrical bit rates through Full Duplex transmission, the current trend seems to lean towards dynamic asymmetric profiles [47], [30].

According to [30], the average user in August 2016 experienced download throughputs three times larger than the uploads. Among the top 6 fastest ISPs, only one displayed proper symmetric rates. The other ISPs exhibit asymmetry ratios between 3.31 and 8.22. Similarly, an average American user experiences a downstream to upstream ratio of 2.49, with values between 1.81 and 3.20 for the top 4 fastest mobile Internet providers.

2) Traffic evolution – Do we really need symmetric links?: In the early 1990s, downloads represented a volume about 10 times bigger than uploads. Most of the traffic was indeed composed of mail and web surfing and typical server to client applications. Later, the emergence of peer-to-peer and cloud storage started to inverse this balance. In 2012, the ratio between download to upload dropped to about 3 on an average day [48] and reached 2.70 in 2016. Due to the huge increase in usage of entertainment services such as Netflix or Youtube and the progressive recession of peer-to-peer traffic, both lead industry analysts and academics envision a more asymmetrical future [49], [48], notwithstanding the need for higher upload bandwidths. However, most links asymmetry ratios are far above the reported usage (Section IV-D1). Furthermore, the upload bandwidth usage growth displays a large variance [50], which makes predictions difficult.

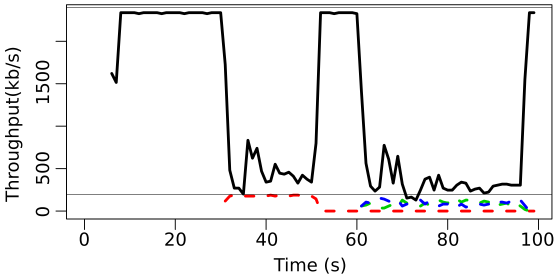


Fig. 3: Impact of uploads (dotted lines) on the throughput of a TCP download (continuous line) sharing the same asymmetric link. Figure from Heusse *et al.* [51]

According to Digi-Capital [52], the AR consumer market is expected to grow more that 20 times by 2020. If this tendency was to be confirmed, then it is likely that the required hardware and infrastructure would not be ready to support most computation-intensive vision based applications by that time. Cloud offloading will therefore be the only solution. We should thus expect and prepare for a large increase of upload traffic on asymmetric links. In such situation, several problems are likely to emerge: first of all, the uplink capacity is primordial for a decent MAR Quality of Service (QoS), and by extension a good Quality of Experience (QoE). Furthermore, it has been shown that for most congestion control protocols, congestion on the uplink could critically impact the performances of downlink connections, especially when the uplink buffer is oversized (which is often the case in mobile networks) [53], [51]. Figure 3 shows the impact of uploads on a single TCP download in a congested asymmetric link with oversized buffers. A MAR transport protocol should therefore focus specifically on optimizing the link utilization, while having minimal repercussions on the downloads sharing the link in the other direction.

V. Network Protocols

Network infrastructures are currently not ready for seamless mobile AR experience. Even if 5G could become a game changer, chances are that rapid usage would soon catch up with the technology. In August 2016, the average American download bandwidth was around 50 Mb/s [30]. According to CISCO analysts [54], 1 Gb/s will soon be required for some household applications, with 10 Gb/s “not being excessive”. Considering this phenomenon, only betting on the performance increase brought by 5G is, at best, delusive. We also can not afford to wait for 6G to solve the problematics rising in the next decade. Seamless Mobile AR will only be achieved through a smart although respectful usage of the different available links. In other words, a connection needs to maximize the available bandwidth by all means while being fair towards other connections. Yet, no transport protocol has been designed with Mobile AR in mind. However, several real time protocols may be a source inspiration for future AR protocols.

A. Multimedia Protocols

Several protocols have been designed specifically for video transmission. As MAR implicates large transfers of video, images, and data in the uplink direction, the multimedia network protocols could be a proper source of inspiration for an AR transport protocol. We discuss some of the most popular such protocols below:

1) RSVP (Resource Reservation Protocol) [55]: is used to reserve resource on a network. This protocol is used by hosts and routers to request and provide quality of service guarantees on specific flows. Although not directly applicable to generic AR applications, the possibility to provide QoS guarantees on specific AR applications could be a commercial argument for mobile broadband operators.

2) RTP (real-time Protocol) and RTCP (real-time Control Protocol) [56]: both run on top of UDP. RTP facilitates the transfer of real-time data over UDP while RTCP adds a layer of QoS. On top of the traditional in-order delivery, those protocols feature several useful characteristics for AR applications: **(1)** Jitter compensation mechanisms. **(2)** Inter-media synchronization, which permits to receive content from different sources. **(3)** QoS informations are reported to the application, which can thus adapt the video quality, transmission rate or other parameters to the network conditions.

3) MPEG-TS (MPEG Transport Stream) [57]: also provides stream synchronization, with the possibility of interleaving several streams together. It also provides forward error correction (FEC) to recover from lost or damaged frames.

4) D2D Multimedia Protocols: In the context of D2D communications, there are no proposals dedicated to multimedia transmission only between mobile devices. However, in the context of mobile ad hoc network various proposals focus on the end-to-end path establishment. Fu *et al*, for example, proposed a variation of (TCP) friendly rate control (TFRC) for multimedia streaming [58], while H. Luo *et al* extended this work to a real-time video streaming rate control protocol [59]. However, these works are not suitable for the MAR applications that need to access a remote server.

B. Improving General Performance

Other generic transmission protocols have been designed to improve overall performance and maximize available links utilization.

1) Multipath TCP [60] and SCTP (Stream Control Transmission Protocol) [61]: are two protocols developed to exploit multiple different paths and enable multi-homing. Using multiple paths has two main advantages: **(1)** Maximize resource utilization by combining the capacities of 4G and Wifi networks in order to meet MAR bandwidth requirements. **(2)** Provide redundancy. This has been successfully used in order to enhance the handover process in WiFi [62].

2) Quick UDP Internet Connections (QUIC) [63]: provides a complete suite of features on top of UDP and combines functionalities from TCP, Multipath TCP, TLS, and HTTP protocols.

3) Datagram Congestion Control Protocol (DCCP) [64]: provides congestion control without reliable in order delivery. New data is always preferred to former data for transmission.

C. Discussion

There does not seem to be an optimal network protocol solution for Mobile AR applications. In terms of Network infrastructure, most current architecture present bandwidths and latencies significantly below the minimal requirements for MAR application. 5G networks may provide a major temporary improvement until usage catches up with the sudden increase of performance. Moreover, commercial deployment of 5G networks is not planned before 2022, while the AR market is expected to flourish before 2020. We therefore expect applications requirements to exceed 5G networks by 2025, while the next generation won't become a reality before 2030.

Another point to take into account regarding mobile broadband network is the cost for the user. Most mobile networks continue to be expensive to user. We can expect the user to be reluctant to transmit large amounts of data for the sake of a seamless MAR experience. Therefore, we cannot rely only on mobile broadband networks to provide MAR offloading.

VI. Towards an AR-oriented Transport Protocol

Designing a transport protocol oriented towards MAR applications will represent a colossal challenge. Exploiting every single communication channel maximally will be necessary in order to have an efficient and effective offloading. We need to rethink the general architecture of offloading systems, and come up with more distributed solutions. The guideline we provide in this section is focused on MAR applications, which are also generalizable to any computation offloading (e.g., VR), or delay/bandwidth constrained mobile applications as in mobile games. We envision the following properties: **(1)** *Classful* traffic: MAR applications may transmit various types of data, with diverse requirements (namely bandwidth, latency, and integrity) and priorities. **(2)** *Fair* to other connections while exploiting the maximum available bandwidth. **(3)** *Low latency* and *fault tolerant*: some selected data should be privileged to retransmission in order to maintain real-time communication. **(4)** *Multipath*: exploiting more links to maximize bandwidth and minimize delays. **(5)** *Distributed* or *Semi-distributed*: multiple servers and device-to-device communication for optimal performance. **(6)** *Secure* in terms of user privacy.

A. Classful Traffic

MAR can generate several types of traffic with various requirements: audio and video, connection meta-data, computation results, etc. Several traffic classes have to be defined in order to accommodate with the various QoS levels required. We consider the following three classes as the baseline: *1) Full Best effort*: for data where latency is the most important parameter and new data is preferred to loss recovery. This includes most of the sensor data sent on the uplink. *2) Best effort with loss recovery*: sensitive data with latency requirements (e.g.,

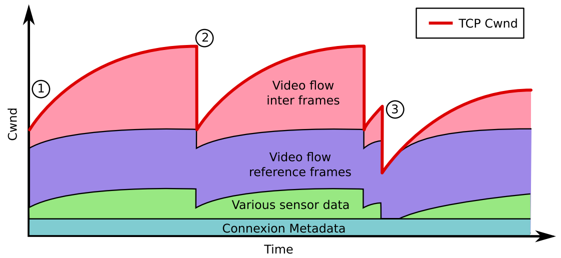


Fig. 4: TCP's congestion window versus graceful degradation.

reference frames of a video stream). *3) Critical data*: where reliable in-order delivery is preferable to latency. This includes most connection metadata.

Due to the huge variety of traffic that may be generated by a MAR application, some other intermediate traffic classes can be defined, with various degrees of latency, bandwidth, and reliability requirements.

In parallel to those traffic classes, the application should set traffic priorities. Indeed, in case of congestion or poor network connectivity, a MAR application should support graceful degradation, *i.e.,* discarding or delaying some traffic while waiting for the situation to improve. At least four priorities should be defined: **(1)** *Highest priority*: data should neither be discarded nor delayed if any other traffic is present on the network. **(2)** *Medium priority 1*: data can be delayed but should never be discarded (ex: data that belongs to the *Critical data* traffic class). **(3)** *Medium priority 2*: data should be discarded but not delayed. For example, new data in a video stream constantly replaces former data, and in-time delivery is more important than the integrity of the flow (as long as the reference frames are transmitted). **(4)** *Lowest priority*: this type of flow can be completely discarded in case of congestion.

For each priority, various levels may be defined to precisely describe the order in which service should be reduced.

B. Congestion Control, Fairness and Graceful Degradation

Due to the amount of traffic generated by MAR, uncontrolled data transmission would be catastrophic to other connections sharing the network. Implementing an efficient congestion control algorithm is the number one priority in order to respect the fairness at use in today's networks. For MAR, in-time reception remains the absolute priority. In order to achieve such result, communication between the protocol and the application is primordial. Contrary to TCP, when congestion is encountered, it is not possible to reduce a congestion window to send less data. In the previous section, we introduced the concept of *graceful degradation*, around which the congestion control algorithm should gravitate. An AR application should ideally function with degraded performance even if no network connectivity is available. In case of network congestion, a choice has to be made in order to decide which traffic to discard or delay, for similar results to TCP.

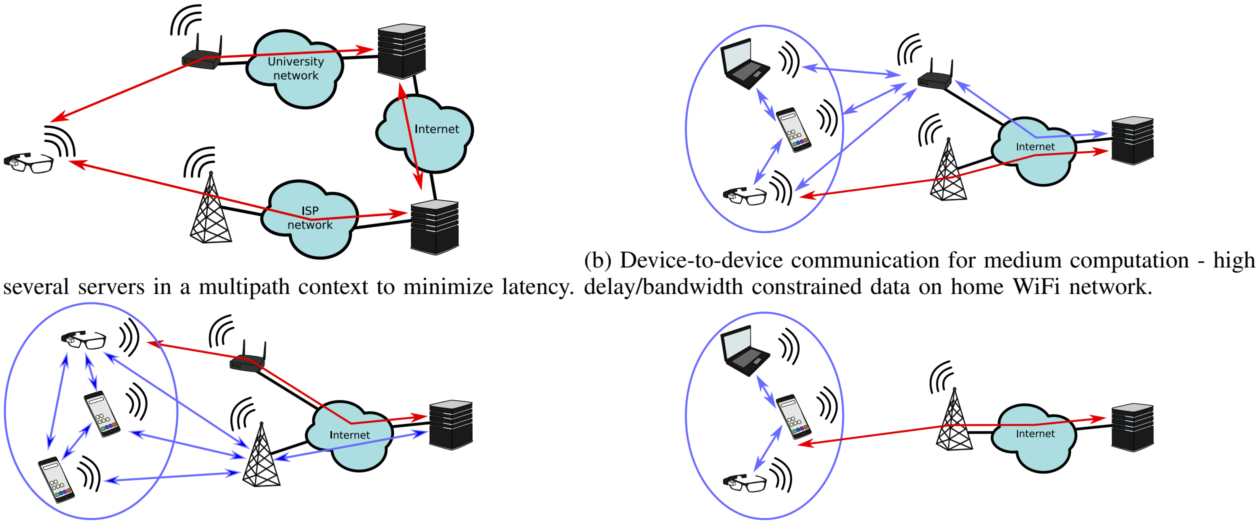
Thanks to traffic classes and priorities, the application can signal to the protocol which data to discard in case of congestion, while continuing to send the most crucial informations. Additionally, the protocol can provide QoS information to the application. In case of congestion, the application can lower the video quality, the number of samples, etc. The user experiences is therefore uninterrupted, although degraded.

Figure 4 summarizes the ideas behind TCP-compatible graceful degradation. For the purpose of the example, we chose an AR flow with four types of traffic: *Connection metadata*: Those data are constantly generated and should not be lost or delayed. They belong to the class of *critical data* with *highest priority*. *Various sensor data*: position, orientation, etc. They do not account for much traffic and can be used as an adjustable variable. They belong to the *Full best effort* class, with *Medium Priority 1*. *Video reference frames*: those frames are primordial to decode the video flow. They should not be lost or delayed. They belong to the *Best effort with loss recovery class*, with *Highest Priority*. *Video interframes*: those frames are encoded using the reference frames. They will be our main adjustable variables. They are *Full best effort* with *Lowest priority*.

We identify 3 different situations on Figure 4: **(1)** without any loss, TCP's congestion window grows linearly. Similarly, the amount of data sent by the application grows linearly. By modifying the number of samples of sensor data and the quality and amount of interframes in the video, the algorithm can try to fill the link without impacting the transmission of essential data (e.g., video reference frames and Connection metadata). **(2)** when a loss happens in TCP, the congestion control algorithm divides the congestion window in order to reduce the transmission rate. In the case of MAR, the algorithm should select which data to stop sending. A compromise is reached by drastically reducing video interframes and sensor data. Connection metadata and video reference frames are not impacted. **(3)** following a change in network conditions, another loss happens. TCP divides its congestion window once again to meet the new network requirements. In our case, stopping the transmission of the sensor data and video interframes is not enough to reach the requested rate. Connection metadata should be unaltered at all cost. In this scenario, the algorithm can temporarily reduce the quality and number of reference frames. The user experiences a severely degraded but functional service. In terms of design, the congestion control algorithm should closely monitor latencies and react accordingly. A sudden rise of delay or jitter should be treated as a congestion indication, with immediate reaction. This should be especially true for data sent on the uplink in order not to interfere with existing downloads. Yet, this strategy may result in unfairness toward the connection when competing with multiple other flows [65]. A trade-off has to be found between the latency and bandwidth requirements.

C. Loss Recovery and Low Latency

As recovery is costly in a latency-constrained context, the protocol should ideally avoid recovery from losses. For instance, TCP takes at least one (usually 2 to 3) round trip time to detect and recover from a loss . If the application generates 30 Frames per Second, with maximum tolerable latency no higher than 75ms, we can afford to recover a single lost frame



(a) Using several servers in a multipath context to minimize latency.

(b) Device-to-device communication for medium computation - high delay/bandwidth constrained data on home WiFi network.

(c) Device-to-device communication for low computation - high delay/bandwidth constrained data using LTE Direct.

(d) Device-to-device communication for medium computation - high delay/bandwidth constrained data using WiFi Direct.

Fig. 5: Several approaches to distribute computation among resources. The pair of smart glasses represent the device with the lowest computing power, that offloads AR operations to other devices.

only if the round trip time is at most 37.5ms. Considering that the average end-to-end latencies for 4G and Wifi are around 80ms and 150ms respectively (see Section IV-A) , loss recovery is not possible without a large service degradation. Some traffic may not be sensitive to losses. In a video flow using inter-frame compression, only reference frames should be protected as they are essential to properly decode the stream. However, if too many of the other frames are lost, the resulting flow may become too degraded to extract useful information. It would be preferable to introduce some redundancy in the data flow either by performing network coding, forward error correction [66], [67], or by exploiting multiple paths [68], [69]. Nevertheless, most of those solutions add a non-negligible amount of information to transfer on links where resources are sparse, forcing one more time to strike a compromise between latency and bandwidth.

D. Multipath

Due to the high bandwidth and low latency requirement of MAR, exploiting a single link may not be sufficient to provide the expected QoE. An AR protocol should provide the possibility to exploit multiple paths simultaneously.

Utilizing 4G and Wifi could indeed introduce a significant improvement here. Firstly, the resulting available bandwidth increases, partially fixing the high data rates requirements. Secondly, latency constrained data can be sent on the link with the lowest observed round trip times. Data packets belonging to a traffic class with loss recovery could also be sent on both links in order to prevent a costly recovery process. Finally, multipath has been proven to efficiently reduce the time without connectivity when performing handover [70].

4G networks are usually expensive for the end user, with limited amount of data included in the base plans. We envision three behaviors: **(1)** WiFi all the time, 4G for handover: permits an user with lower-end data plans to experience seamless MAR while barely using LTE networks. **(2)** WiFi most of the time, 4G for handover and when WiFi is not available: using this approach, a user can experience almost 100% service while keeping his LTE data usage low. **(3)** WiFi and 4G: by using both WiFi and LTE simultaneously, latency and bandwidth can be significantly improved.

E. Distributed

After using multiple paths to reach a single server, the next step would be to offload data to different servers, depending on latency and throughput requirements. Figure 5 shows an example of how using multiple servers could be beneficial as an extension of *edge computing*, where the nearest server would be selected for a given path. On Figure 5a, the user exploits different servers while utilizing multipath communication. For instance, when connecting to a university's WiFi network, it may be preferable to offload to the university server, while the connection using 4G through a broadband mobile operator may contact a different server. This ensures lower delays with less jitter. However, servers should be interconnected in order to process data efficiently. The question of inter-server synchronization remains with the need for n-way synchronization (n being the number of servers). The connection between multiple servers may present medium to high latency depending on the interconnection, firewalls, and other policies at work on the link.

Another idea is the usage of Device-to-device communication for the most constrained information. This could be particularly useful for low end hardwares such as smart glasses, where even simple feature extraction can considerably slow down the process. In this situation, other nearby smartphones could assist by sharing their available processing power. Figure 5b shows the possibility of using a home WiFi network to offload some computations to the user's smartphone and/or computer, while simultaneously offloading less latency constrained operations to a cloud server. Figures 5c and 5d show how to respectively use LTE Direct and WiFi Direct for offloading latency sensitive informations to other devices.

F. Locating Edge Datacenters for MAR Applications

As justified in the previous sections, the current protocols and the existing architectures can not provide guarantees in communication delay in the similar way as the classic quality of service guarantees. To handle this problem, network providers have to install edge datacenters close to the users that will be able to handle the MAR offloading requests. Mathematically, this problem can be formulated, in an abstract way, as follows:

$$\begin{gathered} \min_{\mathcal{C}} |\mathcal{C}| \\ \text{subject to:} \\ P^{(a)}_{offloading}(\mathcal{R}_m, \mathcal{R}_c, f^{(a)}, p^{(a)}, d^{(a)}, o^{(a)}, b_{mc}, l_{mc}, x, y) < \delta_a. \\ \forall c \in \mathcal{C}, \forall m \in \mathcal{M}, \forall a \in \mathcal{A} \end{gathered}$$

where $\mathcal{C}$ is the set of the datacenters and each datacenter is characterized by its location and its resources. $\mathcal{M}$ is the set of the mobile users and $\mathcal{A}$ is the set of the applications.

G. Security and Privacy

Finally, significant effort should be put on security. Indeed, as AR applications transmit audio or video feeds from a camera, user privacy is primordial. Heavy usage of cryptography should be performed for every communication. In a D2D context, data offloaded to other users devices should not be recoverable. It must be anonymized in case a malicious user tries to access those elements. Every element which may lead to recognizing locations, people or other personal information should be removed from this data. For instance, in the case of a picture, at least faces, license plates and visible street plates should be blurred before sending to other users for processing. Several studies focus on which confidential data may be leaked by AR application and mobile picture taking in general, and various solutions have been proposed. Among them, `PrivateEye` and `WaveOff` [71] allow the user to manually define zones on captured pictures that are safe to release to an app. `PrivacyTag` [72] uses physical tags to protect users privacy. `I-PIC` [73] is a full software solution where users can define levels of privacy, protecting their visual features. The article also defines a secure protocol to transmit privacy-sensitive information. A trade-off needs to be found between the user's privacy and the amount of personal data required for proper behavior of the application.

H. Notes Regarding Implementation

The actual implementation of this protocol may be done on top of UDP at the application level, making it easier to integrate in applications as an external library. In the hunt for latency, it may be interesting to integrate the protocol directly in the kernel of the operating systems, for faster processing, but also to spare unnecessary application-kernel communication latencies. Another point to take into account resides in the network buffers implemented in the kernel. An appropriate queuing policy may be designed in order to favor the MAR traffic, while providing low delays for the other connections on the uplink. Indeed, the uplink buffer implemented in the kennel is usually oversized (around 1000 packets), dramatically increasing the overall latency. This result may be achieved by a combination of latency queuing and low priority queues such as FQ CoDel [74]. Although some works seem to show promising results [75], as bandwidth and delay are strongly variable, more studies are needed to assert this affirmation. Regarding queues, it has to be noted that most fair queuing policies in the network may be strongly detrimental to MAR flows by delaying longer flows, sometimes up to starvation.

VII. Conclusion

Even if newer technologies may not be enough for the heaviest ubiquitous MAR applications, we discussed several leads in how next-generation multimedia transport protocols should be designed to get as close as possible to their requirements, with a specific focus on Augmented Reality. Current and future wireless networks will barely withstand the amount of traffic generated by MAR offloading. Future network may provide the boost in performances required for enabling serious offloading, but will quickly get caught up by the usage, especially without AR-specific congestion control. Finally, MAR offloading reverses the original asymmetric paradigm, by sending more data on the uplink (from the mobile device to a cloud server) than receiving on the downlink. This behavior may cause some serious performance issues for other connections sharing the same asymmetric bottleneck, including other interactive applications such as web surfing. After looking at those problematics in networks and MAR, we concluded that we could not only rely on the evolution of network infrastructures, both short and long term. However, several constrains can be undertaken through specific choices in the transport layer. We therefore defined some new guidelines regarding the design and implementation of MAR transport protocols to significantly improve the overall performances of offloaded applications. These guidelines, organized around five main axis, all aim at enhancing the latency and bandwidth observed by offloaded AR applications, while keeping a high level of fairness towards other connections. They are designed to suit any kind of current or future networking infrastructure, including 5G and Edge computing. They can be extended to most offloading operations (for instance Virtual Reality), multimedia transfer, or even online gaming.

VIII. Acknowledgements

This research has been supported, in part, by General Research Fund 26211515 from the Research Grants Council of Hong Kong, Innovation and Technology Fund ITS/369/14FP from the Hong Kong Innovation and Technology Commission.

References

[1] T. Höllerer and S. Feiner, "Mobile augmented reality," *Telegeoinformatics: Location-Based Computing and Services. Taylor and Francis Books Ltd., London, UK*, vol. 21, 2004.

[2] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE computer graphics and applications*, vol. 21, no. 6, pp. 34–47, 2001.

[3] Z. Huang, P. Hui, C. Peylo, and D. Chatzopoulos, "Mobile augmented reality survey: a bottom-up approach," *arXiv preprint arXiv:1309.4413*, 2013.

[4] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Infocom, 2012 Proceedings IEEE*. IEEE, 2012, pp. 945–953.

[5] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[6] K. Sucipto, D. Chatzopoulos, S. Kosta, and P. Hui, "Keep your nice friends close, but your rich friends closer–computation offloading using nfc," *arXiv preprint arXiv:1612.03000*, 2016.

[7] R. A. Earnshaw, *Virtual reality systems*. Academic press, 2014.

[8] A. Wexelblat, *Virtual reality: applications and explorations*. Academic Press, 2014.

[9] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoper. Virtual Environ.*, vol. 6, no. 4, pp. 355–385, Aug. 1997. [Online]. Available: http://dx.doi.org/10.1162/pres.1997.6.4.355

[10] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–22, June 2013.

[11] M. Flintham, S. Benford, R. Anastasi, T. Hemmings, A. Crabtree, C. Greenhalgh, N. Tandavanitj, M. Adams, and J. Row-Farr, "Where on-line meets on the streets: experiences with mobile mixed reality games," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2003, pp. 569–576.

[12] B. Shi, J. Yang, Z. Huang, and P. Hui, "Offloading guidelines for augmented reality applications on wearable devices," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: ACM, 2015, pp. 1271–1274. [Online]. Available: http://doi.acm.org/10.1145/2733373.2806402

[13] Z. Huang, W. Li, P. Hui, and C. Peylo, "Cloudridar: A cloud-based architecture for mobile augmented reality," in *Proceedings of the 2014 Workshop on Mobile Augmented Reality and Robotic Technology-based Systems*, ser. MARS '14. New York, NY, USA: ACM, 2014, pp. 29–34. [Online]. Available: http://doi.acm.org/10.1145/2609829.2609832

[14] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, "Real-time computer vision with opencv," *Communications of the ACM*, vol. 55, no. 6, pp. 61–69, 2012.

[15] B.-K. Seo, J. Park, H. Park, and J.-I. Park, "Real-time visual tracking of less textured three-dimensional objects on mobile platforms," *Optical Engineering*, vol. 51, no. 12, p. 127202, 2013. [Online]. Available: http://dx.doi.org/10.1117/1.OE.51.12.127202

[16] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[17] D. Chatzopoulos and P. Hui, "Readme: A real-time recommendation system for mobile augmented reality ecosystems," in *Proceedings of the 2016 ACM on Multimedia Conference*, ser. MM '16. New York, NY, USA: ACM, 2016, pp. 312–316. [Online]. Available: http://doi.acm.org/10.1145/2964284.2967233

[18] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, vol. 74, no. 1, pp. 59–73, 2007.

[19] T. Blajić, D. Nogulić, and M. Družijanić, "Latency improvements in 3g long term evolution," in *Proceedings of the International Convention on Information and Communication Technology, Electronics and Microelectronics*, 2006.

[20] M. Abrash. Latency the sine qua non of ar and vr. Accessed 23-02-2017. [Online]. Available: http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/

[21] N. S. Foundation, "Nsf follow-on workshop on ultra-low latency wireless networks," National Science Foundation, Tech. Rep., November 2016, http://inlab.lab.asu.edu/nsf/followup.html. [Online]. Available: http://inlab.lab.asu.edu/nsf/followup.html

[22] J. Dolezal, Z. Becvar, and T. Zeman, "Performance evaluation of computation offloading from mobile device to the edge of mobile network," in *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct 2016, pp. 1–7.

[23] P. Jain, J. Manweiler, and R. Roy Choudhury, "Overlay: Practical mobile augmented reality," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2015, pp. 331–344.

[24] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 68–81. [Online]. Available: http://doi.acm.org/10.1145/2594368.2594383

[25] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, and H. Balakrishnan, "Glimpse: Continuous, real-time object recognition on mobile devices," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: ACM, 2015, pp. 155–168. [Online]. Available: http://doi.acm.org/10.1145/2809695.2809711

[26] OpenSignal. State of mobile networks: Usa (february 2016). Accessed 23-02-2017. [Online]. Available: https://opensignal.com/reports/2016/02/usa/state-of-the-mobile-network/

[27] Y. Xu, Z. Wang, W. K. Leong, and B. Leong, "An end-to-end measurement study of modern cellular data networks," in *International Conference on Passive and Active Network Measurement*. Springer, 2014, pp. 34–45.

[28] R. Research, "Mobile broadband explosion," Rysavy, Tech. Rep., 2011.

[29] Motorola, "Realistic lte performance," Motorola, Tech. Rep., 2009.

[30] SpeedTest. Speedtest market report united states. Accessed 23-02-2017. [Online]. Available: http://www.speedtest.net/reports/united-states/

[31] Y. Heisler. A huge 4g milestone: Lte is now available for 98Accessed 23-02-2017. [Online]. Available: https://bgr.com/2015/03/23/lte-coverage-map-united-states/

[32] 3GPP, "Overview of 3gpp release 12 v0.2.0," 3GPP, Tech. Rep., 2015.

[33] S. Mumtaz, K. M. S. Huq, and J. Rodriguez, "Direct mobile-to-mobile communication: Paradigm for 5g," *IEEE Wireless Communications*, vol. 21, no. 5, pp. 14–23, October 2014.

[34] OpenSignal. 802.11ac: Its still hard to find, but its fast. Accessed 23-02-2017. [Online]. Available: https://opensignal.com/blog/2016/05/05/802-11ac-its-still-hard-to-find-but-its-fast/

[35] ——. Lte latency: How does it compare to other technologies? Accessed 23-02-2017. [Online]. Available: https://opensignal.com/blog/2014/03/10/lte-latency-how-does-it-compare-to-other-technologies/

[36] G. Castignani, A. Lampropulos, A. Blanc, and N. Montavont, "Wi2me: A mobile sensing platform for wireless heterogeneous networks," in *2012 32nd International Conference on Distributed Computing Systems Workshops*, June 2012, pp. 108–113.

[37] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11 b," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2. IEEE, 2003, pp. 836–843.

[38] V. Shrivastava, S. Rayanchu, J. Yoonj, and S. Banerjee, "802.11n under the microscope," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008, pp. 105–110. [Online]. Available: http://doi.acm.org/10.1145/1452520.1452533

[39] W. Alliance, "Wifi peer-to-peer (p2p) technical specification version 1.7," Wifi Alliance, Tech. Rep., 2016.

[40] M. Condoluci, L. Militano, A. Orsino, J. Alonso-Zarate, and G. Araniti, "Lte-direct vs. wifi-direct for machine-type communications over lte-a systems," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on*. IEEE, 2015, pp. 2298–2302.

[41] D. Chatzopoulos, K. Sucipto, S. Kosta, and P. Hui, "Video compression in the neighborhood: An opportunistic approach," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[42] N. Alliance, "5g white paper," *Next generation mobile networks, white paper*, 2015.

[43] R. D. C. S. GSD. Sprintlink commercial availability announced. Accessed 23-02-2017. [Online]. Available: http://h-net.msu.edu/cgi-bin/logbrowse.pl?trx=vx&list=edtech&month=9207&week=&msg=An61j4s0%2BR1UHNuEZOgGfw&user=&pw=

[44] P. R. Center. Broadband vs. dial-up adoption over time. Accessed 23-02-2017. [Online]. Available: http://www.pewinternet.org/chart/broadband-vs-dial-up-adoption-over-time/

[45] Orange. Fiber offers. Accessed 23-02-2017. [Online]. Available: https://boutique.orange.fr/internet/offres-fibre/jet

[46] H. Meller and P. Cohen, "The state of mobile web us 2015," SimilarWeb, Tech. Rep., 2015.

[47] S. Chen and J. Zhao, "The requirements, challenges, and technologies for 5g of terrestrial mobile telecommunication," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 36–43, 2014.

[48] T. van der Vorst, R. Brennenraedts, M. Driesse, and R. Bekkers, "Beyond fast - how the speed of residential internet access will develop between now and 2022," Eindhoven University of Technology, Tech. Rep., 2016.

[49] A. Pesovic. Is symmetrical bandwidth a myth or a must? Accessed 23-02-2017. [Online]. Available: https://insight.nokia.com/symmetrical-bandwidth-myth-or-must

[50] T. Cloonan, M. Emmendorfer, J. Ulm, A. Al-Banna, and S. Chari, "Predictions on the evolution of access networks to the year 2030 and beyond," ARRIS, Tech. Rep., 2015.

[51] M. Heusse, S. A. Merritt, T. X. Brown, and A. Duda, "Two-way tcp connections: Old problem, new insight," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 5–15, Apr. 2011. [Online]. Available: http://doi.acm.org/10.1145/1971162.1971164

[52] Digi-Capital. Augmented/virtual reality revenue forecast revised to hit $120 billion by 2020. Accessed 23-02-2017. [Online]. Available: http://www.digi-capital.com/news/2016/01/augmentedvirtual-reality-revenue-forecast-revised-to-hit-120-billion-by-2020

[53] T. Braud, M. Heusse, and A. Duda, "Dynamics of two antiparallel tcp connections on an asymmetric link," in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[54] A. Sumits. The history and future of internet traffic. Accessed 23-02-2017. [Online]. Available: https://blogs.cisco.com/sp/the-history-and-future-of-internet-traffic

[55] B. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (rsvp) – version 1 functional specification," Internet Requests for Comments, RFC Editor, RFC 2205, September 1997, http://www.rfc-editor.org/rfc/rfc2205.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2205.txt

[56] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC 3550: RTP: A Transport Protocol for Real-Time Applications," IETF, Tech. Rep., 2003. [Online]. Available: www.ietf.org/rfc/rfc3550.txt

[57] ITU, "H222 - Information technology - Generic coding of moving pictures and associated audio information: Systems," ITU, Tech. Rep., 2014.

[58] Z. Fu, X. Meng, and S. Lu, "A transport protocol for supporting multimedia streaming in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1615–1626, Dec 2003.

[59] H. Luo, D. Wu, S. Ci, H. Sharif, and H. Tang, "Tfrc-based rate control for real-time video streaming over wireless multi-hop mesh networks," in *2009 IEEE International Conference on Communications*, June 2009, pp. 1–5.

[60] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "Tcp extensions for multipath operation with multiple addresses," Internet Requests for Comments, RFC Editor, RFC 6824, January 2013, http://www.rfc-editor.org/rfc/rfc6824.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6824.txt

[61] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol," Internet Requests for Comments, RFC Editor, RFC 2960, October 2000.

[62] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath tcp," in *ACM SIGCOMM workshop on Cellular Networks (Cellnet'12)*, 2012.

[63] Jana and I. Swett, "Quic: A udp-based secure and reliable transport for http/2," Working Draft, IETF Secretariat, Internet-Draft draft-tsvwg-quic-protocol-00, June 2015, http://www.ietf.org/internet-drafts/draft-tsvwg-quic-protocol-00.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-tsvwg-quic-protocol-00.txt

[64] E. Kohler, M. Handley, and S. Floyd, "Datagram congestion control protocol (dccp)," Internet Requests for Comments, RFC Editor, RFC 4340, March 2006, http://www.rfc-editor.org/rfc/rfc4340.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4340.txt

[65] K. Kurata, G. Hasegawa, and M. Murata, "Fairness comparisons between tcp reno and tcp vegas for future deployment of tcp vegas," in *Proceedings of INET 2000*, 2000.

[66] T. Flach, N. Dukkipati, Y. Cheng, and B. Raghavan, "Tcp instant recovery: Incorporating forward error correction in tcp," Working Draft, IETF Secretariat, Internet-Draft draft-flach-tcpm-fec-00, July 2013, http://www.ietf.org/internet-drafts/draft-flach-tcpm-fec-00.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-flach-tcpm-fec-00.txt

[67] J. K. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network coding meets tcp: Theory and implementation," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 490–512, March 2011.

[68] A. Hunger and P. A. Klein, "Equalizing latency peaks using a redundant multipath-tcp scheme," in *2016 International Conference on Information Networking (ICOIN)*, Jan 2016, pp. 184–189.

[69] A. Frommgen, T. Erbshuer, A. Buchmann, T. Zimmermann, and K. Wehrle, "Remp tcp: Low latency multipath tcp," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[70] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath tcp," in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, ser. CellNet '12. New York, NY, USA: ACM, 2012, pp. 31–36. [Online]. Available: http://doi.acm.org/10.1145/2342468.2342476

[71] N. Raval, A. Srivastava, A. Razeen, K. Lebeck, A. Machanavajjhala, and L. P. Cox, "What you mark is what apps see," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '16. New York, NY, USA: ACM, 2016, pp. 249–261. [Online]. Available: http://doi.acm.org/10.1145/2906388.2906405

[72] C. Bo, G. Shen, J. Liu, X.-Y. Li, Y. Zhang, and F. Zhao, "Privacy.tag: Privacy concern expressed and respected," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '14. New York, NY, USA: ACM, 2014, pp. 163–176. [Online]. Available: http://doi.acm.org/10.1145/2668332.2668339

[73] P. Aditya, R. Sen, P. Druschel, S. Joon Oh, R. Benenson, M. Fritz, B. Schiele, B. Bhattacharjee, and T. T. Wu, "I-pic: A platform for privacy-compliant image capture," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '16. New York, NY, USA: ACM, 2016, pp. 235–248. [Online]. Available: http://doi.acm.org/10.1145/2906388.2906412

[74] T. Hoeiland-Joergensen, P. McKenney, dave.taht@gmail.com, J. Gettys, and E. Dumazet, "The flowqueue-codel packet scheduler and active queue management algorithm," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-aqm-fq-codel-06, March 2016, http://www.ietf.org/internet-drafts/draft-ietf-aqm-fq-codel-06.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-aqm-fq-codel-06.txt

[75] *Linux Magazine*, vol. 175, 2015. [Online]. Available: http://www.linux-magazine.com/Issues/2015/175/FQ-CoDel-and-MPTCP