Analysis of Joint Parallelism in Wireless and Cloud Domains on Mobile Edge Computing over 5G Systems

Glaucio H.S. Carvalho, Isaac Woungang, Alagan Anpalagan, and Muhammad Jaseemuddin

Abstract: **The realization of mobile edge computing (MEC) over emerging fifth (5G) generation of wireless systems arises as a driving-force in the future of cloud computing. In order to cope with the volume, variety, and velocity of the IoT traffic while making optimal use of the network infrastructure, a synergistic exploitation of MEC and 5G should be put forward to support advanced resource management applications. In this paper, we propose the use of joint parallelism between wireless and cloud domains to efficiently respond to mobile data deluge. We review the literature, discuss the enabling network architecture, potentials, challenges, and open issues related to the realization of such level of parallelism. We present and evaluate two design examples – parallel computation offload method (PCOM) and parallel transmission and storage method (PTSM)—which outline the benefits of parallelism for computation-hungry and storage-hungry applications, respectively. Results of our optimization formulation show that PCOM and PTSM are able to make an efficient use of the network resources and support a heavy instantaneous workload by means of the parallelism.**

Index Terms: **Mobile cloud Computing, mobile edge computing, cloud computing, 5G systems, parallelism, mobile storage.**

I. INTRODUCTION

LATENCY-AWARENESS is a driving-force behind the cloudification process faced by mobile network operators (MNOs). This process, which is featured by the amalgamation of a base station (BS) or an access point (AP) and a virtualized edge server, is overarched by what has been recently coined as mobile edge computing (MEC)—a step further on the mobile cloud computing (MCC) pathway.

Fig. 1 shows a representative scenario of a MEC over emerging fifth generation (5G) mobile communications system. As shown, MEC arises as compelling technology to successfully cope with the growing demand for computing resources and storage capacity by enabling a one-hop away connectivity between mobile users and edge-servers through a BS or AP. Since MEC is built at the top of 5G systems, it might harvest several benefits from the small-cell densification and the overlapping nature of heterogeneous network (HetNet) deployment that allow mobile users to associate with the best BS or AP while maintaining a high mobility profile. Moreover, the massive distribution of BSs and edge servers become an enabler for the realization of the Internet of Things (IoT) paradigm [1] [2].

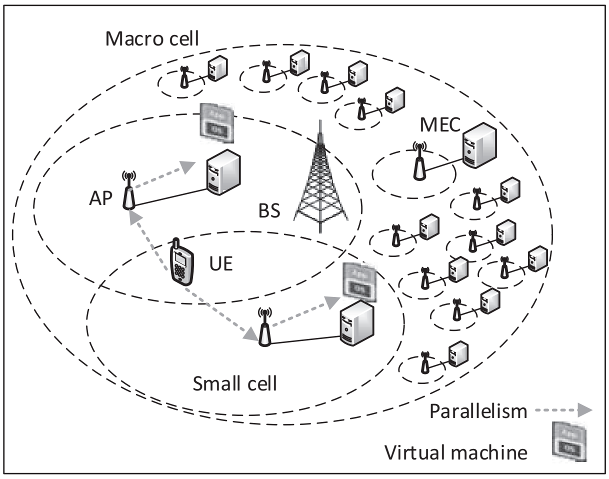

Fig. 1. MEC over emerging 5G systems.

Recently, MEC has attracted a great deal of attention from academia and industry. The need to provide latency-sensitive applications with timely computing and storage resources underpins most of the breakthroughs in literature. Since the success of sophisticated mobile cloud application depends on the effectiveness of MEC, which in turn is tightly coupled to the allocation of communications and computing resources, resource management has become one of the vital aspects within the design of a MEC solution. In this sense, joint computation and communications resource management solutions have been studied in [3] from a signal processing viewpoint and in [4] from a revenue sharing perspective. However, the volume, variety, and velocity of the IoT traffic will pose a threat to the service provisioning in such a way that mobile network operators might face issues in order to guarantee the strict latency-control requirement of computation-hungry and storage-hungry MCC applications. Thus, a solution must be sought to make a better use of the network resources while effectively supporting advanced MCC ap-

Manuscript received February 14, 2018 approved for publication by Phone Lin, Division III Editor, September 21, 2018.

This work is partially supported by a grant from the National Science and Engineering Research Council of Canada (NSERC) held by the 2nd author—REF #RGPIN/2017-04423, NSERC Discovery Grants held by the 3rd and 4th authors.

G. H.S. Carvalho is with the School of Applied Computing, Faculty of Applied Science and Technology (FAST) at Sheridan College, 1430 Trafalgar Rd, Oakville, ON, L6H 2L1. email: glaucio.carvalho@sheridancollege.ca.

I. Woungang is with the Department of Computer Science at Ryerson University, 350 Victoria Street, Toronto, ON, M5B 2K3, Canada. email: iwoungan@scs.ryerson.ca.

A. Anpalagan and M. Jaseemuddin are with the Department of Electrical and Computer Engineering at Ryerson University, 350 Victoria Street, Toronto, ON, M5B 2K3, Canada. email: {alagan, jaseem}@ee.ryerson.ca.

G. H.S. Carvalho is the corresponding author.

Digital Object Identifier: 10.1109/JCN.2018.000089

plication.

Despite the present progress, surprisingly, parallelism which is one of the most fundamental techniques that is used to speed up a task by triggering simultaneous manipulation of its segments, has been neglected in the literature. Parallelism in wireless domain and MCC domain were introduced individually, but the best of our knowledge, a framework that enables the parallel transmissions and the parallel processing/storage of a computation task/media in a timely fashion in MEC over 5G is unprecedented by any work. This principle is illustrated in Fig. 1 where the user equipment (UE) fragments the application into two parts and concomitantly transmits them to be processed into two edge servers. More sophisticate schemes with successive code or media partitioning might be engineered to efficiently use the system resources by spreading the application's segments and execute or store them distributedly if the system has no resource to handle them locally.

In order to bridge this gap in literature, this paper focuses exclusively on the joint parallelism over the wireless and cloud domains as a way of optimizing the communications and processing/storage processes. To this end, we first present a review of the literature in terms of resource management in cloud and wireless networks and parallelism in both domains. Second, we present the enabling network architectures for parallelism and unveil the opportunities to use parallelism on MEC. Third, we outline the potential benefits which might be harvested from the parallel transmissions, processing, and storage. Forth, we exemplify the application of parallelism in MEC by presenting two representative design instances in which a computation-hungry task or storage-hungry task opportunistically exploits the overlapping structure of 5G system to communicate with multiple edge virtualized servers through their corresponding BSs in order to perform their services. We leverage linear programming as the mathematical tool to obtain the optimal task partitions or data chunk based on the network and computing resources availability.

The remainder of the article is organized as follows. Section II presents the recent breakthroughs in terms of resource management on wireless and cloud computing taking into account the parallelism into both fields. Section III outlines the evolution of network architecture and shows how parallelism can be implemented on it. Next, we shed the light on the benefits, potentials, challenges and open issues behind the realization of parallelism in MEC over emerging 5G systems. In order to illustrate the principles addressed on the current paper, we present design examples on Sections V and VI, while in Section VII discusses the computational aspects and major assumptions of the proposed algorithms. Finally, Section VIII concludes the article.

II. REVIEW OF THE LITERATURE

A. Resource Management

From a resource management perspective, the intrinsic diversity of existing technologies in wireless and cloud realms arises as an attractive feature to be exploited in order to successfully deal with the increasing latency that has been perceived by existing mobile computing applications over stand-alone deployments. In wireless domain, the overlapping and collaborative nature of radio access technologies (RATs) is an indication of the *wireless network diversity*. In analogy, clouds that have different peculiarities such as processing capacities, storage capacities, geo-locations, elasticity, security level, among others can be also characterized by the *cloud diversity*. In this paper, we exploit the wireless network diversity and the cloud diversity to realize the parallelism in MEC over 5G systems.

Resource management has been extensively studied in mobile cloud computing. In [5], Chen engineered a game-theoretical model to specify a decentralized computation offloading strategy. The model took into account both computation and communication aspects in the game definition. By analyzing the game structure, the author showed that the Nash equilibrium exists. Results demonstrated that the model is scalable and efficient.

Liang *et al.* [6] proposed a resource allocation scheme in a multi-cloud ecosystem whose objective was to maximize the overall reward for the cloud service provider and the user. The proposed methodology is featured by a SMDP-based controller that decides, based on the resources availability, weather to accept a service request in the home cloud or transfer it to an adjacent cloud. Results illustrated the superiority of this approach when confronted with the greedy policy.

Barbarossa *et al.* [3] put forward a distributed cloud computing architecture over a HetNet. Over this architecture, a service request might be assigned to the closest cloud in order to get both latency and energy consumption optimized. Additionally, whenever the nearest cloud is unable to meet the request, the second nearest one is assigned, and so on until the assignment of a remote cloud is achieved. The resource allocation optimization, which is tackled at the signal processing level, was devised stressing the need to reduce the energy consumption and latency as an integral part of the 5G systems.

Gkatzikis and Koutsopoulos [7] investigated the role played by the virtual machine (VM) migration in an ecosystem with multiple clouds taking into account a MEC infrastructure and a back-end cloud to enhance the service provisioning. The authors contrasted three VM migration approaches, namely, no-migration strategy, load-aware migration strategy, and load- and mobility-aware migration strategy. Considering the task lifetime as a criterion, it was shown that the latter exhibits the shortest task lifetime. Additionally, the authors compared the benefits of each type of task decision making: that initiated by the cloud service provider, by the server, and by the task itself. Focusing on the same criterion, the cloud service provider-initiated migration strategy stands out by achieving the best performance followed by the server-initiated migration.

Felemban *et al.* [8] proposed a multimedia-driven MCC architecture defined by integrated MEC and distributed multimedia data centers. The principle behind the architecture is to take the multimedia information stored at the multimedia data centers to the MEC virtualized servers at the edge of the network that will pass the information to the user via a BS with a shortened latency. No numerical results were presented but a general discussion on how the proposed MCC architecture should perform based on criteria such as complexity and data dropping ratio is outlined.

In [9], an analysis was placed on how the mobile cloud ser-

Table 1. Summary of resource management works in MCC.

Paper	**Description**	**Multiple RATs**	**Multiple clouds**	**Tool**
Chen [5]	Decentralized computation offloading strategy in MCC. The communication model defines the uplink data rate while the computation model defines the computation task as a function of the size of the computation input data and the number of CPU cycles required by the task. Furthermore, the time and the energy required to process the task locally and remotely are considered.	No	No	Game theory
Liang *et al.* [6]	Resource allocation in an inter-domain multi-cloud ecosystem. The objective function takes into account conflicting parameters such as cost of the task offloading and the VM utilization, income due to the service request's acceptance, payment due to inter-domain transferring, MD's energy and resource expense.	No	Yes	SMDP
Barbarossa *et al.* [3]	Joint computation and communication optimization with emphasis on energy and latency minimization. Progressive analysis of different scenarios. Starting with resource allocation of radio resources in a single-user scenario and ending in a multi-user scenario, multi-RAT, and multi-cloud scenario.	Yes	Yes	Convex Optimization
Gkatzikis and Koutsopoulos [7]	Discussion on the benefits, challenges, and open issues on VM migration in a MCC environment.	No	Yes	Search Algorithm
Felemban *et al.* [8]	Discussion on a functional layered multimedia-driven MCC architecture with distributed Cloudlets, BSs, and multimedia data centers, functions and protocols	No	Yes	No numerical results
Kaewpuang *et al.* [9]	Detailed analysis on resource and revenue management in cooperative mobile CSPs. Definition of a comprehensive framework containing various optimization models for different situations. For instance, based on the knowledge of the resources availability and user demand (nominal, probability distributions, and range), a different optimization formulation is applied with the same goal of maximizing the revenue.	Yes	Yes	Linear Programming, Stochastic Programming, Robust Optimization, Markov Chain, Game Theory
Lei *et al.* [11]	Discussion on the main functional building blocks of a HetNet-based MCC architecture as well as the state-of-the-art for each of the blocks	Yes	No	Simulation
Zhang *et al.* [10]	Detailed characterization of the collaborative task offloading process over different types of wireless channels including the NP completeness of the problem, the derivation of two sub-optimal solutions with reasonable computational complexities. In general, it was concluded that the collaborative task offloading process considerable saves the MD's energy	No	No	Search Algorithm, LARAC Algorithm, Simulation
Carvalho *et al.* [4]	Analysis of the marriage between Intercloud and HetNets considering different design issues and challenges such as energy-efficiency and the development mismatch between cloud and wireless technologies. Optimal user association and revenue sharing were approached. In general, the numerical results outlined the benefits for a collaborative interoperation between mobile network operators and public cloud service providers	Yes	Yes	Integer Linear Programming model and Shapley concept

vice providers can share radio and computing resources with one another to form coalitions with the objective to maximize their revenues. After presenting a linear programming, stochastic programming, and robust optimization formulations, the authors devised a revenue management approach, which relies on the concept of core and Shapley value to define the revenue share among the mobile CSPs. Finally, a game model was defined to determine whether a mobile CSP should cooperate. The results showed some evidence of the benefits collected by cooperative mobile cloud service providers in terms of an increase in their revenues.

Collaborative task offloading process, which is featured by an alternating execution of the task on the cloud and on the user equipment, is under analysis in [10]. Zhang *et al.* considered the problem of optimal energy collaborative task execution. After specifying the system as a constrained stochastic shortest path problem, the authors remarkably determined the NP completeness of the collaborative task offloading process. From this point onwards, an enumeration algorithm was proposed to efficiently solve the problem in polynomial time. The enumeration algorithm was proved to work for three types of wireless channels: block-fading channel, IID stochastic channel, and a Markovian stochastic channel. Last but not least, a heuristic, based on the Lagrangian Relaxation Based Aggregated Cost (LARAC) algorithm, was devised to approximate the optimal solution with a complexity lower than that of the enumeration algorithm. Considering a single cloud and RAT scenario, results showed that the collaborative task offloading process is more energy-efficient than the local processing execution and the remote execution.

Taking the latency minimization as a motivation, the work in [11] discussed the adoption of the HetNet as a fundamental step in the development of MCC. To show the benefit of a HetNet deployment, the authors used the task offloading process transmission over a HetNet and a macro cell-only deployment and concluded that despite an uptick in the power consumption, the transmission delay perceived by the offloading process is reduced in the HetNet.

In [4], Carvalho *et al.* presented a comprehensive analysis of the interoperation between Intercloud and HetNets. After discussing the challenges and design aspects behind this interoperation, the authors came up with a revenue maximization framework in a coalition between a mobile network operator and public service providers considering a scenario with HetNet deployment, MEC, and a public cloud service provider. The proposed framework is based on an integer linear programming model that provides the maximum revenue for each coalition formed among the players while giving the optimal end-to-end mapping between the users and a cloud data center through a base station. As in [9], the concept of Shapley value was applied to individualize the contribution of each player based on the maximum revenue.

A summary of the related works on MCC is provided in Table 1 along with their characteristics in terms of number of RATs and cloud data centers as well as the tools used. A remarkable feature in Table 1 is the fact that the joint resource allocation of wireless and cloud resources is a current feature in the design of MEC over 5G systems. However, despite the recent breakthroughs, parallelism taking into consideration wireless domain and cloud domain jointly is still a missing figure in the design of MCC solutions. To shed the light on this issue, we present as follows an analysis of how parallelism has been taking into account individually in the design of wireless and cloud systems.

B. Parallelism in Wireless Networks

The exploitation of wireless network diversity for parallel transmissions enables mobile network providers to harvest numerous benefits such as increased system capacity and reduced operational costs, to name a few. By the same token, mobile computing applications can benefit from an enhanced connectivity, which might considerably improve the system level performance and lead to a reduced latency.

For instance, aiming at the maximization of the system capacity, the work of Choi *et al.* [12] exploited the parallel uplink transmissions over multiple RATs in a HetNet by jointly addressing the bandwidth and the power allocation. A performance comparison with a traditional single RAT transmission method unveiled the increased capacity achieved by parallelism. Advancing the state-of-the-art a step further, the work of Miao *et al.* [13] dealt with the problem of maximizing the total system throughput in a HetNet taking into account the QoS demand for distinguishing services traffic classes. A performance comparison between [12] and [13] revealed that the work in [12] failed in differentiating QoS while the one in [13] did not. From the application viewpoint, [13] showed that some mobile users obtained at least the minimum QoS requirement while others got even higher QoS levels. Following the same line, but considering QoE and access price instead of QoS only, Song *et al.* [14] showed that the exploitation of parallelism is still beneficial for service providers and applications alike. Translating [12]–[14] to MCC, we could infer that a reduced latency might be expected by considering the exploitation of the parallelism.

It is important to state that [12]–[14] focus on uplink parallel transmissions. However, similar conclusion is drawn when downlink parallel transmissions are used [15]. Despite the fact that currently MCC applications have the heaviest workload on the uplink direction (e.g., face recognition), in future, advanced MCC applications might need a symmetric uplink and downlink capacity. In this case, parallelism can also underpin the development of these applications.

C. Parallelism in Cloud Computing

Cloud network infrastructure stands out as the most critical obstacle for the success of parallel applications on cloud. This conclusion, which is based on the execution of scientific applications on cloud, is the commonality of [16]–[18]. Experimental results considering the cluster compute quadruple extra large instances on Amazon EC2 in [16] lined up with the points presented previously. In particular, it was shown that the network might be the major system bottleneck for scientific simulations. To overcome such drawback, the authors suggest to favor communication between VMs with high performance while reducing between VMs with a lower performance. In [17], Freniere *et al.* tested the Amazon's Elastic Compute Cloud (EC2) service for a GPU-accelerated application against a local HPC cluster. Results clearly indicated the EC2's inappropriateness to keep up with the HPC cluster. Still on the EC2 arena, Sadooghi *et al.* [18], who presented the most comprehensive analysis considering a cloud computing environment to scientific applications up to the time of the written of this paper, reached similar conclusion. Among the major obstacles listed in [17] and [18] is the fact that the cloud communication system which is based on commodity network is unmatchable to InfiniBand system used by the benchmarks.

Despite the fact that [16]–[18] dealt with scientific applications, which differs from mobile parallel programs, when it comes to the performance of parallel programs running on cloud data centers, their conclusions provide network engineers with actionable insights on the design of MEC infrastructure. In order to support advanced parallel applications such as machine learning-based video analytics, edge servers must be powerful, and so the communications infrastructure.

To unleash the development of applications that fully exploit the joint parallelism in wireless and cloud systems, Middleware should be developed to simplify and expedite the software development by abstracting the intricacies of distributed applications, heterogeneity of wireless networks, cloud infrastructure, operating systems, UEs, and protocols. This feature is paramount in face of the growing complexity of parallel applications along with the number of options to choose from when it comes to the implementation of parallel programs. The work of D'Angelo and Rampone [20] highlighted the complexity involved in the conception of HPC-oriented parallel programming from its design considering dynamic programming to its implementation in Java message passing library (MPJ). The presented methodology focuses on scientific applications; nevertheless, it provides insights on the specifics of optimum parallel programming and raises the visibility of the need to set mobile developers free from this burden in order to mass-produce advanced parallel mobile applications.

Given the tremendous potential for advanced parallel applications in mobile cloud computing, initiatives such as [21]–[24] might pave the way for the development of parallel applications on MEC over emerging 5G systems. In [21], Badia *et al.* proposed a programming framework, which is called COMPSs, that automates the parallelization of sequential existing applications written in Java, C/C++, and Python and enables their execution in the underlying infrastructure. Based on [21], Lordan *et al.* in [22] and [23] extended the COMPs philosophy to the cloud and mobile cloud environments, respectively. The work in [22] presented a programming framework named Service Superscalar (ServiceSs). In addition to parallelizing sequential programs, ServiceSs allows the execution of programs on dif-

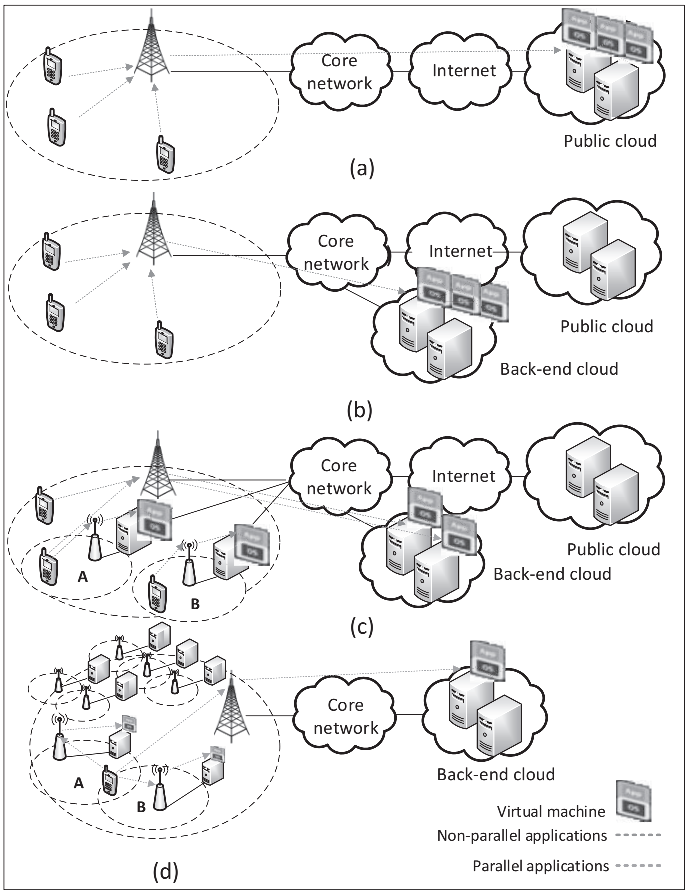

Fig. 2. Mobile cloud computing network architecture: (a) Traditional network design, (b) back-end based network design, (c) MEC-based network design over emerging 5G systems, and (d) densified wireless network access supporting MEC and potential parallel MCC applications.

ferent cloud providers without the need to change the existing code. This feature simplifies and expedites the development process by releasing software engineers from the knowledge of specifics APIs. Moreover, ServiceSs is conceived to maximize the interoperability with different IaaS and PaaS service provides by making the existing services to run on hybrid deployments. Similarly, COMPSs-Mobile, which is presented in [23], automatically parallelizes and offloads regular Android applications to the cloud. In [24], Lin *et al.* proposed the parallelization of cloud application as a service within a framework called Parallel Computing Framework as a Cloud Service (PCFCS), which releases programmers from the cumbersome task of configuring Hadoop—the open-source parallel computing version of MapReduce. As part of the cloud orchestration process, [22]–[24] are infrastructure-independent and provides an abstraction layer that frees programmers from getting acquainted about the underlying infrastructure details.

III. ENABLING NETWORK ARCHITECTURE

Fig. 2 describes the evolution of the MCC architecture from a centralized cloud data center to a fully distributed system. Whilst Fig. 2(a)–(b) show that cloud data centers are reachable from a macro cell only, Fig. 2(c)–(d) unveil the full potential of advanced cloud-based access in which MEC and 5G are holistically exploited in order to realize multiple and simultaneous transmissions, processing, and storage across the small BSs to MEC edge servers.

As it can be seen, the need to strictly control the latency has motivated the mobile network operators to shift from a public cloud data center in Fig. 2(a) to a back-end cloud attached to their core network, Fig. 2(b). The benefit of this architecture is significant since transmissions across the Internet are only needed when the back-end cloud is running out of capacity.

In spite of achieving a diminished latency, the conceptual design in Fig. 2(b) still mirrors the one in Fig. 2(a) in the sense that the computation workload is heavily handled by a resourceful centralized cloud data center. However, with the explosion of MCC applications and the IoT data deluge, there is a necessity to bring the cloud closer to the users to avoid flooding the core network and the Internet with their traffic loads. Additionally, the heterogeneous distribution of potential data sources (users and IoT devices) claims for a more distributed and scalable architecture. To cope with these confluent objectives, the network design outlined in Fig. 2(c) presents a multi-layer cloud deployment with MEC servers at the edge, back-end cloud on the center, and the public cloud data center providing additional capacity in case of unmatched spikes in the traffic load. Fig. 2(c) also shows two types of computation offloading processes. The first one, which is a parallel mobile application, takes place on the cell site **A** in which a UE might simultaneously transmit across to the macro BS and the small cell BS to the back-end cloud and the MEC servers, respectively. In the second one, which occurs in the cell site **B**, the UE can associate with the its home MEC server through its BS.

Fig. 2(d) emphasizes the ultra dense deployment of small cells and MEC infrastructure and the potential it can unleash in terms of parallelism. As we can see, with the ultra dense deployment there might have regions where the UE will be under the coverage of multiples small cells (intersection of the cell sites **A** and **B**) and the macro cell. In such a case, parallelism can be fully orchestrated to amplify the potentials of MCC applications. Consequently, the network architectures in Figs. 2(c) and (d) enable application developers to intelligently coordinate the use of parallelism on MEC over 5G systems.

Fig. 3 displays sophisticate application partitioning strategy in which successive segmentation is performed across the system in order to optimize the use of the network resources while expediting the application execution by distributing its segments if the system has no resource to handle them locally. As shown the application is initially segmented at the UE, simultaneously transmitted across two BSs where they are again segmented into smaller tasks or chunks to parallelly run or be stored at the MEC servers and at the back-end cloud. Since the system resources are fully leveraged to instantaneously supply the application QoS demand, this type of parallelism might lead into a better performance by enabling the system to handle a heavier instantaneous workload if an effective application parallelization is putting in place.

In order to sub-divide an already parallelized application into more atomic levels, the system will need a very specialized engine to partition the code (parallelization engine) and additional

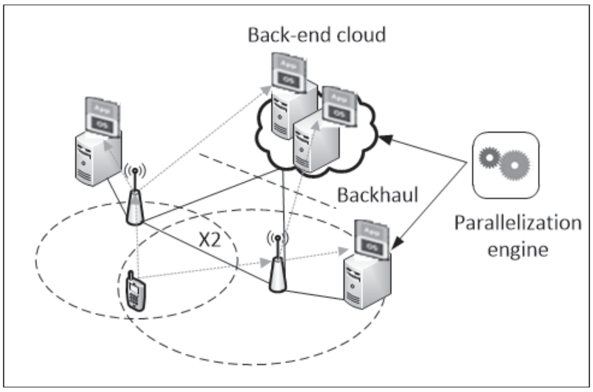

Fig. 3. Sophisticate application partitioning and parallelization engine placement.

computational power to run it. Fig. 3 shows that the parallelization engine might be placed at the MEC server or at the back-end cloud. In both cases, it will be inside the mobile network perimeter which will ensure the system responsiveness.

IV. POTENTIALS, CHALLENGES AND OPEN ISSUES

Previous Sections indicate that there are initiatives in both wireless and cloud arenas individually supporting the parallelism. When it comes to network architecture, it turns out that a joint parallelism might also be supported. As follows, we discuss the benefits, potentials, challenges, and open issues related to the joint parallelism.

From both wireless and cloud computing standpoints, the use of parallelism can lead to a more effective, robust, scalable, and energy-efficient design. Effectiveness is reached by expediting the parallel tasks of the application simultaneously through different wireless networks to be processed into multiples edge serves. Another possibility is the expedition of an application through a single wireless network and the parallelization of the computation workload between the home edge server and the neighboring one if the home edge server is running out of capacity. In both scenarios, the application of optimal control algorithms such as constrained continuous time Markov decision process (C-CTMDP) arises as a tool to derive optimal policies which can maintain the latency between the two servers under control for a varying workload. In this sense, the work in [6] might provide some design guidelines since it specifies when a service request should be transferred between adjacent cloud data centers on MCC. At the cloud level, it is paramount to expedite and automate the VM creation process to accommodate the new requests for service without delay. To this end, the work of Righi *et al.* [19] provides a platform called AutoElastic which can manage the cloud elasticity by optimizing the VM allocation and deallocation procedures. As [22]–[24], AutoElastic also sets programmers free from configuring elasticity and rewriting their codes.

Since there are redundant communication links and edge servers, parallelism might result in a more reliable and robust design if a holistic design is considered. While scalability is an inherent feature of cloud computing, it is not easily realizable in the wireless landscape. Despite the fact that most of UEs are able to sense and connect with multiple networks currently, they are not able to establish simultaneous connections. To overcome such drawback, software defined radio technology, which is a mature technology presently, might be leveraged to enable it.

In 5G systems, energy-efficiency is achieved because the UE's transmission power is reduced thanks to the small-cell deployment and the offloading of energy-hungry computation modules to edge servers. Considering parallelism, energy-efficiency may be enhanced by the exploitation of the intrinsic heterogeneity of BSs, APs, and edge servers. In this case, joint communications and computing resource allocation might be orchestrated to opportunistically and simultaneously transmitted across multiple energy-efficient wireless links and processed/stored into multiple energy-efficient edge servers.

Energy-efficient server consolidation algorithms can also be leveraged to attain energy savings by re-distributing VMs among edge servers in the home MEC data center or between the two adjacent MEC data centers. While the application of server consolidation is a current practice within servers in the same site, its application when considering adjacent MEC sites might be challenging since the VM migration between two sites will result on an overload on the backhaul infrastructure if not controlled properly.

While energy-efficiency and reliability are paramount operational targets to tackle, the design of an energy- and reliability-aware joint wireless and cloud resource management is unprecedented in literature. The reason steams from the fact that reliability increases as redundancy scales up whilst energy savings are achieved by consolidation. Given the contrasting nature of both objectives, how to orchestrate such operation remains unclear. Sharma *et al.* [25] shed the light on this subject considering a cloud computing environment only. However, since the interplay between the BS and the MEC edge servers is also an open question, a viable resource management scheme contemplating both goals is challenging. Nevertheless, if failure is foreseeable, the application of optimal control techniques might pave the way for a practical solution to this problem. For instance, if the mean time between failures is exponentially distributed, then SMDP might be applied to wake up the replicas before the failure occurrence which would satisfy both constraints simultaneously. Given the curse of dimensionality inherently present in MDP-based solutions, reinforcement learning techniques arise as a compelling technique when dealing with densified systems. On the other hand, if the Poisson model fails in capturing the breakdown occurrence, the resulting optimal policy might have an opposite effect because of the extra energy that will be demanded to restart all the exiting parallel applications that were interrupted during the failure and consequently the longer response time perceived by the parallel applications which might violate existing service level agreements (SLAs).

One of the most challenging aspects of the realization of parallelism is the synchronization among the multiple parallel tasks and the overhead control which can increase the execution time of a given parallel application. In this case, it is important to understand the nature of the application. For embarrassingly parallel, which are featured by little or no dependency or need for communication between the parallel tasks, breaking the application at the UE side and shipping their fragments over different wireless networks to be executed on edge servers will not represent a major obstacle. However, for applications that heavily rely on inter-process communication or applications whose processes exchange a huge volume of data, the cloud network status must be taken into account into the system design. In this case, the joint resource allocation must be overhauled. As presented in Section II, most of the resource allocation schemes optimize the use of wireless resources only, computing resources only, or the jointly wireless and computing resources. However, in order to fulfill the latency requirement for parallel MCC applications running on multiple MEC sites, the resource allocation problem should be engineered taking jointly into consideration the wireless, the computing, and the communications infrastructure between the MEC sites.

Fig. 3 shows that parallel tasks might exchange messages across the X2 interface, which currently enables BSs to coordinate their actions in a LTE deployment. Presently, X2 interface is playing a similar role by coping with mobility management, transference of user's data in case handoff, and inter-cell-interference coordination [26]–[28]. The usefulness of the X2 interface has been subject of investigation in [26] where a ring-based ethernet passive optical network (EPON) architecture is proposed to facilitate its implementation and minimize the handoff latency across it. By using fiber-based backhaul solution rather then the traditional microwave links, higher data rates are achieved which is mandatory in order to maintain the LTE performance when it comes to a lower likelihood of call drops. From a performance evaluation perspective, queueing theoretical models have been proposed to dimension the X2 interface. In [29], Li *et al.* have categorized the data traffic across the X2 interface into two service classes—real time and elastic—and provided the respective models to quantify their performances. Similar to [29], Renard *et al.* in [27] have assumed that the traffic across the X2 is elastic. Considering the use of fiber-based backhaul as underling communication infrastructure and service differentiation at network level, the inter-process communication might be classified into either real-time or elastic traffic based on the latency requirement of the communicating parallel tasks and dispatched through the network to ensure the their proper operation in a timely manner. Thus, employing X2 interface emerges as a convenient solution to manage the inter-process communication.

In this work, parallelism takes place at the application and the physical layers concomitantly. To optimize this procedure, the utilization of cross-layer design techniques arises as a compelling tool for the development of parallelism-aware MCC applications which could benefit from the synergistic operation between MEC and 5G. In this sense, the application should be able to opportunistically detect multiple wireless connections and fragment the code into parallel tasks accordingly taking also into account the edge server's available processing capacity. However, how to optimize the code partitioning tasks considering multiple links simultaneously is still an open issue. The work in [3] provides some design guidelines in how to merge code partitioning and wireless resource allocation. However there is a need to consider the parallelism. From a practical perspective, results in [22]–[24] have paved the way for such degree of parallelism in the sense that they automate the parallelism considering as an input a sequential code and release software engineers from the knowledge about the underlying infrastructure. However, the exploitation of multiple wireless networks simultaneously is disregarded. In order to achieve a joint parallelism as proposed here, this feature must be included in their conceptions. Also, the parallelism across different wireless networks should be transparent to the users only when there is no additional cost for using multiple networks and edge servers. In this case, developers must include this feature into the automation process which should let users know about it, so that they can make a judicious decision whether use multiple infrastructures to enhance their experiences while paying an additional fee. Thus, the confluence of ideas and principles tackled in [3], [22]–[24] might be a starting point for advanced MCC parallel computing.

Last but not least, given the distributed nature of MCC parallel applications, their processes can run into different edge servers while synchronizing by exchanging messages with one another. While the inter-process communication between processes running on MEC servers on different cell sites might not be a major problem since the MEC infrastructure might belong to the same mobile network operator, the inter-process communication between processes running in a MEC server and in a public cloud data center might be a challenge due to factors that range from firewalls which can block the communications to a growing latency caused by the physical distance that sets the virtual edge servers apart from each other. To resolve this problem, SLA must be design considering the presence of these distributed applications.

V. DESIGN EXAMPLE 1: PARALLEL COMPUTATION OFFLOADING

In order to reduce the latency perceived by mobile applications, we design a parallel computation offload method (PCOM) that exploits the parallel transmissions in the HetNet and parallel computing in the MEC to streamline the offloading process. PCOM divides the latency minimization problem into two subproblems: problem P_1 that copes with the data transmissions between the UE and BSs and problem P_2 that deals with the data transmissions and processing between the BSs and the clouds. We formulate P_1 and P_2 as linear optimization problems and use the Simplex algorithm to obtain the optimal solutions.

A. System Assumptions

We consider a HetNet formed by overlapping cells. Each cell hosts its own MEC infrastructure attached to its BS. Additionally, other cloud data centers can be attached into the mobile network operator's infrastructure such as a back-end cloud [7] or a public cloud data center. Regardless of the type of the attached cloud, the purpose is to add extra CPU cycles that can handle the limitation of the MEC, particularly when the peak demand exceeds their capacities.

The set of UEs is denoted by $\mathscr{A} \triangleq \{1, 2, \cdots, A\}$, where A is the total number of UEs. Let $T_{iv} \triangleq (B_i, D_i)$ be the computation task of the type v that will be offloaded by the ith UE in $\mathscr{A}$, where B_i denotes the size of the computation input data and D_i denotes the total number of CPU cycles required by T_{iv} [5]. The rate at which the ith UE generates a task is given by λ_i.

The set of BSs is denoted by $\mathscr{G} \triangleq \{1, 2, \cdots, G\}$, where G is the total number of BSs. The uplink data rate between the ith UE and the jth BS is denoted by $r_{ij}^{\text{ue}\to\text{bs}}$ bps. We assume that once $r_{ij}^{\text{ue}\to\text{bs}}$ is allocated by the BS, it remains constant during the period of the task offloading process. To support this assumption, we consider a setting with no mobility. Given the asymmetry between the uplink and downlink traffic in a MEC setting, we focus on the latency minimization problem in the uplink direction [5].

The set of clouds is denoted by $\mathscr{E} \triangleq \{1, 2, \cdots, E\}$, where E is the total number of clouds. The computational capability of the kth cloud in $\mathscr{E}$, in CPU cycles per second, is given by F_k. Let $r_{jk}^{\text{bs}\to\text{cld}}$ bps be the data rate over the network that connects the jth BS to the kth cloud.

Some important notations are summarized in Table 2.

Table 2. Important notations.

Symbol	Definition
$\mathscr{A}$	Set of UEs
A	Total number of UEs
B_i	Size of the computation input data
$c_{ij}^{\text{ue}\to\text{bs}}$	Transmitting cost between the ith UE and the jth BS
$c_{jk}^{\text{bs}\to\text{cld}}$	Transmitting cost between the jth BS and the kth cloud
D_i	Total number of CPU cycles of T_{iv}
$\mathscr{E}$	Set of clouds
E	Total number of clouds
F_k	Computational capability of the kth cloud
$\mathscr{G}$	Set of BSs
G	Total number of BSs
λ_i	Generation rate of the computation task
$l_{\text{UE}\to\text{BS}}$	Total latency in the air interface
$l_{ij}^{\text{ue}\to\text{bs}}$	Latency perceived by the x_{ij}
$l_{\text{BS}\to\text{CLD}}$	Total latency between the BSs and the clouds
$r_{ij}^{\text{ue}\to\text{bs}}$	Uplink data rate between the ith UE and the jth BS
$r_{jk}^{\text{bs}\to\text{cld}}$	Data rate between the jth BS to the kth cloud
T_{iv}	Computation task of the type v offloaded by the ith UE
w_k	Processing cost of the kth cloud
x_{ij}	Optimal amount of offloaded computation task in P_1
x_{ijk}	Optimal quantity of CPU cycle in P_2

B. Parallel Computation Offload Method (PCOM)

The principle behind the PCOM is to break a computation task into smaller offloadable portions containing the parallel tasks and simultaneously transmit them across the overlapping cells. At the BSs, the optimal offloadable portions may be broken again in order to be delivered to different clouds in a timely manner.

B.1 P_1: Minimizing the Total Latency between the UEs and the BSs

Let $l_{\text{UE}\to\text{BS}}$ be the total latency experienced by transmitting the computation tasks from all the UEs to the BSs. The optimization of $l_{\text{UE}\to\text{BS}}$ is obtained from (1) to (4) as follows:

$$\text{minimize} \quad l_{\text{UE}\to\text{BS}} = \sum_{i\in\mathscr{A}} \sum_{j\in\mathscr{G}} x_{ij} c_{ij}^{\text{ue}\to\text{bs}} \tag{1}$$

$$\text{subject to} \quad \sum_{j\in\mathscr{G}} x_{ij} = B_i,\ i \in \mathscr{A} \tag{2}$$

$$\lambda_i x_{ij} \leq r_{ij}^{\text{ue}\to\text{bs}},\ i \in \mathscr{A}, j \in \mathscr{G} \tag{3}$$

$$x_{ij} \geq 0,\ i \in \mathscr{A}, j \in \mathscr{G}. \tag{4}$$

The quantity x_{ij} in (1) represents the optimal amount of computation task that is offloaded from the ith UE through the jth BS with the transmitting cost $c_{ij}^{\text{ue}\to\text{bs}} = 1/r_{ij}^{\text{ue}\to\text{bs}}$. The constraints in (2) ensure that all the offloadable portions will be transmitted. The constraints in (3) ensure that the amount of offloadable portions received will not exceed the instantaneous uplink data rate while the constraints in (4) ensure that x_{ij} are non-negative.

B.2 P_2: Minimizing the Total Latency between the BSs and the Clouds

The optimization problem addressed in P_2 consists of the transmissions of the optimal offloadable portions from the BSs to the clouds and the processing of their CPU cycles into the clouds. To cope with these two issues in a single formulation, we convert the optimal offloadable portions into their respective CPU cycles and deal with their transmissions from the BSs to the clouds rather than the transmissions of the offloadable portions. Let $l_{\text{BS}\to\text{CLD}}$ be the total latency experienced to transmit the CPU cycles of the optimal offloadable portions that lie on the BSs and process them into the clouds. The problem of how to minimize it by optimally partitioning the CPU cycles among different clouds is defined from (5) to (8) as follows:

$$\text{minimize} \quad l_{\text{BS}\to\text{CLD}} = \sum_{i\in\mathscr{A}} \sum_{j\in\mathscr{G}} \sum_{k\in\mathscr{E}} x_{ijk}(c_{jk}^{\text{bs}\to\text{cld}} + w_k) \tag{5}$$

$$\text{subject to} \quad \sum_{i\in\mathscr{A}} \sum_{k\in\mathscr{E}} x_{ijk} = \frac{x_{ij}}{B_i} D_i,\ x_{i,j} > 0, j \in \mathscr{G} \tag{6}$$

$$\sum_{i\in\mathscr{A}} \sum_{j\in\mathscr{G}} \frac{x_{ijk}}{l_{ij}^{\text{ue}\to\text{bs}}} \leq F_k,\ x_{i,j} > 0, k \in \mathscr{E} \tag{7}$$

$$x_{ijk} \geq 0,\ i \in \mathscr{A}, j \in \mathscr{G}, k \in \mathscr{E}. \tag{8}$$

To solve this problem, we use the outcomes of P_1 as the entries for the P_2. Based on the optimal x_{ij} offloadable quantity, the amount of CPU cycles of the original computation task that x_{ij} carries is $\frac{x_{ij}}{B_i} D_i$. Considering this, the objective function given in (5) determines the minimal latency to transmit and to process the optimal CPU cycle x_{ijk} quantity from the jth BS to the kth cloud where $c_{jk}^{\text{bs}\to\text{cld}} = (B_i/D_i)(1/r_{jk}^{\text{bs}\to\text{cld}})$ is the transmitting cost over the link that connects the jth BS and the kth cloud and $w_k = 1/F_k$ is the processing cost of the kth cloud.

It should be noted that by multiplying x_{ijk} by $c_{jk}^{\text{bs}\to\text{cld}}$, we turn it into the latency perceived by the optimal x_{ij} offloadable quantity due to its transmission over the network that connects the jth BS to the kth cloud. The constraints in (6) ensure that all the CPU cycles will be processed. The constraints in (7) ensure that the instantaneous processing capacity of the kth cloud will not be violated while the constraints in (8) ensure that the x_{ijk} are non-negative. In (7), $l_{ij}^{\text{ue}\to\text{bs}} = x_{ij} c_{ij}^{\text{ue}\to\text{bs}}$ is the latency perceived by the x_{ij} offloadable portion when transmitted from

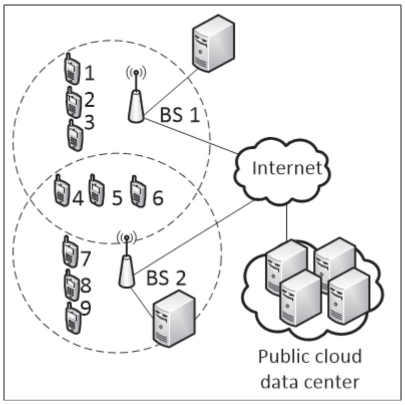

Fig. 4. Scenario under analysis.

the ith UE to the jth BS. This way, $x_{ijk}/l_{ij}^{\text{ue}\rightarrow\text{bs}}$ is expressed in CPU cycles per second.

C. Performance Analysis

Fig. 4 illustrates the scenario under analysis. The HetNet consists of two overlapping cells, each of which is with its own MEC infrastructure and a public cloud data center. The UEs in the set $\{1, 2, \cdots, 8, 9\}$ are distributed over the region under coverage in the following way: UEs in the set $\{1, 2, 3\}$ are covered by the BS $\#1$ only; those in the set $\{4, 5, 6\}$ are covered by both cells, those in the set $\{7, 8, 9\}$ are covered by BS $\#2$ only. Therefore, the HetNet consists of two single-coverage regions and a double-coverage region. We consider that $r_{ij}^{\text{ue}\rightarrow\text{bs}} \approx 10$ Mbps ($i \in \mathscr{A}, j \in \mathscr{G}$), which correspond to the small cell performance reported by Huawei [30]. Additionally, a face recognition application is assumed, so that $T_{ij} =$(420 KB, 1,000 Megacycles) [5]. We are particularly interested in how UEs in the double coverage area will exploit the parallelism as the workload increases. This way, the UEs generate task offloading requests (λ_i) according the following settings: low traffic load (S1) profile {1, 1, 1, 1, 1, 1, 1, 1, 1} task offloading requests per second, medium traffic load (S2) profile {1, 1, 1, 3, 3, 3, 1, 1, 1} task offloading requests per second, and high traffic load (S3) profile {1, 1, 1, 5, 5, 5, 1, 1, 1} task offloading requests per second. Note that the UEs in the single coverage region have the same QoS profile $\lambda_i B_i = 420$ KBps ≈ 3.36 Mbps while for those in the double coverage area, the QoS profile changes according to $\{3.36, 10.08, 16.6\}$ Mbps for S1, S2, and S3 settings, respectively. Thus, for the given $r_{ij}^{\text{ue}\rightarrow\text{bs}}$, only the setting S1 will be instantaneously fulfilled. The computation capabilities of the clouds are $F_1 = F_2 = 100$ GHz and $F_3 = 500$ GHz. The data rates in the wired networks are $r_{11}^{\text{bs}\rightarrow\text{cld}} = r_{22}^{\text{bs}\rightarrow\text{cld}} = 1$ Gbps and $r_{13}^{\text{bs}\rightarrow\text{cld}} = r_{23}^{\text{bs}\rightarrow\text{cld}} = 100$ Mbps.

Due to the lack of a benchmark, we design a greedy algorithm and use it as the baseline for our analysis. Because of the greediness feature, the UEs are always associated with the small cell with the lowest transmitting cost and its MEC virtualized servers. If the local servers cannot support the computation task, then the remote cloud will be chosen. Under the greedy policy, when there is a match among small cells or clouds, the algorithm will randomly select one of these matches. No parallel transmission and processing are supported by the greedy algorithm.

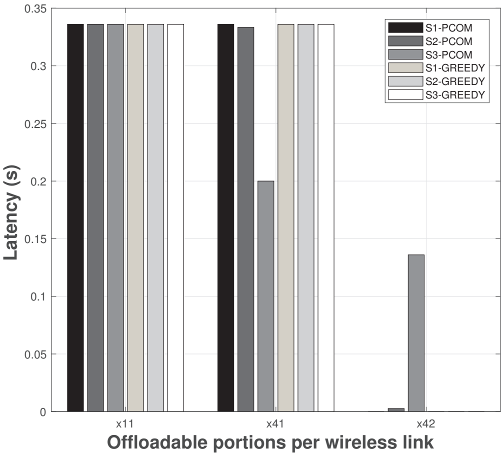

Fig. 5. Solution of P_1 for the scenario under analysis: Latency (s) versus offloadable portions per wireless link.

Fig. 5 presents an instantaneous snapshot of the system showing how the computation task of each UE is transmitted over each small cell for the three traffic load settings. Since all the UEs in the single coverage area have the same performance and the UEs in the double coverage also have the same performance, we summarize the results by considering the UE $\#1$ and $\#4$.

In the single coverage area, PCOM and greedy algorithm have similar performance for all settings since there is no option for parallelism. The same applies for the S1 setting in which in both PCOM and the greedy algorithm are able to instantaneously accommodate the QoS profile $\lambda_i B_i$ within the given uplink data rate $r_{ij}^{\text{ue}\rightarrow\text{bs}}$. However, as the workload increases, the PCOM starts to split the computation tasks into smaller optimal offloadable portions in order to streamline the data transmissions over the HetNet. For instance, for the S2 and the S3 settings, it can be observed that the PCOM breaks the computation task placed by the 4th UE into two offloadable portions: x_{41} and x_{42}. Considering S2, it is observed that the BS $\#1$ supports 10 Mbps for each UE in the double coverage area while the BS $\#2$ deals with 0.08 Mbps. For S3, in which the parallelism is more evident, the BS $\#1$ supports 10 Mbps for each UE in the double coverage area while the BS $\#2$ supports 6.8 Mbps. By doing this, PCOM successfully deals with the limitations in the instantaneous uplink data rate to support the entire demand simultaneously.

Nevertheless, the greedy algorithm fails in dealing with all the workload concurrently. In such a case, the computation task or the part of it that could not be immediately offloaded can either be executed locally which will increase the energy expenditure of the UE or may be buffered and wait until the transmission of the actual workload is ended which will increase the delay.

Fig. 6 illustrates how the offloadable portions which lie on the BSs have their CPU cycles transmitted to and processed into the MEC servers. As shown, the PCOM method does not break the optimal offloadable portions which were obtained in the solution of P_1 into new ones, but similarly to the greedy algorithm, it directly transmits them to MEC servers. This happens because the MEC servers, although being resource-constrained

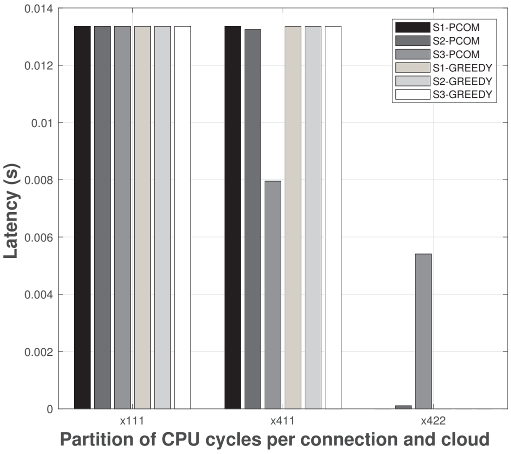

Fig. 6. Solution of P_2 for the scenario under analysis: Latency (s) versus partition of CPU cycles per wired connection and cloud data center.

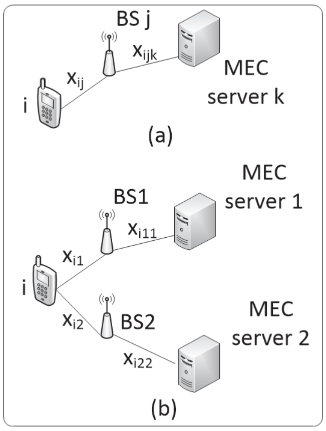

Fig. 7. Optimal task offloading policy in a double coverage region: (a) Low traffic (b) medium to high traffic load.

when compared to the public cloud, are still resourceful in face of the demand. Furthermore, the method avoids transmitting data through a higher cost wired connection. Fig. 6 also shows that the optimal offloadable portions x_{42} which is the solution of the problem P_1 is transmitted to its MEC server x_{422}.

Finally, Fig. 7 presents the optimal computation offloading policy for latency minimization for the setting under analysis. Fig. 7(a) indicates that whenever possible, the computation task will be transmitted over a single cell, i.e., while the uplink data rate is enough to cope with the QoS profile. However, Fig. 7(b) shows that as the traffic load increases and the uplink data rate is not able to instantaneously meet the QoS profile, then PCOM fragments the computation tasks into smaller optimal offloadable portions and ships them over multiple BSs simultaneously. From the BSs onwards, the method does not break the optimal offloadable portions into new ones, but transmits them to the closer cloud computing center as long as it can support them. Thus, the proposed method exploits the parallel transmission and the parallel processing when a medium to heavier workload is sensed.

Table 3. Important notations.

Symbol	Definition
$\mathbb{A}$	Set of arcs
c_{ij}	Transmitting cost in the arc $(i,j) \in \mathbb{A}$
$\mathbb{D}$	Set of demand nodes
$l_{\mathrm{BS}\to\mathrm{CLD}}$	Total latency needed to offload the mobile data chunks
λ_i	Generation rate of files
$\mathbb{N}$	Set of all nodes
O_k	Available storage capacity in Bytes of the kth cloud
r_{ij}	Data rate in the arc $(i,j) \in \mathbb{A}$
$\mathbb{S}$	Set of supply nodes
$\mathbb{T}$	Set of intermediate nodes
Y_i	Size of the file
x_{ij}	Optimal size of mobile data chunk in the arc $(i,j) \in \mathbb{A}$

VI. DESIGN EXAMPLE 2: PARALLEL TRANSMISSION AND STORAGE FOR EFFICIENT UBIQUITOUS ACCESS

Currently, digital media such as videos and photos are generated at unprecedented levels by mobile users equipped with powerful UEs. Given the limitations for local storage, cloud-based storage emerged as an attractive solution to accommodate all the mobile data while ensuring a worldwide Internet-based access to it as well as the application of big data analytics techniques.

In a setting considering MEC over 5G, storage-based applications may rely on parallelism in order to enhance reliability and robustness of the data transferring and storage. From the transmission viewpoint, UEs can take advantage of the overlapping nature of HetNets and efficiently and effectively upload their data over multiple wireless networks simultaneously if possible. At the BSs, the data chunks can be fragmented again into smaller chunks or follow directly to a single cloud to be stored in a timely fashion.

In order to minimize the latency in this setting, optimization techniques may be applied to find the best transmission and storage strategy that would lead to a minimum latency. In [4] it was suggested the application of network optimization techniques to address this open issue, but no formal solution was putting forward. As follows, we present the PTSM which is based on network optimization model and has as the objective to find the best path in the network which results in the minimum latency.

A. System and Model Assumptions

We consider that the size of the data, which will be offloaded by the ith UE with rate λ_i file/s, is Y_i bytes. Let O_k be the available storage capacity in Bytes of the kth cloud when an UE makes a storage request. We assume that during the data upload, users have no mobility as well as the number of UE remains unchanged [5]. Consequently, the upload data rate will remain constant during the data offloading process.

B. Parallel Transmission and Storage Method (PTSM)

We assume that UEs, the BSs, and clouds are all interconnected in such a way that we can formulate the latency minimization problem for ubiquitous storage for mobile data as a network optimization model. Thus, UEs belong to the set of *supply nodes* $\mathbb{S}$, which represents the entry of the mobile data into the network and the clouds belong to the set of *demand nodes*

$\mathbb{D}$ from which the mobile data leave the network. The set of *intermediate nodes* is defined by $\mathbb{T}$ and contains the BSs. At these nodes, the mobile data could be assembled or divided before being forwarded. The set of all nodes is computed as $\mathbb{N} = \mathbb{S} \cup \mathbb{T} \cup \mathbb{D}$ and the set of arcs is $\mathbb{A} \subseteq \mathbb{N} \times \mathbb{N}$ [1]. Considering $\mathbb{A}$, we are interested in the flow x_{ij} along the arc $(i, j) \in \mathbb{A}$ that specifies the size of the mobile data chunk. Thus, the network optimization problem becomes to

$$\text{minimize} \qquad l_{\mathrm{BS}\rightarrow\mathrm{CLD}} = \sum_{(i,j)\in\mathbb{A}} x_{ij} c_{ij} \tag{9}$$

subject to

$$\sum_{(i,j)\in\mathbb{A}} x_{ij} = Y_i, \ i \in \mathbb{S}, j \in \mathbb{T} \tag{10}$$

$$\lambda_i x_{ij} \leq r_{ij}, \ i \in \mathbb{S}, j \in \mathbb{T} \tag{11}$$

$$\sum_{(i,j)\in\mathbb{A}} x_{ij} - \sum_{(j,k)\in\mathbb{A}} x_{jk} = 0, \ i \in \mathbb{S}, j \in \mathbb{T}, k \in \mathbb{D} \tag{12}$$

$$\sum_{(i,j)\in\mathbb{A}} x_{ij} \leq O_j, \ i \in \mathbb{T}, j \in \mathbb{D} \tag{13}$$

$$x_{ij} \geq 0, \ i, j \in \mathbb{A}. \tag{14}$$

The objective function in (9) is to minimize the total latency $l_{\mathrm{BS}\rightarrow\mathrm{CLD}}$ need to offload the mobile data chunks, where $c_{ij} = 1/r_{ij}$ is the transmitting cost and r_{ij} is the data rate in the arc $(i, j) \in \mathbb{A}$. The constraint in (10) implies that the flow out the supply nodes is equal to the data produced which means that all media produced will be stored into the clouds, which is suitable for thin clients. The constraint in (11) ensures that the instantaneous uplink capacity of the mobile user will not be violated by the offloading data chunks. The constraint in (12) guarantees that the flow is conserved at the *intermediate* nodes while the one in (13) ensures the clouds will not store more than their capacities. Finally, (14) specifies that $x_{i,j} \in \mathbb{A}$ are non-negative. The above optimization problem is a linear program (LP) which can be solved when the system parameters are known.

Table 3 summarizes some important notations used in the formulation of PTSM.

C. Performance Analysis

Fig. 8(a) shows the network design for the system under analysis while Fig. 8(b) shows its abstraction in terms of a network optimization problem. It is important to mention that the back-end cloud is connected to the BSs through the core network as shown in Fig. 2. For simplicity, we represent a direct connection between the BSs and the back-end cloud which in practice could be interpreted as a logical connection. For Fig. 8(b), we consider again that $r_{12} = r_{13} \approx 10$ Mbps. We also assume $r_{24} = r_{36} =$ 1 Gbps and $r_{25} = r_{35} = 100$ Mbps what is the data rates of commodities data communication systems.

Three settings are evaluated. In the first setting (S1), the cloud storage traffic $\lambda_1 Y_1 = 1.6$ Gigabytes per month [31], which results in $\lambda_1 Y_1 \approx 5$ kbps. Considering $Y_1 = 1$ MB, then $\lambda_i \approx$ 0.0006 files/s. For S1, there are lots of free space left to accommodate the incoming requests and the cloud storage availability is $O_4 = O_6 = 8$ GB and $O_5 = 100$ GB. For the second setting (S2), $\lambda_1 Y_1 \approx 14$ Mbps. For $Y_1 = 1.75$ MB, then $\lambda_1 \approx 1$ file/s. For this scenario, the MEC servers are running out of space and the cloud storage availability is $O_4 = 1$ MB, $O_6 = 0$ MB, and $O_5 = 100$ GB. The third setting (S3) resembles S2, but in this case $O_4 = O_6 = 0$ MB.

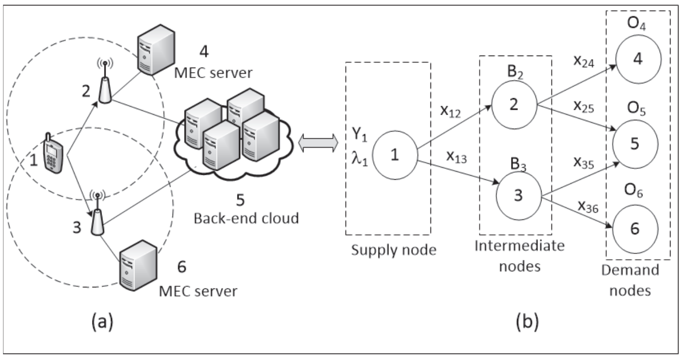

Fig. 8. Scenario under analysis: (a) Network architecture and (b) PTSM as a network optimization problem.

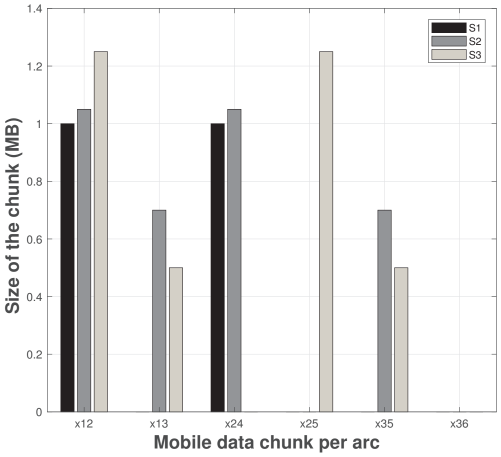

Fig. 9. The sizes of data chunk (in MB) transmitted over different network paths in different network settings.

Fig. 9 shows how the mobile storage data traffic is fragmented over the nodes in the network depicted in Fig. 8(b). As can be observed, for the S1 setting, which is featured by a low cloud storage traffic and abundance of storage space, the UE does not fragment its mobile data traffic into smaller chunks to exploit the parallelism over the HetNet. Thus, all the data is uploaded through the BS #2 – x_{12} – to its MEC server (node #4) – x_{24}. However, as the cloud storage traffic surpasses the instantaneous uplink data rate and the available storage capacity diminishes, the UE starts to transmit simultaneously over both small cells and to deposit the data in the back-end cloud. Thus, for the S2 setting both x_{12} and x_{13} are activate. Since there is no space left on the node #6, i.e., $O_6 = 0$, the data chunks that uploaded through the BS #3 are forwarded directly to the back-end cloud (node #5), i.e., $x_{13} = x_{35} > 0$. For S3, MEC servers are un-

[1] Since PTSM is methodologically and conceptually different from PCOM, we have intentionally used different notation to express the sets of UEs, BSs, and clouds which is more meaningful within the context of the current problem and the applied approach.

able to cope with the income requests, because of that the cloud storage traffic is forwarded to the back-end clouds—$x_{25} > 0$ and $x_{35} > 0$ – simultaneously through both BSs. For the three settings, the total latency is 0.80 s, 1.46 s, and 1.54 s for S1, S2, and S3, respectively.

It is worth mentioning that despite their differences, PTSM and PCOM reach similar conclusion, they both rely on parallelism in wireless and cloud domain to streamline the transmission and computation processes when the system is running out of capacity. Thus, the parallelism exploitation appears as a compelling solution for the problem of resource optimization.

VII. DISCUSSION ON PCOM AND PTSM

For PCOM and PTSM, the linear optimization problems were implemented and solved using the AMPL/CPLEX solver running in the NEOS Server, a free internet-based service for solving numerical optimization problems [32]. As it is well-known, linear programming problems are solvable in weakly polynomial-time. In order to get the optimal solution, we apply the Simplex algorithm, which despite getting the exponential worst case complexity, performs well in practice. For larger instances, the Interior Point algorithm can be applied to get the optimal solution.

It is implicit assumed that PCOM and PTSM have the knowledge of the whole set of parameters in advance to perform their activities. To this end, both rely on the cooperation and/or collaboration between the resource managers in both domains—wireless networks and cloud networks—in order to intelligently exploit the common pool of wireless resources in a HetNet while optimizing the CPU cycles allocation in a setting with multiple cloud data centers to streamline the computation offloading process or optimizing the use of the available storage capacity. In HetNets, the resource allocator is known as common radio resource management server (CRMS) while in a multi-cloud setting it is the cloud service broker (CSB). In [4], it is presented how the inter-operation between CRMS and CSB can be leveraged in order to make the systems integration effective as well as the challenges and open issues behind it.

Furthermore, PCOM and PTSM assume a semi-static scenario. In this case, users are static during the offloading period; therefore, they sense no change in their wireless conditions. In addition to affording tractability and actionable insights, this assumption is justified by the fact that the offloading process may be finished in a time scale that is shorter than the one of the user's mobility [5].

When it comes to the type of application, PCOM is engineered to operate with embarrassingly parallel applications. Due to the loose dependence between the tasks, code partition is noncritical for these applications. However, for non-embarrassingly parallel applications, new constraints reflecting the time-dependence among multiple parallel tasks must be added to enforce their synchronization.

It is paramount to emphasize that for computation-hungry task, parallelism is critical since there is a need to tight the code partitioning with the availability of radio resources in multiples wireless networks and multiple clouds. On the other hand, for storage-hungry applications, parallelism is more straightforward since the segmentation process of the media is less critical due to the lack of inter-dependence and communication among the individual pieces.

VIII. CONCLUSION

This paper raised an ambitious question: *Is it possible to jointly exploit the parallelism at wireless network and cloud computing to support advanced MCC applications?* From the evidences that paper has presented, we believe that the answer is *yes*. There are individual initiatives taking place in both arenas stating that it is viable to take advantage of the synergistic among MEC and 5G to streamline advanced parallel mobile cloud applications. To illustrate the benefits of parallelism, we have presented PCOM and PTSM as representatives design examples for the problem of computation offload and mobile data storage, respectively. Their conclusions emphasize that fact that wireless and cloud resources can be better use and applications can be better supported when parallelism is jointly employed at application and network levels.

Currently, our research efforts are particularly focused on specifying the design of cross-layer MCC applications that could optimize the application partitioning taking into account multiple wireless and cloud data centers. Also, we have worked towards the qualification of potential scenarios and applications that can make fully use of the ideas presented in this paper.

Finally, our expectation towards the ideas and principles outlined in this paper is not to exhaust the discussion about parallelism, but open it up for both communities—wireless and cloud scientists and practitioners—in the sense that given the interdisciplinary aspect of the topic, a collaborative and holistic approach is the best manner to come up with practical and theoretical solutions for it.

REFERENCES

[1] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 38–43, Mar. 2017.

[2] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.

[3] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[4] G. H. S. Carvalho, I. Woungang, A. Anpalagan, M. Jaseemuddin, and E. Hossain, "Intercloud and HetNet for mobile cloud computing in 5G systems: Design issues, challenges, and optimization," *IEEE Network*, vol. 31, no. 3, pp. 80-89, May/June 2017.

[5] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[6] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, "An SMDP-based service model for interdomain resource allocation in mobile cloud networks," *IEEE Trans. Veh. Tech.*, vol. 61, no. 5, pp. 2222–2232, June 2012.

[7] L.Gkatzikis and I. Koutsopoulos, "Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp.24–32, June 2013.

[8] M. Felemban, S. Basalamah, and A. Ghafoor, "A distributed cloud architecture for mobile multimedia services," *IEEE Network* vol. 27, no. 5, pp. 20–27, Sept.–Oct. 2013.

[9] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[10] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81–93, Jan. 2015.

[11] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 34–44, June 2013.

[12] Y. Choi, H. Kim, S. Han, and Y. Han, "Joint resource allocation for parallel multi-radio access in heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3324–3329, Nov. 2010.

[13] J. Miao, Z. Hu, C. Wang, R. Lian, and H. Tian, "Optimal resource allocation for multi-access in heterogeneous wireless networks," in *Proc. IEEE VTC*, 2012, pp. 1–5.

[14] Y. Song, Y. Han, and Y. Choi, "Radio resource management based on QoE-aware model for uplink multi-radio access in heterogeneous networks," in *Proc. IEEE VTC*, 2014, pp. 1–5.

[15] G. Lim, C. Xiong, L. J. Cimini, and G. Y. Li, "Energy-efficient resource allocation for OFDMA-based multi-RAT networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2696–2705, May 2014.

[16] Y. Nakai, D. Perrin, H. Ohsaki, and R. Walshe, "Performance evaluation of cloud-based parallel computing," in *Proc. IEEE COMPSAC*, 2013, pp. 351–355.

[17] C. Freniere, A. Pathak, M. Raessi, and G. Khanna, "The feasibility of amazon's cloud computing platform for parallel, GPU-accelerated, multiphase-flow simulations," *Comput. Sci. & Eng.*, vol. 18, no. 5, pp. 68–77, Sept.–Oct. 2016.

[18] I. Sadooghi *et al.*, "Understanding the performance and potential of cloud computing for scientific applications," *IEEE Trans. Cloud Comput.*, vol. 5, no. 2, pp. 358–371, Apr.–June 2017.

[19] R. R. Righi *et al.*, "AutoElastic: Automatic resource elasticity for high performance applications in the cloud," *IEEE Trans. Cloud Comput.*, vol. 4, no. 1, pp. 6–19, Jan.-Mar. 2016.

[20] D'Angelo and Rampone, "Towards a HPC-oriented parallel implementation of a learning algorithm for bioinformatics applications," *BMC Bioinformatics*, vol. 15 (Suppl 5):S2, pp. 1–15, 2014.

[21] R. M. Badia *et al.*, "COMP superscalar, an interoperable programming framework," *Elsevier SoftwareX*, vol. 3–4, pp. 32–36, Dec. 2015.

[22] F. Lordan *et al.*, "ServiceSs: An interoperable programming framework for the cloud," *J. Grid Comput.*, vol. 12, no. 1, pp. 67–91, Mar. 2014.

[23] F. Lordan and R. M. Badia, "COMPSs-mobile: Parallel programming for mobile-cloud computing," in *Proc. IEEE/ACM CCGrid*, 2016, pp. 497–500.

[24] R. Lin, H. Tu, and H. Zou, "Parallel computing framework as a cloud service," in *Proc. IEEE ICWS*, pp. 672–673, 2012.

[25] Y. Sharma, B. Javadi, W. Si, and D. Sun, "Reliability and energy efficiency in cloud computing systems: Survey and taxonomy" *J. Network Comput. Applicat.*, vol. 74, pp. 66–85, Oct. 2016.

[26] A. Mukhopadhyay and G. Das, "A ring-based wireless optical network to reduce the handover latency," *J. Lightwave Technology*, vol. 33, no. 17, pp. 3687–3697, Sept. 2015.

[27] B. Renard, S. E. Elayoubi, and A. Simonian, "A dimensioning method for the LTE X2 interface," in *Proc. IEEE WCNC*, 2012, pp. 2718–2723.

[28] A. Adeel, H. Larijani, and A. Ahmadinia, "Resource management and inter-cell-interference coordination in LTE uplink system using random neural network and optimization," *IEEE Access*, vol. 3, no. 3, pp. 1963–1979, 2015.

[29] X. Li *et al.*, "Dimensioning of the LTE access network," *Telecommun. Syst.*, vol. 52, no. 4, pp. 2637–2654, 2013.

[30] S. Agarwal, M. Philipose, and V. Bahl, "Vision: The case for cellular small cells for cloudlets," in *Proc. ACM MobiSys*, June 2014.

[31] Cisco Global Cloud Index: Forecast and Methodology, 2014-2019 White Paper.

[32] [Online]. Available: http://www.neos-guide.org

Glaucio H.S. Carvalho is a Professor of the School of Applied Computing, Faculty of Applied Science and Technology (FAST) at Sheridan College. He received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the Federal University of Para (UFPA), Brazil, in 1999, 2001, and 2005, respectively. He worked as a Professor at UFPA from 2005 to 2015, Department of Computer Science where he served as a Chair of the Faculty of Information Systems (2005–07) and Vice-Chair of the Graduate Program in Computer Science (2010–12). From 2010 to 2015 he served as an Associate Editor of the *Computers and Electrical Engineering (CAEE)-Elsevier* where he was a Top Associate Editor in 2011. He was a Guest Editor for the CAEE special issue on the *Design and Analysis of Wireless Systems: New Inspirations.* He worked as a Postdoctoral Fellow (PDF) and Instructor at Ryerson University, Department of Computer Science and served as the Chair of the IEEE Toronto Section Signals & Computational Intelligence Joint Society (2016). Dr. Carvalho's research interests include security and performance analysis of cloud systems and distributed systems.

Isaac Woungang received his Ph.D. degree in Mathematics from Universite du South, Toulon & Var, France, in 1994. From 1999 to 2002, he worked as Software Engineer at Nortel Networks, Ottawa, Canada. Since 2002, he has been with Ryerson University, where he is now a Professor of Computer Science & Director of the DABNEL Lab. His current research interests include radio resource management in wireless networks, cloud security, and routing in opportunistic networks. He has published 8 edited & 1 authored books, and over 80 refereed journals and conference papers. He serves as Editor in Chief of the International Journal of Communication Networks and Distributed Systems (IJCNDS), Inderscience, UK, and has served as Chair of the Computer Chapter, IEEE Toronto Section.

Alagan Anpalagan is a Professor in the Department of Electrical and Computer Engineering at Ryerson University where he directs a research group working on radio resource management (RRM) and radio access and networking (RAN) areas within the WINCORE Lab. Dr. Anpalagan served as Editor for the IEEE Communications Surveys and Tutorials (2012-14), IEEE Communications Letters (2010-13), Springer Wireless Personal Communications (2011–13), and EURASIP Journal of Wireless Communications and Networking (2004–2009). He co-authored three edited books, Design and Deployment of Small Cell Networks, Cambridge University Press (2014), Routing in Opportunistic Networks, Springer (2013), Handbook on Green Information and Communication Systems, Academic Press (2012).

Dr. Anpalagan currently serves as TPC Vice-Chair, IEEE VTC Fall-2017, and served as TPC Co-Chair, IEEE GLOBECOM'15: SAC Green Communication and Computing, IEEE WPMC'12 Wireless Networks, IEEE PIMRC'11 Cognitive Radio and Spectrum Management, and IEEE CCECE'04/08. He served as Vice Chair, IEEE SIG on Green and Sustainable Networking and Computing with Cognition and Cooperation (2015–), IEEE Canada Central Area Chair (2012–14), IEEE Toronto Section Chair (2006–07), ComSoc Toronto Chapter Chair (2004–05), and IEEE Canada Professional Activities Committee Chair (2009–11). He is the recipient of the Dean's Teaching Award (2011), Faculty Scholastic, Research and Creativity Award (2010, 2014, 2017), Faculty Service Award (2011, 2013) in Ryerson University. He also received Exemplary Editor Award from IEEE ComSoc (2013) and Editor-in-Chief Top10 Choice Award in Transactions on Emerging Telecommunications Technology (2012). He received BASc., MASc. and Ph.D. degrees in Electrical Engineering from the University of Toronto. He is a registered Professional Engineer in the province of Ontario, Canada, Senior Member of IEEE and Fellow of the Institution of Engineering and Technology.

Muhammad Jaseemuddin received B.E. from N.E.D. University, Pakistan, M. S. from The University of Texas at Arlington, and Ph.D. from University of Toronto. He worked in Advanced IP group and Wireless Technology Lab (WTL) at Nortel Networks. He is Associate Professor at Ryerson University. His research interests include caching in 5G network; context-aware mobile middleware and mobile cloud; localization, power-aware MAC and routing for sensor networks; impact of mobility on routing and transport layers; heterogeneous wireless networks; and IP routing and traffic engineering.