COMMUNICATIONS, CACHING, AND COMPUTING FOR CONTENT-CENTRIC MOBILE NETWORKS

Big Data Caching for Networking: Moving from Cloud to Edge

Engin Zeydan, Ejder Baştüğ, Mehdi Bennis, Manhal Abdel Kader, Ilyas Alper Karatepe, Ahmet Salih Er, and Mérouane Debbah

ABSTRACT

In order to cope with the relentless data tsunami in 5G wireless networks, current approaches such as acquiring new spectrum, deploying more BSs, and increasing nodes in mobile packet core networks are becoming ineffective in terms of scalability, cost, and flexibility. In this regard, context-aware 5G networks with edge/cloud computing and exploitation of big data analytics can yield significant gains for mobile operators. In this article, proactive content caching in 5G wireless networks is investigated in which a big-data-enabled architecture is proposed. In this practical architecture, a vast amount of data is harnessed for content popularity estimation, and strategic contents are cached at BSs to achieve higher user satisfaction and backhaul offloading.

INTRODUCTION

Nowadays, wireless data traffic is experiencing tremendous growth due to pervasive mobile devices, ubiquitous social networking, and resource-intensive applications of end users with anywhere-anytime-to-everything connectivity. This unprecedented increase in data traffic, chiefly driven by mobile video, online social media, and over-the-top (OTT) applications, is compelling mobile operators to look for innovative ways to manage their increasingly complex networks and scarce backhaul resources. In fact, a major driver of this backhaul problem is wireless video on demand traffic in which users access contents whenever they wish in an asynchronous fashion (unlike live streaming and digital TV), and has unique characteristics (i.e., users’ demands concentrate on a small set of popular contents, resulting in heavy-tail distribution) [2]. The explosion of data traffic stemming from diverse domains (healthcare, machine-to-machine communication, connected cars, etc.) with different characteristics (i.e., structured/non-structured) falls into the framework of big data [3]. Indeed, the potential offered by big data has spurred great interest from industry, government, and academia [4, references therein].

At the same time, mobile cellular networks are moving toward fifth generation (5G) wireless networks, in which ultra-dense networks, massive multiple-input multiple-output (MIMO), millimeter-wave communication, edge caching, and device-to-device communications are heavily investigated [5, references therein]. In contrast to the base-station centric architectures (possibly) designed for dumb mobile terminals where requests are satisfied in a reactive way, 5G networks will be user-centric, context-aware, and proactive in nature.

Driven by the surge of social and mobile applications, today’s mobile network architectures ought to contemplate a new paradigm shift. Indeed, the era of collecting and storing information in data centers for data analysis and decision making has dawned. Telcos are looking for decentralized and flexible network architectures where predictive resource management plays a crucial role, thanks to the recent advances in storage/memory, context awareness, and edge/cloud computing [6–9]. In the wireless world, big data brings about a new kind of information set to network planning which can be interconnected to get a better understanding of users’ behavior and network characteristics (location, user velocity, social geo-data, etc.).

In light of the above, this work investigates the exploitation of big data in mobile cellular networks from a proactive caching point of view. Because human behavior is highly predictable and a large amount of data is streaming through operators’ networks, this article proposes a proactive caching architecture. This architecture optimizes 5G wireless networks where a large amount of available data is exploited by harnessing big data analytics and machine learning tools for content popularity estimation. We also show how this new architecture can be exploited for caching at the edge (particularly at base stations [BSs]),...
yielding higher user satisfaction and backhaul offloading gains by moving contents closer to users. Nowadays, it is common to have terabytes of data per second flowing in a typical mobile operator consisting of 10–20 million subscribers, which translates into roughly exabytes monthly. As a real-world example, we process a large amount of data collected on a big data platform from one of the major mobile networks in Turkey with 17 million subscribers. These traces are collected from several BSs in hundreds of time intervals and analyzed inside the network to ensure privacy concerns and regulations. To the best of our knowledge, this is perhaps the first attempt to showcase the potential of big data for caching in 5G mobile networks.

**Prior Work and Our Contribution**

Not surprisingly, the exploitation of big data in mobile computing has been investigated recently in many works (e.g., [10]). Caching at the edge of mobile wireless networks (i.e., BSs and user equipment) is also of high interest as evidenced in [6, 11, 12]. Briefly, technical misconceptions of caching for 5G networks are introduced in [6]. A study on improving the video transmission in cache-enabled wireless networks via opportunistic reuse of cache-enabled devices is given in [11]. Cooperative caching for delivering layered videos to mobile users is studied in [12].

Compared to existing works, our main contribution is to highlight and assess the potential gains of big data processing techniques for cache-enabled wireless networks by using real traces of mobile users collected from BSs in a large urban area. To the best of our knowledge, none of the previous approaches has focused on deployment of a Hadoop-based big data processing platform inside a mobile operator’s (MO’s) core network in order to validate the performance gains of caching with real data trials. By using tools from machine learning to predict content popularity, further improvements in user’s quality of experience (QoE) and backhaul offloading are achieved via proactive caching at the edge.

The rest of the article is structured as follows. The role of big data and proactive edge caching in wireless networks is briefly discussed. An architecture based on a big data platform and cache-enabled BSs is proposed. A practical case study is carried out, where the traces collected from an MO’s network are processed on the big data platform and gains of proactive caching are validated via numerical simulations. We conclude the article and provide future directions.

**Big Data Analytics for 5G Networks: Requirements, Challenges, and Benefits**

Today’s networking requirements are being software-defined in order to be more scalable and flexible against big data. Tomorrow’s big networks will be even more complex and interconnected. For that matter, MOs’ data centers and network infrastructures will need to monitor traffic patterns of tens of millions of clients using possible collection units of user statistics data (e.g., location, traffic demand pattern, capability) for proper analysis.

**Current Challenges and Trends in Big Data Networking**

Recently, data traffic patterns inside MOs’ data centers have changed dramatically. Big data has enabled high traffic exchange between gateway elements at backhaul. Although wireless technology has improved tremendously from 2G to 4G, backhaul connections of cellular networks have not seen such a rapid evolution. Hence, the mobile backhaul intra-traffic is slowly becoming larger than the inter-traffic between mobile backhaul elements and end users. Indeed, in today’s carrier networks, in addition to handling mobile users’ traffic via mobile backhaul, fetching data from a number of different backend, database, and cache servers, as well as the data generated by gateway and backhaul elements also contribute to this traffic load within the operator’s infrastructure. In fact, interactions of user terminals (UTs) trigger various interactions with hundreds of servers, routers, and switches inside the backhaul and core network. For example, for an original user’s HTTP request of 1 KB, the intranet data traffic can increase up to 930× [13]. This is contrary to the traditional carrier network architecture, which assumes client and wireless access nodes as bottlenecks lacking computational overhead rather than the backhaul infrastructure. Moreover, since data growth is a major challenge in today’s mobile infrastructures, managing this big-data-driven networks in cloud environments is a pressing issue. For this reason, mobile edge computing (sometimes nicknamed fog computing) is yet another emerging technology where edge devices provide cloud-computing-like capabilities within the radio access network to carry out functionalities such as communication, storage, and control [9]. However, for 5G networks, it should be noted that deploying distributed cloud computing capabilities near to each BS site (especially at locations where traffic volume is relatively low) may also increase the deployment cost considerably compared to centralized computing solutions due to availability of hundreds of sites in a typical MO. Moreover, for modeling and prediction of spatio-temporal users’ behavior in user-centric 5G networks, network traffic arriving to a centralized location needs to be scaled out horizontally across servers and racks, which is only feasible inside the core site of an MO for proper analysis rather than distributed locations with relatively low traffic.

**When Big Data Analytics Meets Caching: A Hadoop Case Study**

Due to the recent developments in networking technology and standards as well as new forms of personal communications, big data has gained increased popularity, especially inside data centers and MOs. With the enormous challenges of big data in the networking world, it is evident that the only way to cope with the growing network data traffic is through better data management and movement of data from the cloud into the edge. In recent years, Hadoop has been successful as a big data management software solution offering dramatic cost savings over traditional tier one database infrastructures, processing capabilities of various data formats, and...
The proposed architecture parallelizes the computation and execution of the content prediction algorithms at core site and cache placement at BSs. By doing so, users’ demands are highly satisfied yielding low latency and higher QoE.

![Diagram of cache-enabled architecture](image)

**Figure 1. Illustration of the proposed architecture.** The contents are moved from the cloud to the edge (BSs) by first inferring strategical contents on a big data platform inside an MO’s core site, then proactively storing them at cache-enabled BSs.

parallel processing over multiple nodes. Additionally, advanced analytic techniques in machine learning in conjunction with non-relational databases that can exploit big data (e.g., NoSQL databases) have increased the opportunity of understanding big data.

It is clear that moving contents’ proximity to the edge is important whenever a user’s connectivity times out while performing streaming and/or downloading activities. To mitigate this, allowing data to be closer to users by reducing the distance of content to users and pushing the right content and applications at the edge yield better user experience. For instance, allowing Hadoop’s distributed data processing engine for analyzing users’ behavior from enormous amounts of streaming data (through the core site of MOs) as well as exploiting proactively caching strategic contents at edges (e.g., at BSs) can ease the backhaul traffic and improve users’ quality of experience (QoE) by latency reduction. The following section discusses a Hadoop-based big data processing platform and its relation with edge caching as one way of dealing with big data inside MOs.

**BIG-DATA-AIDED CACHE-ENABLED ARCHITECTURE**

The goal of this section is to investigate a new practical system architecture to gather, analyze, and proactively tackle the skyrocketing data surge. Motivated by highly predictable human behavior, the proposed architecture collects contextual information (e.g., users’ viewing history and location information) and predicts users’ spatio-temporal demand to proactively and judiciously cache selected contents at the network edge. The proposed architecture parallelizes the computation and execution of the content prediction algorithms at the core site and cache placement at BSs. By doing so, users’ demands are highly satisfied, yielding low latency and higher QoE. Figure 1 shows such a combined network architecture where a big data platform deployed at the core site is in charge of tracking/predicting users’ demand, whereas cache-enabled BSs store the strategic contents predicted by the big data platform. The following sections examine the architecture details.

**CACHE-ENABLED BSs**

Let us assume a small cell network composed of $N$ small cells, where backhaul link and wireless link capacities of small cell $n$ are denoted by $C_n$ and $C_{n'}$, respectively. We assume that $C_n < C_{n'}$ reflects a limited backhaul capacity scenario [5]. A set of users are requesting a total number of $D$ contents during $T$ time duration from a library of $\mathcal{F} = \{1, \ldots, F\}$ where each content $f$ in this library has a size of $L(f)$ with $l_{\min} < L(f) < l_{\max}$ and a finite bit rate requirement of $B(f)$ during its delivery. To offload the capacity-limited backhaul, small BSs (SBSs) are equipped with finite storage capacity and cache a subset of contents from the library $\mathcal{F}$. However, due to the sheer volume of contents and users, it is very challenging to process and extract useful information to cache all users’ contents at BSs, mainly due to limited storage constraints and lack of sufficient backhauls.

As alluded to earlier, minimizing the backhaul load via edge caching is very challenging. In this regard, a joint optimization of content popularity matrix (denoted by $P$ where columns are contents, and rows are users or BSs depending on the scenario) and content cache placement at specific small cells are required while considering content sizes, bit rate requirements, backhaul, and so on. Moreover, limited storage capacities of SBSs, the backhaul and wireless links, large library size, and number of users with unknown ratings (i.e., empirical value of content popularity) have to be considered while dealing with an intractable cache decision [11]. Assuming that this intractable cache placement can be handled with greedy or approximate approaches [11, 12], the SBSs learn and estimate the sparse content popularity/ratings. The following subsection is dedicated to this task.

**BIG DATA PLATFORM FOR ANALYSIS**

In this section, a general big data processing framework for analyzing users’ data traffic is discussed. The purpose of this platform is to store users’ data traffic and extract useful information
for proactive caching decisions. Supposing that Hadoop is deployed inside the core site of an MO, some of the requirements of this platform for our analysis are as follows.

**Huge Data Volume Processing in Less Time:** In order to make proactive caching decisions, the data processing platform inside the mobile core network infrastructure should be capable of reading and combining data from disparate data sources and delivering intelligent insights quickly and reliably. For this reason, after mirroring the data and streaming interface through network analyzing tools, the collected raw data need to be exported into a big data storage platform such as the Hadoop distributed file system (HDFS) via enterprise data integration methods (e.g., Spring Integration) for detailed analysis.

**Cleaning, Parsing, and Formatting Data:** Data cleaning is an essential part of the data analysis process. In fact, before performing any machine learning and statistical analysis on data, data itself has to be cleaned, and usually this process takes more time than the machine learning analysis. Indeed, there are multiple steps involved in the data cleaning process. First, raw data needs to be cleaned. The raw data itself may contain malfunctioning interfaces in the core network. A fast speed of 200 Mb/s at peak hours is observed through one of the mirrored interfaces in the core network. The total average traffic over all regional areas consists of approximately over 15 billion packets in the uplink and over 20 billion packets in the downlink direction daily. This is equivalent to almost 80 TB of total data. This mirrored network traffic is analyzed on a data processing platform that is essentially based on Hadoop. In particular, the big data platform is composed of Cloudera’s Distribution Including Apache Hadoop (CDH4) version on four nodes including one cluster name node, with each node empowered with an Intel Xeon CPU E5-2670 running @2.6 GHz, 32 cores, 132 GB RAM, and a 20 TB hard disk. As stated before, the platform is in charge of extracting the useful information from raw data. In our analysis, the traffic of approximately 7 h (from 12 p.m. to 7 p.m. on Saturday, 21 March 2015) is collected.2 The traces processed on the big data platform have approximately four million HTTP content requests stored in a comma-separated text file format after following steps 1 and 2, as described earlier. After some post-processing (i.e., calculating content sizes), the final-traces table/file, which includes arrival time (abbreviated as FRAME-TIME), requested content (abbreviated as HTTP-URI), and content size (abbreviated as SIZE), is obtained, and is used in the rest of this study. A detailed description of the data extraction process is given in [1]. Note that the data extraction process is specific to our scenario for proactive caching. However, similar studies in terms of usage of a big data platform and exploitation of big data analytics for telecom operators can be found in the literature (e.g., [14]).

**Data Analysis:** Using the formatted data in the HDFS, different data analytics techniques can be applied over header or/and payload information of both control and data planes using high level query languages such as Hive Query Language (HiveQL) and Pig Latin. The aim of such a step is to find relationships between control and data packets, such as the location or core network. A fast speed of 200 Mb/s at peak hours is observed through one of the mirrored interfaces in the core network. A fast speed of 200 Mb/s at peak hours is observed through one of the mirrored interfaces in the core network. The total average traffic over all regional areas consists of approximately over 15 billion packets in the uplink and over 20 billion packets in the downlink direction daily. This is equivalent to almost 80 TB of total data. This mirrored network traffic is analyzed on a data processing platform that is essentially based on Hadoop. In particular, the big data platform is composed of Cloudera’s Distribution Including Apache Hadoop (CDH4) version on four nodes including one cluster name node, with each node empowered with an Intel Xeon CPU E5-2670 running @2.6 GHz, 32 cores, 132 GB RAM, and a 20 TB hard disk. As stated before, the platform is in charge of extracting the useful information from raw data. In our analysis, the traffic of approximately 7 h (from 12 p.m. to 7 p.m. on Saturday, 21 March 2015) is collected.2 The traces processed on the big data platform have approximately four million HTTP content requests stored in a comma-separated text file format after following steps 1 and 2, as described earlier. After some post-processing (i.e., calculating content sizes), the final-traces table/file, which includes arrival time (abbreviated as FRAME-TIME), requested content (abbreviated as HTTP-URI), and content size (abbreviated as SIZE), is obtained, and is used in the rest of this study. A detailed description of the data extraction process is given in [1]. Note that the data extraction process is specific to our scenario for proactive caching. However, similar studies in terms of usage of a big data platform and exploitation of big data analytics for telecom operators can be found in the literature (e.g., [14]).

**System Parameters and Studied Methods**

In the numerical setup, we assume that D contents are requested from the processed data (i.e., from the final-traces table) over a time interval of 6 h 47 min. Information on FRAME-TIME, HTTP-URI, and SIZE are also taken from the final-traces table. Then the requests are pseudo-randomly assigned to the BSs. The wireless link capacities of small cells, backhaul link, and storage capacities are set to identical values within each other for ease of revealing the caching gains. The list of simulation parameters are summarized in Table 1. The global procedure contains the following two major steps.

**Estimation of content popularity P (where columns are contents and rows are BSs):** This is done on the big data platform by processing a large amount of collected data and exploiting machine learning.

---

1 In fact, the general trend of data in the network is following exponential growth: in 2012, the total average data traffic per day was over 7 TB in both uplink and downlink.

2 Note that the size of raw data is around 1.2 TB for observed time duration 7, and for offline processing, it can take up to five days to extract the relevant headers from this data using a single server.
learning tools. Two methods are examined in the numerical setup:

- Ground truth: The $P$ matrix is constructed by considering all available information in the final-traces table. The matrix has 6.42 percent of rating density in total.
- Collaborative filtering: 30 percent ratings available in the final-traces table are picked uniformly at random for training of $P$ matrix estimation. Then, the remaining missing entries/ratings in the traces are predicted via the regularized singular value decomposition (SVD) from collaborative filtering (CF) methods [15].

Caching strategic contents:
- The cache decision procedure at the BSs is made by storing the most popular contents greedily at the SBSs until no storage space remains, as in [1].
- In regard to the performance metrics: request satisfaction, as the QoE metric, is defined as the amount of contents delivered at a given target rate, and backhaul load corresponds to the percentage of the traffic passing over the backhaul links over the total possible traffic volume induced by the content requests. Detailed analytical formulas of request satisfaction and backhaul load can be found in [1].

NUMERICAL RESULTS AND DISCUSSIONS

In this section, based on the available information in the final-traces table, we conduct a numerical study to reveal the gains of caching. The impact of storage size on users’ request satisfaction is plotted in Fig. 2. Therein, 0 percent of storage size corresponds to no caching, whereas 100 percent of storage is equivalent to caching the entire library (17.7 GB). In the figure, we note that the users’ request satisfaction has a monotonically increasing behavior, and, somewhat intuitive, 100 percent of satisfaction is achieved in both methods when the complete content catalog (with 100 percent of storage size) is stored by fixing parameters in our setting to plausible (and realistic) values in order to see the regimes where 100 percent satisfaction is achieved. However, a performance gap between the ground truth and CF is observed until 79 percent of storage size, which is mainly due to the estimation errors. For instance, when the BSs have 40 percent of storage size for caching, the ground truth yields 89 percent satisfaction, whereas the performance of CF stays at 75 percent.

Figure 2 also shows the impact of storage size on the backhaul load/usage. In the figure, we see that both methods yield less backhaul load (higher offloading gains). For instance, having 79 percent of storage size at BSs, both methods offload 98 percent of backhaul. However, the ground truth outperforms the CF method since it has complete information of the content ratings. On the other hand, after a certain storage size, a dramatic decrease of backhaul is observed in both approaches. Compared to previous works, which mostly consider identical content sizes, we are dealing with real traces with non-identical content sizes. In the simplest form, one can write the backhaul load of a particular content as $load = popularity \times size$ if not cached. Therefore, relatively

![Figure 2. Numerical results for proactive caching at base stations.](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Time duration</td>
<td>6 hours and 47 minutes</td>
</tr>
<tr>
<td>$D$</td>
<td>Number of requests</td>
<td>422,529</td>
</tr>
<tr>
<td>$F$</td>
<td>Number of contents</td>
<td>16419</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of small cells</td>
<td>16</td>
</tr>
<tr>
<td>$l_{\text{min}}$</td>
<td>Min. size of a content</td>
<td>1 B</td>
</tr>
<tr>
<td>$l_{\text{max}}$</td>
<td>Max. size of a content</td>
<td>6.024 GB</td>
</tr>
<tr>
<td>$B(f)$</td>
<td>Bit rate of content $f$</td>
<td>4 MB/s</td>
</tr>
<tr>
<td>$\sum c_m$</td>
<td>Total backhaul link capacity</td>
<td>3.8 MB/s</td>
</tr>
<tr>
<td>$\sum c_n$</td>
<td>Total wireless link capacity</td>
<td>120 MB/s</td>
</tr>
</tbody>
</table>

Table 1. List of simulation parameters.
less popular but very large contents might lead to such behavior on the backhaul load if not cached at the SBSs. This points out the importance of taking into account content sizes in caching decision making, which reflects a more practical/realistic characterization of backhaul usage.

Figure 3 illustrates the evolution of users’ request satisfaction with respect to the backhaul capacity ratio, defined as the ratio of total backhaul link capacity \( \Sigma_n C_n \) over total wireless link capacity \( \Sigma_n C_n' \). It is clear from the figure that increasing the backhaul link capacity yields higher satisfaction in both the ground truth and CF approaches. This is due to the fact that the bottleneck in the backhaul becomes less relevant with the increment of this ratio.

The above performance results demonstrate the case with 30 percent of rating density in CF. However, it is clear that by increasing the training rating density of CF, less estimation error and hence closer satisfaction gains to ground truth are expected. In order to show this, Fig. 4 demonstrates the effect of training rating density on root mean square error (RMSE) where the error is defined as the root mean square of the difference between users’ content satisfaction of the ground truth and CF approaches over all possible storage sizes. Figure 4 clearly validates the fact that performance of CF can be improved via higher training rating density.

**CONCLUSIONS AND FUTURE WORK**

In this article, we have introduced a proactive caching architecture for 5G wireless networks by processing a huge amount of available data on a big data platform and leveraging machine learning tools for content popularity prediction. Additionally, relying on this prediction and using extracted traffic information from this data, the gains of caching have been investigated throughout numerical studies. One possible direction of this work is to investigate the proposed big data analysis framework in a real-time fashion. For this, recent frameworks that exist in the Hadoop eco-system such as Apache Spark and its built-in libraries Spark Streaming for real-time data processing and MLlib for machine learning libraries, are of interest.

**REFERENCES**


**BIOGRAPHIES**

ERGEN ZETIS received his Ph.D. degree in February 2011 from the Department of Electrical and Computer Engineering at Stevens Institute of Technology, Hoboken, New Jersey. Previously, he received his M.S. and B.S. degrees from the Department of Electrical and Electronics Engineering at Middle East Technical University, Ankara, Turkey, in 2006 and 2004, respectively. He worked as an R&D engineer for Avea, a mobile operator in Turkey, between 2011 and 2016. He is currently with Turk Telekom Labs working as an innovation and applied research engineer. His research interests are in the area of telecommunications and big data networking. He is a recipient of the Exemplary Reviewer for IEEE Communications Letters Award, 2011. He recently gave tutorial presentation at IEEE NOMS 2016. He is primarily responsible for carrying out European Commission and nationally funded research activities at Turk Telekom Labs.

EJDER BAŞTUG (ejder.bastug@centralesupelec.fr) is currently a postdoctoral researcher at CentraleSupelec. He obtained his Ph.D. from CentraleSupelec in December 2015, under the guidance of Prof. M. Demetriou and Jean-Claude Belfiore. His Ph.D. topic was on distributed caching methods in small cell networks, whereas his main research interests are related to stochastic geometry and machine learning tools for wireless communications. He has been on the Executive Committee of IEEE WCNC 2014, Organizing Committee of IEEE BlackSeaCom 2015, Technical Program Committee of EuCNC 2015, and Chair of several international workshops. He is on the Organizing
Committee of IEEE ICC 2017, to be held in Paris. He was the recipient of the Supélec Foundation’s Best Ph.D. Theses/Publications Prize 2015 and the IEEE Communications Society Best Tutorial Paper Award 2016.

Mohsen Bennis (UM) received his M.Sc. degree in electrical engineering jointly from EPFL, Switzerland, and the Eurecom Institute, France, in 2002. From 2002 to 2004, he worked as a research engineer at IMRA-EUROPE investigating adaptive equalization algorithms for mobile digital TV. In 2004, he joined the Centre for Wireless Communications (CWC) at the University of Oulu, Finland, as a research scientist. In 2008, he was a visiting researcher at the Alcatel-Lucent chair on flexible radio, Supélec. He obtained his Ph.D. in December 2009 on spectrum sharing for future mobile cellular systems. His main research interests are in resource management for future context-aware networks, game theory, and machine learning in 5G networks. His research interests are in software development, SDN, system integration, and network management systems. His research interests are in the area of software development methodologies, big data analytics, and data mining.

Ahmet Salih Ery received his B.S. degree in computer engineering from Istanbul Technical University, Turkey, in 2008. From December 2008 to August 2011, he was with Vodafone Turkey working as a software development engineer on the Service Development Platform. In August 2011, he joined Turkcell as a software development engineer for the Campaign Management System. In December 2013, he joined Avealabs of Avea as a senior software engineer. His current research interests include SDN, big data analytics, social media analysis, and service oriented architectures. He is also studying for his Ph.D. degree in the Kadir Has University Management Information Systems program since September 2014.

Mérouane deBBah entered Ecole Normale Supérieure de Cachan, France, in 1996 where he received his M.Sc and Ph.D. degrees. He worked for Motorola Labs, Saclay, France, from 1999 to 2002 and at the Vienna Research Center for Telecommunications, Austria, until 2003. From 2003 to 2007, he was with the Mobile Communications Department of Institut Eurecom, Sophia Antipolis, France, as an assistant professor. Since 2007, he has been a full professor at CentraleSupélec, Gif-sur-Yvette, France. From 2007 to 2014, he was the director of the Alcatel-Lucent Chair on Flexible Radio. Since 2014, he is vice-president of Huawei France R&D and director of the Mathematical and Algorithmic Sciences Lab. His research interests lie in fundamental mathematics, algorithms, statistics, and information and communication sciences research. He is an Associate Editor-in-Chief of Random Matrix: Theory and Applications and was an Associate and Senior Area Editor for IEEE Transactions on Signal Processing in 2011–2013 and 2013–2014, respectively. He is a recipient of the ERC grant MORE (Advanced Mathematical Tools for Complex Network Engineering). He is a WWRF Fellow and a member of the academic senate of Paris-Saclay. He has managed 8 EU projects and more than 24 national and international projects. He received 15 best paper awards, including the 2007 IEEE CLOBECOM best paper award, WiOpt 2009 best paper award, 2010 Newcom++ best paper award, WJN CogCom Best Paper 2012 and 2013 Awards, 2014 WCNC Best paper award, 2015 ICC best paper award, 2015 IEEE Communications Society Leonard G. Abraham Prize, 2015 IEEE Communications Society Fred W. Ellersick Prize, and 2016 IEEE Communications Society Best Tutorial paper award as well as the Valuetools 2007, Valuetools 2008, CrownCom2009, Valuetools 2012, and SAM 2014 best student paper awards. He was the recipient of the Mario Boella award in 2005, the IEEE Glaveux Prize Award in 2011, and the Qualcomm Innovation Prize Award in 2012. He is a co-founder of Ximinds and Ulanta.