Mobile Edge Computing: A Taxonomy

Michael Till Beck, Martin Werner, Sebastian Feld
Ludwig Maximilian University of Munich
{michael.beck,martin.werner,sebastian.feld}@ifi.lmu.de

Thomas Schimper
Nokia Networks
thomas.schimper@nsn.com

***Abstract*—Mobile Edge Computing proposes co-locating computing and storage resources at base stations of cellular networks. It is seen as a promising technique to alleviate utilization of the mobile core and to reduce latency for mobile end users. Due to the fact that Mobile Edge Computing is a novel approach not yet deployed in real-life networks, recent work discusses merely general and non-technical ideas and concepts. This paper introduces a taxonomy for Mobile Edge Computing applications and analyzes chances and limitations from a technical point of view. Application types which profit from edge deployment are identified and discussed. Furthermore, these applications are systematically classified based on technical metrics.**

***Index Terms*—edge deployment, cellular networks, classification**

I. Introduction

Increasing utilization of network resources is one of the most apparent challenges for mobile network operators. Data traffic contributes heavily to today's overall mobile network traffic. Many mobile applications rely on data and services hosted in remote data centers. This induces high network load, since data have to be up- and downloaded to and from mobile devices and data centers connected to the Internet.

Moreover, new mobile applications accessing Internet services are expected to further contribute to this trend: In fact, bandwidth demands are expected to continue doubling each year [1]. And this trend does not yet incorporate for effects due to wearable devices and the Internet of Things, which add new devices such as Google Glass to the mobile ecosystem. With the increased computational power of these devices, novel application scenarios become realistic including augmented reality leading to an even higher bandwidth demand.

To keep up with these increasing demands, network operators are obliged to enhance and upgrade capacities of existing network resources continuously. Furthermore, they are impelled to integrate novel technologies into their infrastructure in order to provide sufficient quality of experience for mobile end users. New technologies like LTE Advanced introduce higher bandwidth capacities and lower latency. Higher edge capacities, however, also directly affect utilization within the network core and entail further investments. Both, enhancing existing resources and integrating new technologies, comes with significant operational cost.

Mobile Edge Computing (MEC) has recently been proposed as a promising technology to overcome this dilemma in certain scenarios. MEC aims at reducing network stress by shifting computational efforts from the Internet to the mobile edge. Traditionally, devices deployed at the mobile edge solely act as mobile access points: Base stations forward traffic, but do neither actively analyze nor respond to user requests. Thus, they do not provide computing resources for hosting edge services beyond network connectivity. MEC introduces new network elements at the edge, providing computing and storage capabilities at the edge. Therefore, new devices are deployed and co-hosted at base station towers. In the following, these devices are referred to as MEC servers.

Fig. 1 depicts the MEC ecosystem and the integration of MEC servers into the mobile network topology. There are four stakeholders involved in this scenario: 1) Mobile end users using User Equipment (UE), 2) network operators owning, managing, and operating base stations, MEC servers, and the mobile core network, 3) Internet infrastructure providers (InPs) maintaining Internet routers, and 4) application servive providers (ASPs) hosting applications within data centers and content delivery networks (CDN). Mobile devices (UE) connect to the eNodeBs which translate Radio signals so they can be routed through the wired access and core networks. MEC Servers are deployed in close proximity of the eNodeB, typically by physically attaching it there and looping the traffic through the MEC server for further processing the data. The MEC server is capable of participating both in user traffic and control traffic (S1-U and S1-C interfaces). MInP and ASPs deploy rulesets, filters, and MEC services at the MEC servers, defining how to handle specific traffic. In this way, MEC services are capable of managing specific user requests directly at the network edge, instead of forwarding all traffic to remote Internet services. MEC servers either process a request and respond directly to the UE or the request is forwarded to remote data centers and content distribution networks (CDNs).

Being directly handled by services hosted on MEC servers, these requests do not need to be forwarded through the core infrastructure. Traditionally, all data traffic is routed through the core network to a base station which delivers the content to mobile devices. In the MEC scenario, MEC servers take over

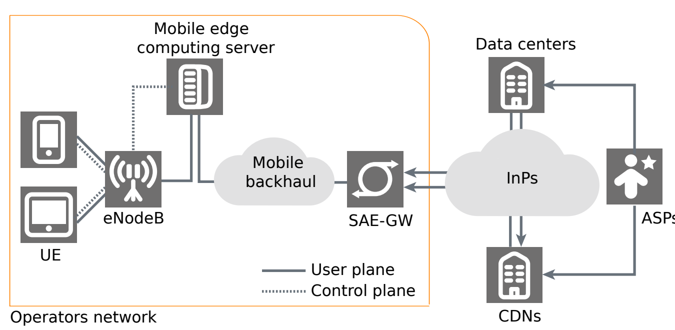

Fig. 1: Mobile Edge Computing Topology

 ISBN: 978-1-61208-377-3

some or even all of the tasks originally performed by Internet services. Being co-located next to base stations, computing and storage resources of MEC servers are also available in close proximity to mobile users, eliminating the need of routing these data through the core network. Therefore, MEC is seen as a future, promising approach to increase quality of experience in cellular networks [2]–[4]. Furthermore, it enables the deployment of novel application types at the mobile edge.

This paper provides an analysis of technical chances and limitations of MEC by identifying, discussing, and classifying various applications and application types for the deployment at the mobile edge.

II. Related Work

Until today, MEC servers have not been deployed in cellular networks; thus, the MEC concept has been discussed only from a theoretical perspective so far. However, there are some related approaches that are similar to this concept. For example, mobile cloud computing is highly related to mobile edge computing. A survey on mobile cloud computing is given by Dinh et al. [5], describing cloud-affine mobile application types. As an example for mobile cloud computing, the cloudlet concept is discussed by Satyanarayanan et al. [4]: Cloudlets are trusted, resource-rich, mostly stationary computers with fast and stable Internet access, offering computing, bandwidth, and storage resources to nearby mobile users. While MEC servers are operated by mobile infrastructure provider, cloudlets are owned and managed by mobile end users. Mobile users access cloudlet via a local area network such as Wi-Fi in order to instantiate services. Not being connected to the mobile network, cloudlets do not share network operator related knowledge. Thus, cloudlets are suitable for offloading resource-intense tasks from the mobile end user device in order to increase execution speed or battery lifetime.

Fesehaye et al. [6] focus especially on interactivity when stating that cloudlets are also capable of caching and transferring content. A content-centric local networking approach is introduced, using interconnected cloudlets. Their contribution is threefold: First, a mobile infrastructure as a service cloud is defined, using both cloud technology and cloudlets. Second, in order to realize content-centric features, a wireless routing protocol is proposed in order to enable communication between two cloudlets as well as between two mobile users via a cloudlet. Third, the impact of cloudlets on interactive mobile cloud applications like file editing, video streaming, and messaging is analyzed. However, offloading and content caching are just two of the use cases for MEC. In contrast to cloudlets, MEC servers are widely deployed and available to all mobile users, not just to some specific ones. Being co-located with base stations, MEC servers provide additional features such as being able to access position and mobility information. This paper discusses these features, listing and classifying application types for the deployment at the mobile edge.

Until now, the MEC concept itself has mainly been discussed from a non-technical perspective. E.g., IBM discusses economical benefits for businesses and M2M applications [3]. A first real-world MEC platform was introduced and motivated by Nokia Networks [2] in 2014. In this concept, MEC servers are standard IT equipment with processing and storage capacity directly placed at mobile network's base stations. Being placed at the mobile edge, MEC servers are capable of collecting real-time network data like cell congestion, subscriber locations, and movement directions. Furthermore, some individual application types (but neither analyzed nor classified from a technical perspective) are motivated for running at the mobile edge. This concerete mobile edge computing platform will be described in the following section.

III. A first Mobile Edge Computing Platform

As discussed in the previous section, some initial ideas on Mobile Edge Computing have been discussed in literature before. However, these discussions are limited to a more or less theoretical perspective, since no real implementation of MEC servers was introduced so far. Just in 2014, Nokia Networks has introduced a very first real-world MEC platform [2]: Radio Applications Cloud Servers (RACS) represent concrete incarnations of MEC servers.

This section shortly discusses NSN's approach as an example for a realistic MEC deployment. The section is structured as follows: First, hardware configuration is described; then, the software architecture is explained; and third, traffic forwarding and filtering rules are depicted.

In line with Figure 1, NSN's MEC servers are deployed next to base stations: they are co-hosted with base stations and are directly linked to them. MEC servers are equiped with commodity hardware, i.e. usual server CPUs, memory, and communication interfaces. Application deployment is based on cloud technology and virtualization. Therefore, RACS provide a VM hypervisor (see Fig. 2) for the deployment of VM images running MEC applications.

VMs have to fulfil certain requirements, like providing a self-monitoring service that keeps sending heartbeat messages to the RACS system. The hypervisor will reboot the VM if heartbeat messages are not sent by the VM, ensuring that the VM is automatically reinitialized after some applications crashed. Furthermore, for security considerations, VMs have to be signed before deployment. This enables the operator to verify that the VM state has not been altered by malicious offenders. VMs are able to communicate with the RACS platform via a message bus, as most applications running on the mobile edge are expected to be event driven. Via the message bus, VMs subscribe to message streams, i.e., topics. This way, VMs are able to retrieve UE data streams and cell-related notifications. As an example, some subscription topics refer to specific traffic classes sent or received by mobile devices. VMs can subscribe to all traffic with a specific destination address or port number.

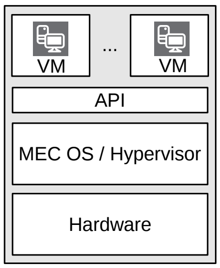

Fig. 2: RACS Architecture

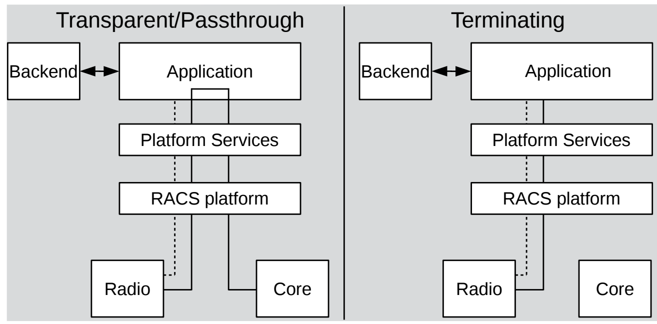

Fig. 3: Application Types

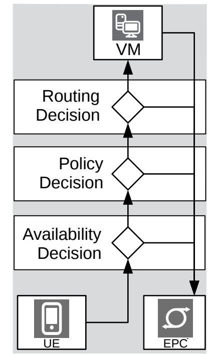

Fig. 4: Forwarding and Filtering Rules

Two main categories of applications can be deployed at RACS servers, depending on the traffic flow: transparent/pass-through and terminating applications (cf. Fig. 3). The dotted line in Fig. 3 represents the control plane, which is available in both applications types. The line having arrowheads represents out-of-band communication that both application types are optionally able to perform. The solid line is the user plane.

Transparent/pass-through applications are capable of monitoring, rerouting, and augmenting UE traffic. In this situation, additional header information can be introduced to HTTP requests including network-specific information, which is not available to ASP services in traditional cellular networks.

For terminating applications, UE traffic is encapsulated into IP packages with a virtual IP address. This packet flow is then routed into a VM, where it terminates (note the truncated solid line in Fig. 3). If the VM is not running or no MEC server is co-hosted with the current base station, the encapsulated packages are routed to a server in the mobile network core handling the requests. Packages are rerouted transparently, which means that developers of mobile applications can refer to the same URL in order to access the service which is provided by the MEC server's infrastructure were applicable, or by a backend service, where needed.

Mobile network operators define forwarding and filtering rulesets for traffic routed through MEC servers. Based on both privacy considerations and application providers' demands, these rulesets specify which data are sent to which type of application. In accordance with the subscriptions to the topics mentioned above, mobile traffic is routed through the VMs. Fig. 4 provides an overview of this static decision tree: UE traffic is sent to the base station and its co-located MEC server. For each rule, it is validated whether the application corresponding to this rule is up and running, i.e., whether the VM hosting the application is active. If this is the case, the filtering ruleset is applied to identify information that is permitted to be accessed by the application. The last step is the actual routing decision. After a positive result, information is visible to the application.

In the following, application types and use cases will be discussed which are promising candidates for being hosted by MEC/RACS servers.

IV. Applications and Use Cases

Introducing a Mobile Edge Computing platform into a cellular network allows for applications to be executed directly at the serving base station. While the concept of Mobile Edge Computing has been introduced in literature before, it still remains an open question, which applications profit from being deployed at the edge. This section categorizes and discusses several application types which are promising candidates for the deployment at the mobile edg: Subsection A introduces a classification scheme for MEC applications; subsection B discusses several applications and subsection C highlights the main benefits of Mobile Edge Computing.

A. Classification

This section introduces a classification for several approaches. It is based on three levels of abstraction (cf. Figure 5). As a first distinction, the application classes "Offloading", "Edge Content Delivery", "Aggregation", "Local Connectivity", "Content Scaling" as well as "Augmentation" were identified. While there might be additional classes of applications which could benefit from edge computing, we believe that these application classes will have the strongest impact. In order to further organize application examples and their demands, subgroups of applications showing a similar footprint with respect to resource demands were introduced. Then, concrete examples of applications were given to show the variability of applications inside classes exploiting mobile edge computing resources in a similar way. Another perspective on the classification of applications is given by starting with advantages of edge computing: The most obvious advantage of edge computing inside cellular networks is given by a reduction of end-to-end delay. When packets do not have to travel through the evolved packet core to the application server on the Internet, an application can provide real-time services with strong, constant and known bounds on the delay. A reduced delay motivates the deployment of applications from all given classes: In every case, a solution without edge computing would involve a transmission through the core network as well as through Internet links towards

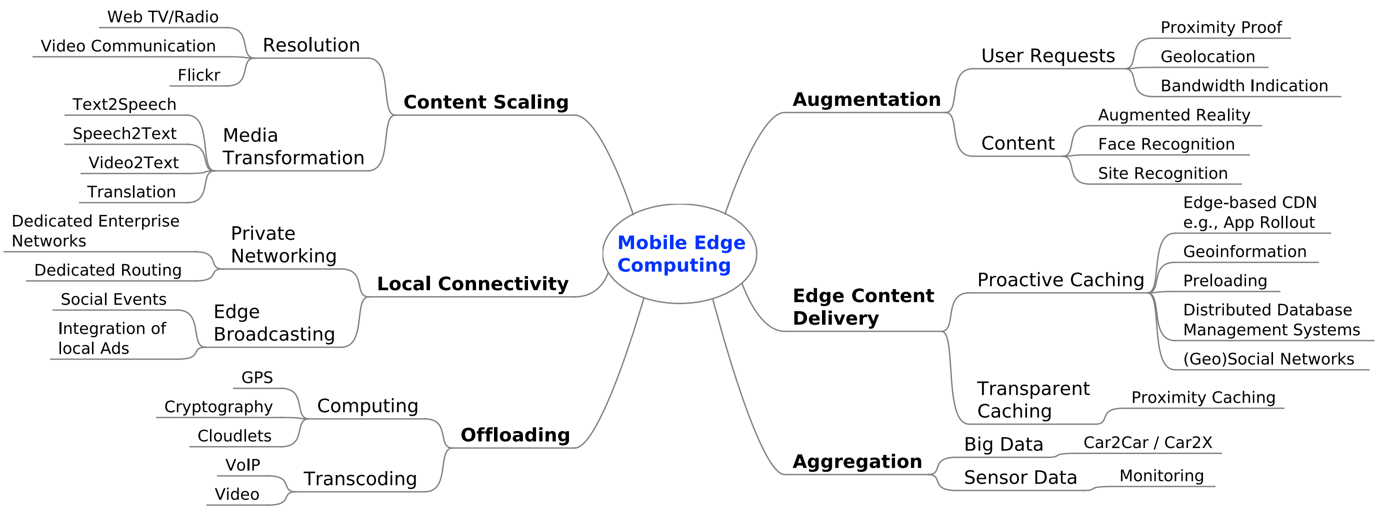

Fig. 5: Mobile Edge Computing: Applications and Use Cases

the application host and back. Additionally, the use of edge computing makes offloading feasible in more cases, as today's radio bandwidth is much higher compared to usable Internet bandwidth and tasks that would typically be performed on the mobile device due to the size of their input can be performed on the edge. Finally, a third motivation for edge computing is given by the local nature of the edge computing servers: By storing only relevant information for the coverage of a single cell, many computational tasks can be performed on small datasets reducing overhead and possibly increasing privacy: A location-based information service, for example, can provide its service inside each cell and in the case of edge-terminating traffic, the location of mobile users keeps inside the mobile network, where it was previously known.

In the following, advantages and disadvantages of each application class are discussed using the examples given in Figure 5. To this end, several key metrics are discussed and a rating for each of these metrics is provided with respect to the three main entities: Mobile user (UE), cellular network provider (MInP), and application service provider (ASP). For the purpose of this evaluation, the following metrics are considered in order to evaluate the feasibility of edge computing for a given class of applications:

Power Consumption: The effect on power consumption of each power-consuming device including the mobile device and also the base stations. Furthermore, power consumption is relevant to network operators, though with a lower impact.

Delay: The effect on delay introduced due to modified communication, computation, and system's complexity.

Bandwidth utilization: The effect on bandwidth demand and cost for each entity.

Scalability: The effect on scalability of algorithms exploiting the available location information at the edge.

B. Applications

Figure 6 gives a qualitative assessment of the impact of the identified application classes on the three main stakeholders. The following sections explain examples for each application class and motivate the decisions made for the assessment of the table.

Offloading: Even today, many mobile applications delegate resource- or power-intense tasks to remote services due to limited hardware capabilities. MEC servers offer additional capacities for hosting such services at the mobile edge. The concept is expected to increase limited computing, storage, bandwidth, or battery capacities of mobile devices by referring to external, resource-rich systems. Compute-intense tasks are offloaded either because they can not be executed in-time by the UEs due to limited hardware capabilities or they are offloaded in order to reduce power consumption of mobile devices in cases where the power consumption needed for computation exceeds the power consumption needed for wireless transmission. If no MEC server is available, mobile devices can degrade gracefully to a more distant MEC server, Internet cloud servers, or fallback to their own hardware resources [4], [6]. For example, calculating GPS positions is a power-intense task which can gainfully be offloaded to remote servers. Also, asymmetric encryption requires much more battery power than symmetric approaches. Therefore, asymmetric encryption is offloaded, and more battery-friendly encryption methods are chosen in order to encrypt communication between UEs and the base station. Offloading of power-intense tasks like transcoding of multimedia traffic also falls into this category. VoIP applications transcode traffic depending on the current load of the base stations, enabling real-time bitrate adaptations and better QoE.

With respect to our metric system, offloading is motivated by reduced delay due to the fact that traffic between MEC servers and mobile devices has not to be routed through the core network. Another objective of offloading is the reduction of power consumption of the mobile device. Of course, the sending of the task request and response does not have to consume more power than the local execution. Core and ASP

 ISBN: 978-1-61208-377-3

Class	Entity	Metric			
		Power Consumption	Delay	Bandwidth Usage	Scalability
Offloading	UE	++	++	o	o
	MInP	+	+	++	o
	ASP	o	++	+	+
Edge Content Delivery	UE	o	++	o	o
	MInP	o	++	++	+
	ASP	+	++	+	++
Aggregation	UE	o	-	o	o
	MInP	+	+	++	+
	ASP	++	-	++	+
Local Connectivity	UE	o	++	o	o
	MInP	+	+	++	++
	ASP	o	++	++	++
Content Scaling	UE	o	++	o	o
	MInP	-	+	+	++
	ASP	+	++	++	++
Augmentation	UE	o	+	o	o
	MInP	o	o	o	o
	ASP	o	+	o	o

Fig. 6: Classification of Mobile Edge Computing Applications

can benefit indirectly from offloading, namely when more calculations are performed at the edge as compared to the situation without offloading. Offloading introduces cost due to higher system complexity: Even simple systems become complex distributed systems and have to deal with communication, marshalling, availability, and errors.

Edge Content Delivery: MEC servers offer resources for the deployment of additional content delivery services at the network edge. Content traditionally hosted by Internet services/CDNs is now shifted more to the network edge. MEC servers operate as local content delivery nodes and serve cached content. Caching techniques, not only in the context of Mobile Edge Computing, can be classified as being either reactive/transparent or proactive. *Transparent Caching*: Caching is transparent if neither the UE nor the ASP are aware of the caching MEC server. As shown by Ericcson, 10% of mobile data traffic is expected to be generated by web browsing, and more than 50% by video data [7] in 2019. Therefore, caching content at the edge is a promising approach to reduce communication quantity and latency for core network providers. *Proactive Caching*: Content is non-transparently cached before it was requested, since it is expected to generate high network utilization in the future. One example here is the roll-out of software updates before they are actually requested by mobile devices. Another example is caching proximity-related data: Geo-Social Networks (GSNs) like Google Latitude and Yelp store region-related content. Mobile users often use these services to request information about geographically nearby locations and places (restaurants, etc.).

Proactive caching is highly related to content distribution networks and is expected to lead to further improvements in terms of bandwidth reduction for the core net and the ASP, and in terms of shorter transmission delays for the mobile devices. ASPs play an important role in this scenario, since they provide relevant information on which content should be distributed throughout the network. Another example is the pre-loading of user content. In order to reduce transmission delays at the UE site, ASPs can preload content that is expected to be requested by the UE user. In contrast to proactive caching, decisions whether (and which) additional content should be send to MEC servers depend on actions performed by each specific UE user. Pre-loading is well known and actively used by companies like Amazon, for example: Amazon silently pre-loads content on the client side that might possibly be requested by the user in the near future while the user is browsing the Amazon website [8]. This leads to decreasing transmission delay and an improved user experience. In the context of Mobile Edge Computing, pre-loading is shifted from UEs to MEC servers in order to decrease power consumption caused by the transmission of data to the ME.

Both approaches can be used either isolated or shared. In the isolated scenario, each cache works independently of other caches: Content already cached by other MEC servers is not shared. In the shared scenario, MEC servers cooperate and obtain content from other MEC servers.

Technically, edge content delivery reduces network utilization and network delay. Similar to distributed database management systems (DDBMS), edge content delivery aims at storing data in close proximity to where they are usually requested. This kind of data localization leads to a reduction of computational complexity, compared to centralized database systems. But it also decreases access delays with respect to latency, since communication paths are kept short [9]. Also, overall bandwidth usage decreases, since less network resources are needed to transfer data: On the one hand, MEC servers have to synchronize with each other to ensure that data are stored consistently, which comes with additional communication overhead. But, on the other hand, UEs frequently requesting data can fetch them directly from a DDBMS instance nearby instead of having to establish remote connections to a centralized service. Therefore, this leads to an overall reduction of communication overhead in the core network and also in the ASP network. Of course, the applicability of edge content delivery depends on the locality of the data. Utilization of the core network resources decreases if UEs frequently request data stored by the local DDBMS instances.

Aggregation: Instead of routing all UE data to core routers separately, MEC servers are capable of aggregating similar or related traffic and, thus, reduce network traffic. As an example, many Big Data applications like Car2Car solutions generate a lot of similar and region-related event notifications which can be aggregated. This also applies in the context of monitoring applications where many devices measure similar data that can be aggregated at the edge.

Due to the fact that the quantity of data received by ASPs would decreases, aggregation has a positive effect in terms of ASPs' bandwidth utilization, power consumption, and scalability. However, delay increases since data need to be processed by MEC servers. Since core network traffic decreases, the same applies for bandwidth utilization, power consumption, and scalability of the MInP. Operating MEC servers comes with additional power consumption cost, however, total power consumption is expected to decrease as a result of lower core utilization.

Local Connectivity: With traffic being routed through MEC servers, servers are capable of separating traffic flows and redirecting traffic to other destinations. An application of this class is connecting enterprise users directly via base stations deployed on enterprises' rooftops to the enterprise network. As an example, this applies on sports/music events where cameras catching additional viewpoints broadcast their content among users in the cell. Furthermore, Local Breakin allows for local redistribution of data fed into the cell, for example, advertisements and information related to the geographical location of the base station. Thus, MEC servers broadcast locally generated and locally relevant content within the cell.

Traffic is routed by circumventing Internet routers, leading to lower communication delay for UEs and ASPs. Furthermore, MInP's bandwidth utilization and power consumption is reduced, since traffic is not routed through the core network. Reduced network utilization has a positive impact with respect to MInP's communication delay.

Content Scaling: MEC enables downscaling of user-generated traffic before it is routed through the mobile core network. Content scaling can also be applied to traffic sent by Internet servers. Scaling UE-generated content before it is delivered to ASPs' data centers decreases bandwidth demands of ASPs. As an example, image sharing sites like flickr and facebook downscale user generated content in order to reduce storage demands. Downscaling UE content directly at the edge also reduces MInP's core network utilization. Additionally, MEC also enables real-time scaling of Internet content – if traffic congestion occurs at base station site, MEC servers are able to downscale traffic in order to both reduce stress of MInPs' base stations and increase network speed.

Augmentation: Since additional information is available at the base station site, these data can be shared with ASPs in order to enhance quality of experience. To this end, mobile network operators enhance requests sent by the UEs by also including statistics on the number of connected UEs, bandwidth utilization, and so on. As an example, current and expected cell congestion are two factors enabling real-time adaption of ASP's service parameters like content resolution as well as communication and notification behavior.

MEC enables mobile network operators to also provide user-related information, since these data are available in the cellular network and get lost as soon as packets are processed by Internet routers. Thus, in order to provide enhanced services tailored to the needs of the UE user, mobile network operators can inject additional data (e.g., age, sex, postal address, cell movement patterns, etc.) into the original requests. Obviously, privacy aspects have to be taken into account when applying these feasibilities in the real world. In addition to this non-technical enhancement, MEC-based augmentation comes with reduced network delay due to the fact that ASPs are able to adapt service parameters in real-time, rather than reactively: MEC enables ASPs to tailor content in real-time to the needs of the UEs.

C. Advantages of Mobile Edge Computing

The following considerations can be concluded from the previous subsections: From a technical perspective, end users benefit mostly from reduced communication delay. Here, one interesting application class is offloading: Due to its close proximity to the end user, MEC servers enable new kind of applications to be considered as offloading candidates. From the MInPs point of view, the most interesting aspect of MEC is bandwidth reduction and scalability. Here, interesting applications are edge content delivery, aggregation, and local connectivity. ASPs profit with respect to scalability and faster services. MEC enables them to host services at the edge, which results in lower bandwidth demands within data centers. Furthermore, augmentation enables novel possibilities for ASPs, since cellular network specific information can be integrated into the traffic flow that are, due to technical limitations, not available in conventional networks.

V. Conclusion and Future Work

This paper discussed several applications for the deployment at the mobile edge and classified them based on six categories. These categories were evaluated based on the technical parameters power consumption, delay, bandwidth usage, and scalability. Benefits for stakeholders, namely mobile end user, network operator, and ASPs were analyzed. As discussed before, in most deployment scenarios, mobile end users and MInPs profit from reduced network delay, and, thus, faster services. Furthermore, from the ASPs' point of view, MEC enables the integration of additional, congestion- or user-related information into the traffic flow.

ISBN: 978-1-61208-377-3

Several questions remain open for future work: Whilst being quite promising in this context, offloading has not been analyzed so far with respect to MEC. In contrast to offloading approaches that apply in cloud networks, several constraints have to be taken into account in the MEC scenario: Mobile applications have to be aware of the fact that MEC servers are deployed in a decentralized way and, since the mobile user might move from its current geographical position, connectivity between MEC servers and end user device is constrained. Thus, applications that rely on MEC services have to be mobility-aware and need to fallback gracefully to other MEC servers, distant cloud servers, or even the UE itself. Beyond, efficient and power-saving offloading approaches for VoIP systems have not been discussed in this context. We already initiated some measurements and experiments in that direction, which look quite promising. Offloading decision factors need to be evaluated, deciding when to offload data to MEC servers (e.g., depending on link quality, interferences, and congestion). With respect to Edge Content Delivery, proximity-aware caching algorithms are needed, deciding when and how MEC servers request remote data for storing at the edge, avoiding congestion and enhancing Quality of Experience.

ACKNOWLEDGMENT

We would like to thank Uwe Puetzschler (Nokia Networks) for kindly supporting our work.

REFERENCES

[1] Ericcson, “Ericcson Mobility Report – June 2013.”
[2] Intel and Nokia Siemens Networks, “Increasing mobile operators’ value proposition with edge computing.”
[3] IBM Corporation, “Smarter wireless networks; add intelligence to the mobile network edge.”
[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.
[5] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: architecture, applications, and approaches: A survey of mobile cloud computing,” *Wireless Communications and Mobile Computing*, vol. 13, pp. 1587–1611, Dec. 2013.
[6] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, “Impact of cloudlets on interactive mobile cloud applications,” *16th International Enterprise Distributed Object Computing Conference*, pp. 123–132, Sept. 2012.
[7] “Ericsson Mobility Report – November 2013.”
[8] P. Jones, C. Newcombe, R. Ellis, D. Birum, and M. Thompson, “Method and system for preloading resources,” Feb. 22 2011. US Patent 7,895,261.
[9] M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems, Third Edition*. Springer, 2011.