This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing

Computing-Aware Base Station Sleeping Mechanism in H-CRAN-Cloud-Edge Networks

Ali Alnoman and Alagan Anpalagan

Abstract—In this paper, a power minimization problem using base station sleeping is proposed for heterogeneous cloud radio access networks (H-CRANs) taking into account the computing delay constraints. In the proposed system, which is modeled using M/M/k queues, the edge device coexists with the small base station (SBS) to provide computing capabilities beside the central cloud. In general, the SBS sleeping is governed by the availability of resources provided the macro base station (MBS) which is in charge of accommodating offloaded users from sleeping SBSs. However, switching off lightly loaded SBSs can impose significant burdens on cloud servers. Here, the proposed sleeping scheme allows SBSs serving more computing tasks to remain active in order to fulfill the task completion deadlines requested by mobile users and to keep the cloud response time within a predefined limit. In other words, the proposed scheme aims to save power by undertaking a centralized selection of active and sleeping SBSs taking into account the delay constraints of both cloud and mobile devices. First, we consider a disjoint cloud-edge system, where computing services can be provided by either the cloud or the edge device, and aim to minimize the number of active SBSs. The problem is formulated as a 0-1 knapsack problem with SBS utilization considered as the weight while the ratio of computing tasks to all incoming tasks is considered as the value of that SBS. In this problem, which is solved using dynamic programming, SBSs processing less computing tasks are given higher values; and as a result, higher chance to sleep compared to others. Secondly, a shared computing system is proposed whereby active SBSs (edge devices) contribute to the total computing capability. Here, an exhaustive search approach is used to achieve the optimal power saving. We also proved that the shared computing system performs better in terms of response time compared to the disjoint system depending on the number of active SBSs.

Index Terms—H-CRAN, cloud-edge computing, energy, response time, M/M/k.

1 Introduction

Future cellular networks are characterized by their capability to satisfy the stringent needs of mobile users in regard with latency and data rate [1]. Due to the vast diversity of radio access technologies (RATs) deployed in heterogeneous networks (HetNets), the management of such networks is becoming more complicated and challenging. To this end, performing data aggregation from all network nodes in the centralized baseband unit (BBU) pool for processing, in the well-known architecture of heterogeneous cloud radio access networks (H-CRANs), can achieve huge success in this direction [2]. The remote radio heads (RRHs) and small base stations (SBSs) in H-CRANs are basically deployed to provide high data rates by exploiting the spatial reuse of frequencies. Meanwhile, macro base stations (MBSs) are in charge of providing cross-tier management such as user association, handover management, traffic flow, and network-wide coverage. In other words, SBSs belong to the data plane whereas MBSs belong to the control plane.

From the computing perspective, having the complex computing tasks such as computer vision and data analytics processed in the central cloud is a big step towards improving the computing performance for users and machines [3]. Nevertheless, the ever increasing number of connected devices in the context of Internet of Things, smart homes, autonomous driving, etc., will eventually overload or even crash cloud servers. Thus, it is essential to filtrate data to reduce the burden on the cloud and network resources, and to improve the quality of experience (QoE) especially in regard with end-to-end delay [4] [5].

Bringing computing services at the vicinity of mobile users in the paradigm of edge (fog) computing can significantly reduce the end-to-end delay experienced by users. This reduced delay helps support the emerging delay-sensitive applications such as E-health, real-time control, and vehicular communications [6] that can tolerate a delay of only few milliseconds [7]. Edge devices are equipped with the necessary hardware to enable small-scale cloud-like functions such as computing and storage. Moreover, edge computing benefits the close proximity with mobile users to offer geo- and context-aware services such as content caching. It is thus obvious why edge computing which complements the cloud is described as ”fog” because fog physically resides closer to the ground (users) compared to the ”cloud” seen in the sky [8]. To take full advantage of edge computing, it is necessary to coordinate edge devices with the central cloud on one hand, and with the H-CRAN on the other hand [9]. With the help of software-defined networking (SDN) technology, efficient coordination of computing and communication nodes can be achieved with less complexity.

One of the main constraints that stands in the way of future networks is the high energy consumption. Not only because energy raises the operational expenditures, but also because it causes detrimental impacts on our planet. Adopting smart SBS operation mechanisms can significantly reduce energy consumption since base stations account for 80% of the overall energy consumption in cellular networks [10]. Controlling the operation of SBSs can be achieved in a distributed or centralized manner. In the former, an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing

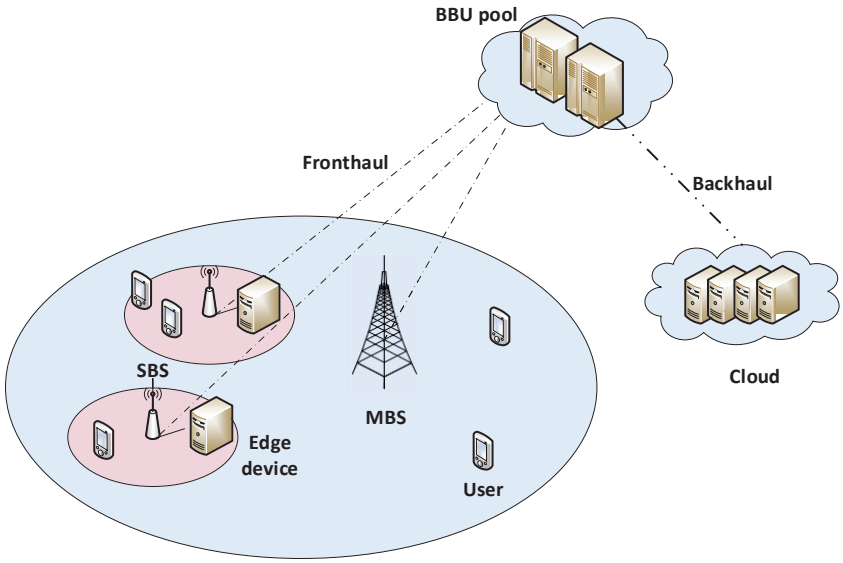


Fig. 1: H-CRAN-CE system layout.

SBS operates as a stand alone entity using intelligent self-organized features. Whereas in the centralized control, data from SBSs, MBSs, and other supporting nodes enter the central BBU pool for an optimal network-aware processing. The overhead of the centralized control is naturally higher compared to the distributed one; however, the informed and ceratin decisions of the centralized control boosts the overall system performance. Therefore, prior to initiating the On/Off and traffic offloading processes, network nodes should be well coordinated to maintain high QoS [11].

Similar to traffic offloading in cellular networks, computing tasks can also be offloaded from edge devices to the cloud and vice versa depending on the desired QoS requirements such as energy and delay [12]. In other words, computing tasks can be processed either locally by the edge device or remotely by the cloud via the MBS through backhaul links [13]. However, offloading tasks to the central cloud will inherently increase the burden on cloud servers, communication resources, and backhaul links. Moreover, adopting coordinated task offloading in the layered cloud-fog architecture can increase the communication overhead and thus extra delay [14]. Therefore, it is essential to take into account the consequences of task offloading on both the communication and computing nodes. From the aforementioned, we propose a coordinated cellular-computing architecture that considers both communication and computing resources towards optimal SBS sleeping operation. Fig. 1 depicts the state-of-the-art H-CRAN-cloud-edge system.

The organization of this paper is as follows. Section 1 provides an overview, related work, and the main contributions of this work. Section 2 describes the power, network, and computing models. The computing-aware SBS sleeping scheme is introduced in Section 3, followed by SBS sleeping in the proposed shared computing model in Section 4. In Section 5, simulation setup and results are demonstrated, and finally, Section 6 provides concluding remarks.

1.1 Related Work

Over the last few years, SBS sleeping gained considerable attention in the context of HetNets. Nevertheless, limited amount of research considered SBS sleeping from both communication and computing perspectives. In [15], a sleeping strategy was proposed by which all RATs are activated when resource utilization in the MBS reaches a threshold value. The N-policy scheme in [10] is concerned with the energy-delay tradeoff in SBS sleeping without considering traffic offloaded from sleeping SBSs to the MBS. Furthermore, the SBS activation delay was the goal of [16], wherein authors used iterative approaches to maximize energy efficiency considering wake-up times and coverage probability regardless of the MBS traffic load. All aforementioned works were considering performance in a communication environment; that is to say, no computing aspects were involved.

However, the proliferation of computing hungry applications have brought the attention of both academic and industrial communities recently. For instance, a hierarchical edge-cloud architecture was proposed in [17] to achieve workload balancing among different computing tiers. By dynamically distributing the workload on different servers, over 25% improvement in program execution time was obtained. In a similar context, authors in [8] considered workload scheduling to find the optimal power-delay tradeoff in cloud-fog computing systems. Furthermore, a scheduling algorithm was proposed in [18] to minimize the queue delay in cloud servers in order to guarantee the ultra-low latency in Internet services.

Since communication nodes play a major role in linking computing tasks with computing infrastructure, it is essential to consider both communication and computing nodes in contemporary research work. Here, a joint energy harvesting and SBS sleeping was studied in [19] aiming at minimizing energy consumption and improving the caching performance in cache-enabled SBS networks. The work con-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing

sidered the effect of SBS sleeping while maximizing the hit ratio of cached contents.

Unlike most related work, we aim to maximize power saving considering the SBS load, MBS load, cloud response time, delay experienced by users, and traffic offloaded from sleeping SBSs to the MBS. The joint operation of both communication and computing nodes can improve the network-wide performance and provide sophisticated sleeping mechanism for future networks. Table 1 compares this work with related ones in the literature.

1.2 Contribution

The contribution of this work is three-fold:

- A SBS sleeping mechanism is proposed to save energy in integrated H-CRAN-cloud-edge networks under the constraints of cloud response time and task completion deadline. In other words, two types of constraints are considered namely the long-term statistical cloud response time, and the instantaneous task completion time. In this part of the work, the cloud and edge servers are assumed to have disjoint operation; that is, the workload cannot be shared (disjoint queue model). The problem is formulated as a 0-1 knapsack problem wherein the SBS utilization represents the weight whereas the amount of incoming computing tasks represents the value of that SBS. Here, SBSs serving less amount of computing tasks are given higher values than others. The proposed problem, which is solved using dynamic programming, is a centralized SBS sleeping scheme that aims to select the optimal subset of sleeping SBSs considering cloud and user constraints.
- A novel shared cloud-edge computing architecture is introduced in coordination with the cellular infrastructure. Here, edge and cloud servers are integrated in a unified queue system i.e. one queue and shared servers. Thereby, edge devices contribute to the improvement of the computing response time by increasing the total number of functioning servers.
- The optimal subset of sleeping SBSs is then found in the later system using exhaustive search approach. Again, the computing response time and task completion deadline are considered as constraints in this problem.

2 System Model

In this section, power, network, and computing models are presented. The general view of the integrated H-CRAN-cloud-edge system can be well perceived in Fig. 1.

2.1 Power Model

The MBS is assumed to remain active all the time in order to provide coverage, cross-tier control, and to accommodate users offloaded from sleeping SBSs. Accordingly, the MBS has approximately a constant power and thus does not affect the SBS sleeping performance. For this reason, the MBS is not taken into account when calculating the total network power. The SBSs, on the other hand, coexist with the MBS and carry out a flexible On/Off operation. The power consumption of the jth SBS is given by

$$P_j = \begin{cases} P_s, & \text{if SBS is On} \\ 0, & \text{if SBS is Off} \end{cases} \tag{1}$$

where P_s denotes the power consumption during the active mode. Hence, the power consumed by all active SBSs can be expressed as

$$P_t = \sum_{j=1}^{N_s} x_j P_j, \tag{2}$$

where x_j is the On/Off indicator of the jth SBS, such that $x_j = 1$ and $x_j = 0$ indicate the On, Off mode, respectively. Now, let $x'_j = 1 - x_j$ denotes the complement of x_j such that $x'_j = 1$ indicates the Off mode, then the total power saving P'_t can be written as

$$P'_t = \sum_{j=1}^{N_s} x'_j P_j. \tag{3}$$

Thus, two modes of operation are considered, namely "On" (SBS in full operation) with 100 % power consumption, and "Off" with 0 % power consumption [16]. It should also be noted that the term "sleeping SBS" indicates an SBS that is operating in the "Off" mode and has 0 % power consumption.

2.2 Network Model

We consider a heterogeneous network consisting of one MBS and a set of N_s SBSs denoted by $\mathcal{S}$, where users can be associated with either the MBS or a nearby SBS. The management of all network elements is performed in the central BBU pool which is capable of taking network-wide decisions. In the context of H-CRANs, remote radio heads (RRHs) generally have lighter processing capabilities compared to SBSs; nevertheless, both SBSs and RRHs are denoted as SBSs in this work assuming they have similar functionality. The MBS is modeled as an $M/M/k_m$ queueing system in which k_m servers (radio channels) can serve k_m users simultaneously without waiting in the queue. Similarly, each SBS is modeled as an $M/M/k_s$ system with equal service rate but different arrival rates. Now, let λ_m, k_m, and μ_m denote the arrival rate of tasks (users) within only the MBS coverage (no SBS coverage), number of MBS servers, and MBS service rate, respectively, then the MBS utilization can be expressed as

$$\rho_m = \frac{\lambda_m}{k_m \mu_m}. \tag{4}$$

where ρ_m must be less than or equal to 1 in order to maintain system stability. To showcase the effect of SBS sleeping on cloud computing, it is assumed that tasks arriving at the MBS have no computing demands. In other words, ρ_m has no direct effect on the the cloud response time; nevertheless, it affects the number of sleeping SBSs (edge devices), and as a consequence, the amount of computing tasks offloaded on to the cloud.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing

TABLE 1: Base station sleeping strategies

Reference	Network Model	Computing Model	Performance Indicator(s)	Sleeping Initiator
[10]	HetNet	None	Energy-delay tradeoff	Number of tasks
[15]	HetNet	None	Energy and blocking probability	Traffic load
[16]	HetNet	None	Energy efficiency and coverage probability	Traffic load
[19]	HetNet	Cache-enabled SBSs	Power consumption & cache hit ratio	Harvested energy and traffic load
[20]	HetNet	None	Power consumption and throughput	Traffic load and user location
[21]	HetNet	None	Power Consumption	Traffic load
[22]	HetNet	None	Power consumption and coverage probability	Traffic load and network coverage
This work	H-CRAN	Cloud-edge	Power consumption, cloud response time, and user energy	Traffic, cloud response time, and task completion time

2.3 Computing Model

Tasks offloaded from sleeping SBSs are accommodated by the central cloud which is modeled as an $M/M/k_c$ queueing system with k_c servers or virtual machines (VMs). We generally classify tasks into two categories, computing tasks that require realtime processing and feedback from edge or cloud servers, and non-computing tasks that require telephony services without powerful computing capabilities thus can be handled by the cellular nodes. Let λ_j and α_j denote respectively the arrival rate of tasks and the ratio of computing tasks to all incoming tasks (computing plus non-computing) at SBS j, then the total arrival rate of computing tasks at the cloud is

$$\lambda_c = \sum_{j=1}^{N_s} x'_j \lambda_j \alpha_j, \tag{5}$$

Accordingly, the cloud utilization ρ_c is expressed as

$$\rho_c = \frac{\lambda_c}{k_c \mu_c}, \tag{6}$$

where μ_c denotes the service rate of each server. The performance metric of the system under consideration is the response time offered by the cloud. To this end, we consider the steady state analysis based on the continuous-time Markov chain (CTMC) of the $M/M/k_c$ cloud system as shown in Fig. 2.

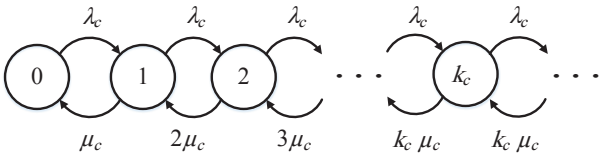


Fig. 2: Cloud queue model.

The probability that a user will have to queue (all servers are occupied) can be calculated as

$$\begin{aligned} P_Q &= \sum_{i=k_c}^{\infty} \pi_i \\ &= \pi_0 \frac{k_c^{k_c}}{k_c!} \frac{\rho_c^{k_c}}{1-\rho_c}, \end{aligned} \tag{7}$$

where π_i represents the steady state probability that i servers are occupied. π_0, which is the steady state probability that zero tasks exist in the cloud, can be written as

$$\pi_0 = \left[\sum_{i=0}^{k_c-1} \frac{(k_c\rho_c)^i}{i!} + \frac{k_c^{k_c}}{k_c!} \frac{\rho_c^{k_c}}{1-\rho_c} \right]^{-1}, \tag{8}$$

Therefore, the cloud response time can be obtained by

$$\begin{aligned} E[T_c] &= \frac{1}{\lambda_c} \cdot \frac{\rho_c}{1-\rho_c} \cdot P_Q + \frac{1}{\mu_c}, \\ &= \frac{k_c^{k_c}}{\lambda_c k_c!} \cdot \frac{\rho_c^{k_c+1}}{(1-\rho_c)^2} \cdot \left[\sum_{i=0}^{k_c-1} \frac{(k_c\rho_c)^i}{i!} + \frac{k_c^{k_c}}{k_c!} \frac{\rho_c^{k_c}}{1-\rho_c} \right]^{-1} + \frac{1}{\mu_c}. \end{aligned} \tag{9}$$

2.4 Cost of Task Migration from Edge to Cloud

In the proposed system, a VM is allocated to each computing task arriving to the edge device or the central cloud with fixed CPU speed. When the serving SBS enters the sleep mode, all associated VMs will be migrated to the central cloud. Thus, the cost of VM migration is considered as a delay constraint in the sleeping mechanism. Each task has a particular data size S_t that includes both application data and VM state [23], and a completion deadline θ_t by which the task must be executed and delivered to end-user. Therefore, the total experienced delay for accomplishing task t consists of three components (a) task execution time (b) data transmission time and (c) response time. Mathematically we can formulate the total delay as follows

$$d_t = \begin{cases} \frac{S_t}{v_e} + 2\frac{S_t}{b_s} + E[T_e], & \text{if SBS is On} \\ \\ \frac{S_t}{v_c} + 2\frac{S_t}{b_m} + 2\frac{S_t}{b_{fl}} + E[T_c], & \text{if SBS is Off} \end{cases} \tag{10}$$

where v_e and v_c denote the VM's CPU clock speed (cycles/sec) in the edge device and the cloud respectively. b_s, b_m, and b_{fl} are respectively the bit rate provided by the SBS, MBS, and the fiber backhaul link. $E[T_e]$ and $E[T_c]$ denote the response time at the edge device and the cloud, respectively. The multiplier "2" is used to calculate the time required for both uplink and downlink transmission. Taking into account the cost of task migration, the energy consumed by a mobile device is thus depends on whether the task t is processed by the near edge device or the distant cloud.

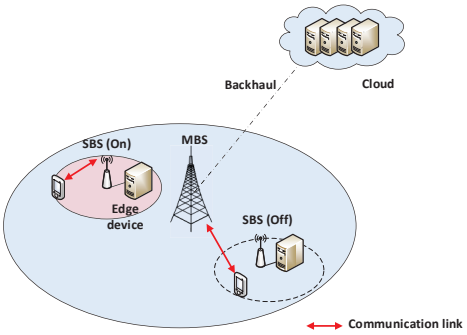
$$p_t = \begin{cases} p^s d_t, & \text{if SBS is On} \\ \\ p^m d_t, & \text{if SBS is Off} \end{cases} \tag{11}$$

where p^s and p^m denote the user transmit power to the SBS and MBS, respectively. Since this work incorporates different communication and computing aspects, we assume an appropriate service level agreement (SLA) is committed by cloud providers to ensure all clients have access to cloud facilities [24]. Moreover, the SLA between cloud operators and mobile devices ensures that all computing tasks are completed before the deadline and are compensated for the extra energy consumption due to sleeping SBSs.

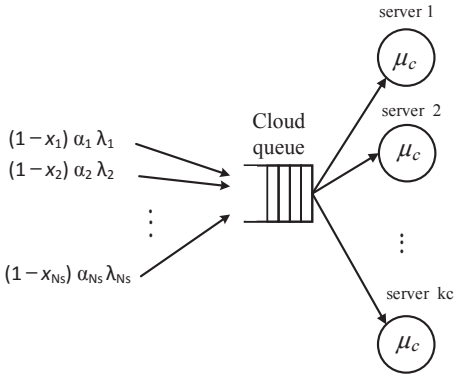
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing

3 Computing-Aware SBS Sleeping

As mentioned earlier, edge devices are assumed to coexist with SBSs and follow the same On/Off operation. Also, computing tasks associated with a sleeping SBS are offloaded to the cloud. However, offloading tasks from the edge to the central cloud can increase the time delay experienced by users; thus, it is essential to take into account the loading effect of computing tasks before deciding on whether to put an SBS in "On" or "Off" mode. Fig. 3 illustrates the system and queue models for the proposed sleeping mechanism.



(a) System layout.



(b) Cloud queue model.

Fig. 3: Proposed H-CRAN-Cloud-Edge system.

The computing-aware SBS sleeping mechanism is formulated as a 0-1 knapsack problem which in general aims to optimize the total value under the total weight constraint. Here, the arrival rate of tasks at SBS j (λ_j), which directly affects the utilization of the SBS $\rho_s = \frac{\lambda_j}{k_s \mu_j}$, is considered as the weight of the SBS. Let $\alpha'_j = 1 - \alpha_j$ denotes the ratio of non-computing tasks to all incoming tasks at SBS j, then $(\alpha'_j \lambda_j)$ is considered the value of that SBS. In other words, the objective of the problem is to maximize the number of sleeping SBSs that have less computing duties as follows

$$\begin{aligned} \textbf{P1: Maximize:} \quad & \max_{x'_j} \sum_{j=1}^{N_s} \alpha'_j \lambda_j x'_j \\ \textbf{Subject to:} \quad & C1: \sum_{j}^{N_s} \frac{\lambda_j x'_j + \lambda_m}{k_m \mu_m} \leq 1, \\ & C2: E[T_c] < \theta_c, \\ & C3: d_t < \theta_t, \forall t, \\ & C4: x'_j \in \{0,1\}, \quad \forall j \in \mathcal{S}, \\ & C5: \alpha'_j \in [0,1], \quad \forall j \in \mathcal{S}. \end{aligned} \tag{12}$$

It is worth mentioning that $x'_j = 1$ indicates that the jth SBS is Off and all computing tasks associated with that SBS are offloaded to the cloud through the MBS. The rationale behind this optimization problem which is solved using dynamic programming, is that SBSs with less computing duties (i.e., more non-computing tasks or higher α'_j) are considered to have higher values compared to other SBSs and thus put into the Off mode by setting $x'_j = 1$. The constraint C1 ensures that the total incoming tasks at the MBS (offloaded tasks plus MBS tasks) will not exceed the MBS utilization limit (i.e., $\rho_m = 1$) which is considered the total weight limit in the 0-1 knapsack problem. C2 and C3 set the upper time limit for the cloud response and task completion, respectively. C4 indicates that x'_j is a binary variable. C5 shows that α'_j can have any real value from 0 to 1.

It should be noted that the average arrival rates of tasks at all SBSs underlying an MBS are used to drive the base sleeping mechanism rather than the instantaneous number of tasks. This is because the sleeping mechanism in this work is centralized compared to other distributed sleeping schemes such as the N-policy in [10] that allows SBSs to individually decide on the sleep decision based on the instantaneous number of tasks.

4 SBS Sleeping in Shared Cloud-Edge Computing System

In the shared cloud-edge computing model, both cloud and edge servers cooperate in a sense that allows the workload to be shared among all available cloud and edge servers provided that at SBSs operate in the On mode. In other words, the queueing model of the shared computing systems has one queue and joint cloud-edge servers as elaborated in Fig. 4. In the proposed shared computing system, the arrival rate and server utilization are respectively written as

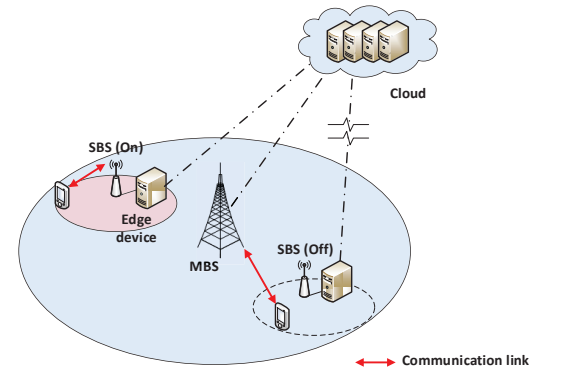
$$\lambda_{sh} = \sum_{j=1}^{N_s} \lambda_j \alpha_j, \tag{13}$$

$$\rho_{sh} = \frac{\lambda_{sh}}{k_{sh} \mu_{sh}}. \tag{14}$$

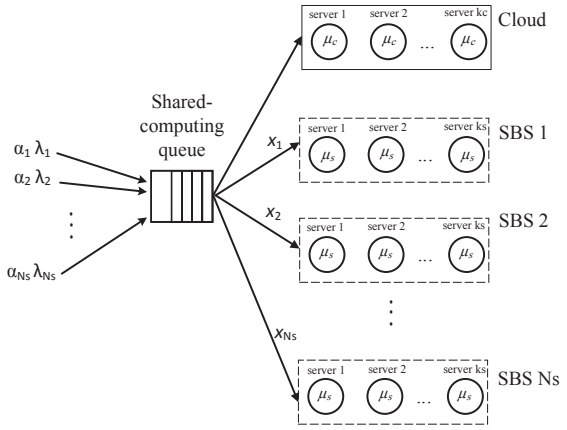
where k_{sh} is the total number of cloud and active edge servers ($k_{sh} = k_c + \sum_j^{N_s} x_j k_s$).

Lemma 1. *The cloud response time in the proposed shared computing model is faster than central cloud system by a factor of* $\left(1 + \frac{\sum_j^{N_s} x_j k_s}{k_c}\right)$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing



(a) System layout.



(b) Queue model.

Fig. 4: Shared computing system model.

Proof.

$$\frac{E[T_c]}{E[T_{sh}]} = \frac{\frac{1}{\lambda_c}\frac{\rho_c}{(1-\rho_c)}P_Q^c + \frac{1}{\mu_c}}{\frac{1}{\lambda_{sh}}\frac{\rho_{sh}}{(1-\rho_{sh})}P_Q^{sh} + \frac{1}{\mu_{sh}}} \tag{15}$$

where $E[T_{sh}]$ and P_Q^{sh} denote the response time and queuing probability for the shared computing system, respectively. By comparing the two systems at the same load level and queuing probability (i.e., $\rho_c = \rho_{sh} = \rho$, and $P_Q^c = P_Q^{sh} = P_Q$), we get

$$\begin{aligned}\frac{E[T_c]}{E[T_{sh}]} &= \frac{\frac{1}{\lambda_c}\frac{\rho}{(1-\rho)}P_Q + \frac{1}{\mu_c}}{\frac{1}{\lambda_{sh}}\frac{\rho}{(1-\rho)}P_Q + \frac{1}{\mu_{sh}}} \\ &= \frac{\frac{1}{\lambda_c}\rho P_Q + \frac{(1-\rho)}{\mu_c}}{\frac{1}{\lambda_{sh}}\rho P_Q + \frac{(1-\rho)}{\mu_{sh}}}\end{aligned}$$

where the last step is obtained by multiplying both numerator and denominator with $(1-\rho)$. When the system is heavily loaded (i.e., $\rho \approx 1$ and $P_Q \approx 1$), then the arrival rate equals the service rate (i.e., $\lambda_{sh} = \mu_{sh}$ and $\lambda_c = \mu_c$), thus

$$\frac{E[T_c]}{E[T_{sh}]} = \frac{\lambda_{sh}}{\lambda_c} = \frac{k_{sh}\mu_{sh}}{k_c\mu_c} = \frac{(k_c\mu_{sh} + \sum_j^{N_s} k_s\mu_{sh}x_j)}{k_c\mu_c}$$

Assuming that the service rate of both systems are equal (i.e., $\mu_{sh} = \mu_c$), then

$$E[T_c] = \left(1 + \frac{\sum_j^{N_s} x_j k_s}{k_c}\right)E[T_{sh}]$$

□

To find the optimal set of sleeping SBSs taking into account power consumption and cloud response time, the cost function is formulated as

$$C(\mathbf{x}) = \beta\left(\frac{P_t}{\max\{P_t\}}\right) + (1-\beta)\left(\frac{E[T_{sh}]}{\max\{E[T_{sh}]\}}\right) \tag{16}$$

where $\mathbf{x} = \{x_1, x_2, ..., x_{N_s}\}$ represents the operation status of all SBS in the system. Moreover, $\mathbf{x}$ has 2^{N_s} different combinations of binary numbers in a truth table style. Whereas $C(\mathbf{x})$, P_t, and $E[T_{sh}]$ are respectively the cost, total power consumption, and response time associated with $\mathbf{x}$. β is a weighting factor that determines whether to prioritize the minimization of power consumption or cloud response time. Thus, the problem of SBS sleeping in the shared computing model can be formulated as

$$\begin{aligned}\textbf{P2: Minimize:}\quad & \min_{\mathbf{x}} \; C(\mathbf{x}) \\ \textbf{Subject to:}\quad & C1: \sum_j^{N_s} \frac{\lambda_j x'_j + \lambda_m}{k_m\mu_m} \le 1, \\ & C2: \; E[T_c] < \theta_c, \\ & C3: \; d_t < \theta_t, \forall t, \\ & C4: \; x'_j \in \{0,1\}, \quad \forall j \in \mathcal{S}.\end{aligned} \tag{17}$$

The constraint C1 ensures that the total incoming tasks at the MBS do not exceed the MBS utilization limit. C2 and C3 set the time threshold for the cloud response and task completion, respectively. C4 indicates the On/Off operation of SBS j. To solve this mixed-integer optimization problem, exhaustive search which has been successfully used to find the optimal solution in similar problems [21] will be used. The optimal solution for this problem is obtained by testing 2^{N_s} different combinations of $\mathbf{x}$. For instance, if the system contains two SBSs, then four iterations will be conducted to test all possible SBS configurations 00, 01, 10, and 11, where these two digits represent the operation mode for each SBS. Therefore, when more SBSs exist in the system, the number of iterations to find the optimal solution will increase. Algorithm 1 illustrates the solution search strategy. where $\mathbf{x}^*$ represents the optimal operation for the SBSs under consideration.

5 Simulation Setup and Results

To evaluate the performance of the proposed computing-aware sleeping mechanism, simulation setup and results are provided and elaborated in this section. Table 2 lists the description, notation, and value for each parameter used in the simulation. The data size and task completion deadline are uniformly distributed between [0.5-2] MB and [2-4] sec, respectively. Following the work in [20], the total provided cellular throughput is 27 Mbps by the MBS and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing

Algorithm 1: Searching optimal solution for **P2**

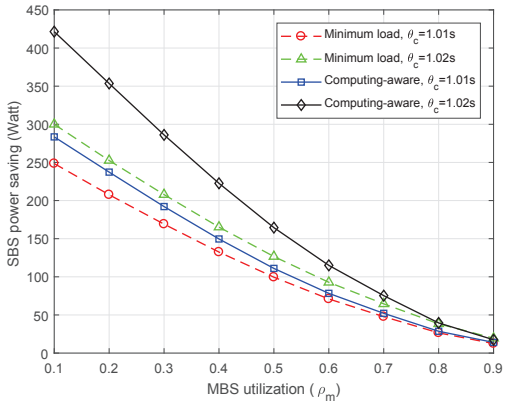
Initialize $\mathbf{x_n}$, $n = 1, 2, ..., 2^{N_s}$;
while $n < 2^{N_s}$ **do**
  Calculate ρ_m according to (4);
  if $\sum_j^{N_s} \frac{\lambda_j x_j' + \lambda_m}{k_m \mu_m} \leq 1 - \rho_m$ **then**
    Calculate $C(\mathbf{x_n})$ according to (16);
  end
  $n \leftarrow n + 1$;
end
$\mathbf{x}^* = \underset{\mathbf{x_n}}{\text{argmin}}\{\mathbf{C}\}$.

15 Mbps by the SBS. It should be noted that the data rates here are shared by all users such that when the number of users increases, the per-user data rate will decrease. Other parameter settings are inspired by [25] and [26].

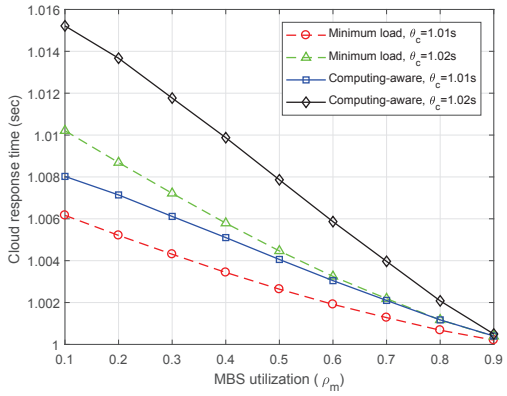
Fig. 5 shows the system performance using both minimum load and computing-aware mechanisms. The minimum load approach is greedy-based and controls SBS sleeping according to only the SBS load (i.e., arrival rate) without considering the computing demand. On the other hand, the computing-aware mechanism determines the sets of active and sleeping SBSs considering both the arrival rate and the amount of computing tasks. It can be seen in Fig. 5 (a) how θ_c affects the SBS power saving since it acts as a constraint on the cloud response time and thus the number of sleeping SBSs. Moreover, the computing-aware mechanism is found to achieve better power saving since it considers the computing load when selecting the sleeping SBSs and that also leads to reduced cloud response time.

It is also clear how the MBS utilization (ρ_m) significantly impacts the overall performance. When the MBS is lightly loaded (e.g., $\rho_m = 0.1$), both power saving and response time were found to achieve higher values since more MBS servers are free and willing to accept more offloaded tasks from more sleeping SBSs. Nevertheless, having more sleeping SBSs increased the cloud response time because more computing tasks are directed to the central cloud instead of being processed locally by edge devices. In contrast, when the MBS is heavily loaded (e.g., $\rho_m = 0.9$), both power saving and cloud response time are decreased since SBSs have smaller opportunities to enter the sleep mode; as a result, less computing tasks are offloaded to the central cloud. The response time in this system does not fall below $1s$ since the proposed service rate in the cloud (μ_c) is set to $1s$ and thus the service time is $1/\mu_c = 1s$. This service time in addition to the cloud queue delay constitutes the cloud response time as shown in equation 9. Also, it should be noted that the task completion deadline θ_t follows a uniform distribution between 2 and 4 seconds (i.e. mean $\bar{\theta}_t$=3s).

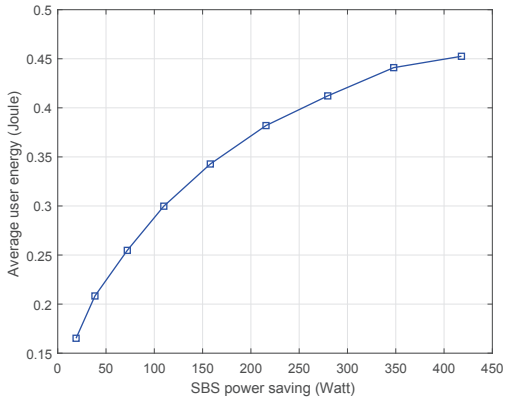
Fig. 5 (c) shows that having more power saving due to SBS sleeping will oblige users to spend more energy since tasks will be forwarded to the longer cloud path via the MBS rather than being processed by the edge device. Moreover, the non-linearity in the user energy consumption comes as a result of the more abundant frequency resources offered by the MBS when it is lightly loaded which is reflected by the higher SBS power saving. As a result, less time will be required to complete tasks since higher data rates will be offered by the MBS, and thus less energy consumption.



(a) SBS power saving.



(b) Cloud response time.



(c) User energy consumption.

Fig. 5: SBS power saving, cloud response time, and user energy consumption under different values of θ_c in disjoint cloud-edge system, $\bar{\theta}_t = 3s$.

In Fig. 6, comparisons between disjoint and shared computing cloud-edge systems are conducted. As proved in lemma 1, having more active SBSs in the system reduces the cloud response time. This can be observed especially when ρ_m is high which forces more SBSs to remain active

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing

TABLE 2: Simulation Parameters

Description	Notation	Value
SBS power	P_s	50 W
Number of SBSs	N_s	10
Number of servers in the MBS	k_m	100
Number of servers in the edge device	k_s	10
Number of servers in the cloud	k_c	50
Task arrival rate at MBS	λ_m	1-100 task/sec
Task arrival rate at the jth SBS	λ_j	1-10 task/sec
Computing ratio at the jth SBS	α_j	[0,1]
MBS service rate	μ_m	1 task/sec
SBS service rate	μ_s	1 task/sec
Cloud service rate	μ_c	1 task/sec
CPU clock speed of each VM in the edge device	v_e	3.2 GHz
CPU clock speed of each VM in the cloud	v_c	3.2 GHz
Total bit rate provided by the SBS	b_s	15 Mbps
Total bit rate provided by the MBS	b_m	27 Mbps
Fiber backhaul link speed	b_{fl}	10 Gbps
Data size of tasks	S_t	0.5-2 MB
User transmit power to the SBS	p^s	0.05 W
User transmit power to the MBS	p^m	0.5 W

thus reducing the overall response time. To maintain fair evaluation, comparisons in Fig. 6 (a) and (b) were obtained using only the minimum load approach without imposing a delay constraint on the cloud response time nor having a task completion deadline, and that is why they seem to have different shapes compared to other results in this section.

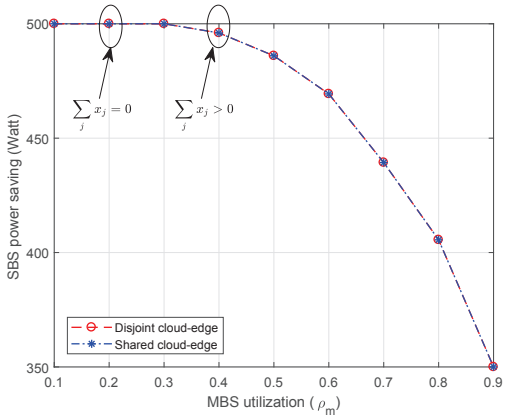
Fig. 7 (a) shows the optimal sleeping solution in the shared computing system. By adjusting the value of the weighting factor β, preference can be given to either saving energy or reducing the response time. Here, when $\beta = 0.8$ more emphasis is put on power saving than reducing cloud response time. Furthermore, adding more stringent requirements such as θ_c on the cloud response time and θ_t on the task completion time will affect the power saving significantly as seen in Fig. 7 (b) and (c).

Finding the optimal solution requires searching all possible operation modes for all SBS within the MBS coverage. Since there are only two operation modes, the total number of required iterations is 2^{N_s} as illustrated in Algorithm 1. Here, it is helpful to measure the time required to find the optimal solution for different numbers of SBSs although 10 SBSs were considered in this work. Table 3 lists the time required to reach the optimal solution.

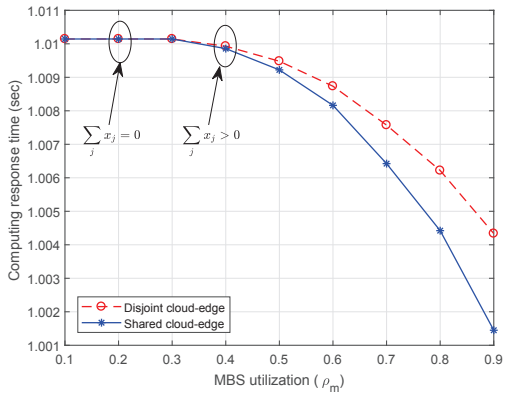
TABLE 3: Simulation Parameters

Number of SBSs (N_s)	Time to find x^*
5	0.015 s
10	1.25 s
15	75 s
16	165 s
17	360 s
18	800 s

It can be observed that finding the optimal solution requires longer time as the number of SBSs underlying an MBS increases; in which case, the search space need to be reduced by assigning particular SBSs a fixed mode of operation using network and cloud characteristics [24]. Furthermore, reinforced learning techniques can be implemented to extract the features of user behaviour to help decide on each particular SBS operation and hence reduce the search space.



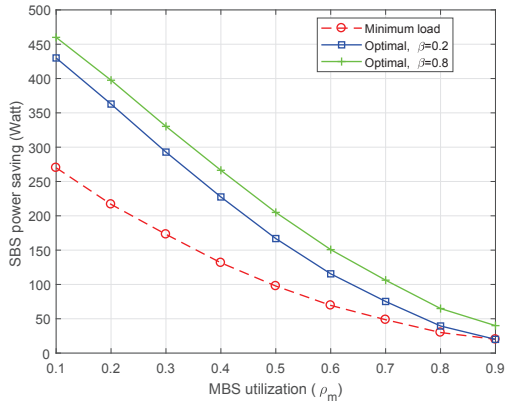
(a) SBS power saving.



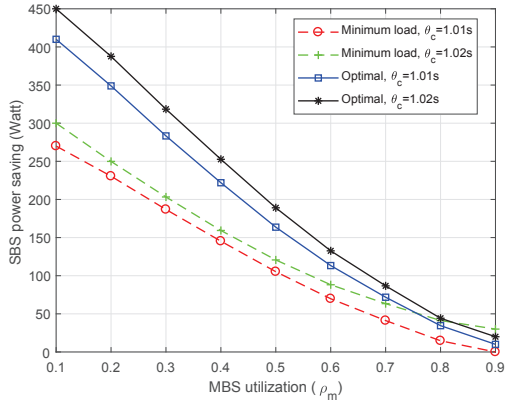
(b) Cloud response time.

Fig. 6: Performance comparison between disjoint and shared computing systems.

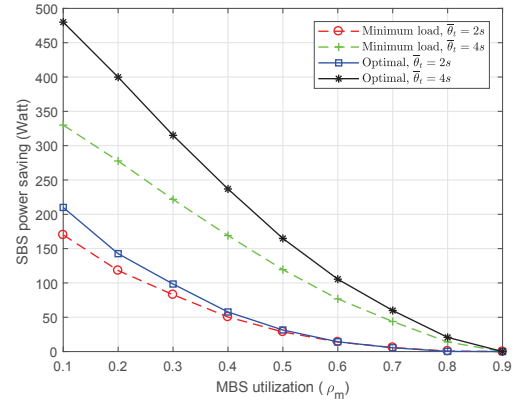
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2019.2893228, IEEE Transactions on Cloud Computing



(a) SBS Power saving using different values of β.



(b) SBS Power saving under different cloud response constraints, ($\theta_t = 3s$, $\beta = 0.8$).



(c) SBS power saving under different task completion deadlines, ($\theta_c = 1.02s$, $\beta = 0.8$).

Fig. 7: SBS power saving in shared cloud-edge system.

6 Conclusion

The problem of SBS sleeping in integrated H-CRAN-cloud-edge networks is studied in this paper. First, a SBS sleeping mechanism was proposed to save energy taking into account the constraints of task completion deadline and cloud response time. The problem was formulated as a 0-1 knapsack problem and solved using dynamic programming. Secondly, a joint cloud-edge computing model was introduced such that edge devices contribute to the total network computing resources beside the cloud to improve the system computing capability. Finally, finding the optimal power saving in the later system was found using an exhaustive search strategy. Abiding by the fact that traffic associated with sleeping SBSs will be eventually served by the MBS, the MBS utilization was considered as a major practical constraint that defines the observations and results obtained in this work.

References

[1] A. Alnoman and A. Anpalagan, "Towards the Fulfillment of 5G Network Requirements: Technologies and Challenges," *Telecommunication Systems*, vol. 65, no. 1, pp. 101–116, May 2017.

[2] A. Alnoman, G. H. Carvalho, A. Anpalagan, and I. Woungang, "Energy Efficiency on Fully Cloudified Mobile Networks: Survey, Challenges, and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1271 – 1291, 2nd Quart. 2018.

[3] J. Li, L. Huang, Y. Zhou, S. He, and Z. Ming, "Computation Partitioning for Mobile Cloud Computing in a Big Data Environment," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2009 – 2018, Aug. 2017.

[4] A. Brogi and S. Forti, "QoS-Aware Deployment of IoT Applications Through the Fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185 – 1192, Oct. 2017.

[5] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the Edge: A Scalable IoT Architecture Based on Transparent Computing," *IEEE Network*, vol. 31, no. 5, pp. 96 – 105, Oct. 2017.

[6] S. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for Vehicular Communications," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 111 – 117, Jan. 2018.

[7] Y. Shih, W. Chung, A. Pang, T. Chiu, and H. Wei, "Enabling Low-Latency Applications in Fog-Radio Access Networks," *IEEE Network*, vol. 31, no. 1, pp. 52 – 58, Dec. 2017.

[8] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171 – 1181, Dec. 2016.

[9] S. Hung, H. Hsu, S. Lien, and K. Chen, "Architecture Harmonization Between Cloud Radio Access Networks and Fog Networks," *IEEE Access*, vol. 3.

[10] Z. Niu, X. Guo, S. Zhou, and P. R. Kumar, "Characterizing EnergyDelay Tradeoff in Hyper-Cellular Networks With Base Station Sleeping Control," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 641 – 650, Apr. 2015.

[11] T. Oo *et al.*, "Offloading in HetNet: A Coordination of Interference Mitigation, User Association, and Resource Allocation," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2276 – 2291, Aug. 2017.

[12] X. Meng, W. Wang, and Z. Zhang, "Delay-Constrained Hybrid Computation Offloading With Cloud and Fog Computing," *IEEE Access*, vol. 5.

[13] Y. Cui and D. Jiang, "Analysis and Optimization of Caching and Multicasting in Large-Scale Cache-Enabled Heterogeneous Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 250 – 264, Jan. 2017.

[14] X. Lyu *et al.*, "Selective Offloading in Mobile Edge Computing for the Green Internet of Things," *IEEE Network*, vol. 32, no. 1, pp. 54 – 60, Feb. 2018.

[15] G. Carvalho, I. Woungang, A. Anpalagan, and E. Hossain, "QoS-Aware Energy-Efficient Joint Radio Resource Management in Multi-RAT Heterogeneous Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6343 – 6365, Aug. 2016.

[16] C. Liu, B. Natarajan, and H. Xia, "Small Cell Base Station Sleep Strategies for Energy Efficiency," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1652 – 1661, Mar. 2016.

[17] L. Tong, Y. Li, and W. Gao, "A Hierarchical Edge Cloud Architecture for Mobile Computing," in *IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[18] T. H. Szymanski, "An Ultra-Low-Latency Guaranteed-Rate Internet for Cloud Services," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 123 – 136, Feb. 2016.

[19] D. Xu, H. Jin, C. Zhao, and D. Liang, "Joint Caching and Sleep-Active Scheduling for Energy-Harvesting Based Small Cells," in *IEEE International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct. 2017, pp. 1–6.
[20] L. Saker, S. E. Elayoubi, R. Combes, and T. Chahed, "Optimal Control of Wake Up Mechanisms of Femtocells in Heterogeneous Networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 664 – 672, Apr. 2012.
[21] J. Kim, W. Jeon, and D. Jeong, "Base-Station Sleep Management in Open-Access Femtocell Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3786 – 3791, May 2016.
[22] C. Chang, W. Liao, H. Hsieh, and D. Shiu, "On Optimal Cell Activation for Coverage Preservation in Green Cellular Networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2580 – 2591, Nov. 2014.
[23] M. Islam, M. Razzaque, M. Hassan, W. Ismail, and B. Song, "Mobile Cloud-Based Big Healthcare Data Processing in Smart Cities," *IEEE Access*, vol. 5.
[24] L. Gkatzikis and I. Koutsopoulos, "Migrate or Not? Exploiting Dynamic Task Migration in Mobile Cloud Computing Systems," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 24 – 32, June 2013.
[25] M. Chowdhury, E. Steinbach, W. Kellerer, and M. Maier, "Context-Aware Task Migration for HART-Centric Collaboration over FiWi Based Tactile Internet Infrastructures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1231 – 1246, June 2018.
[26] H. Wu and K. Wolter, "Stochastic Analysis of Delayed Mobile Offloading in Heterogeneous Networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 461 – 474, Feb. 2018.

Ali Alnoman received his B.Sc. and M.Sc. degrees in electrical engineering from the University of Baghdad, Iraq, in 2009 and 2012, respectively. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at Ryerson University, Canada. His research interests include energy efficiency and resource allocation in HetNets and cloud computing. During 2012-2014, he worked as a faculty member at Ishik University, Erbil, Iraq. He also served as a technical program committee member in the IEEE vehicular technology conference VTC2017-Fall, in Toronto.

Alagan Anpalagan (S98-M01-SM04) received the B.A.Sc., M.A.Sc., and Ph.D. degrees in electrical engineering from the University of Toronto, Canada. He joined with the ELCE Department, Ryerson University, Canada in 2001, and was promoted to Full Professor in 2010. He served the Department as Graduate Program Director (2004–2009) and the Interim Electrical Engineering Program Director (2009–2010). During his sabbatical (2010–2011), he was a Visiting Professor at Asian Institute of Technology, Thailand, and Visiting Researcher at Kyoto University, Japan. His industrial experience includes working for three years with Bell Mobility, Nortel Networks, and IBM. He directs a research group working on radio resource management (RRM) and radio access and networking (RAN) areas within the WINCORE Laboratory. He also completed a course on Project Management for Scientist and Engineers at the University of Oxford CPD Center, Oxford, U.K. He has coauthored three edited books: *Design and Deployment of Small Cell Networks* (Cambridge University Press, 2016), *Routing in Opportunistic Networks* (Springer, 2013), *Handbook on Green Information and Communication Systems* (Academic Press, 2012) and a book: *Game-theoretic Interference Coordination Approaches for Dynamic Spectrum Access* (Springer, 2016). His research interests include 5G wireless systems, energy harvesting and green communications technologies, cognitive radio resource management, wireless cross layer design and optimization, cooperative communication, M2M and sensor communication, small cell, and heterogeneous networks. He served as an Editor for the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (2012-2014), IEEE COMMUNICATIONS LETTERS (2010-2013), and EURASIP Journal of Wireless Communications and Networking (2004-2009). He also served as Guest Editor for six special issues: IEEE WIRELESS COMMUNICATIONS: Sustainable Green Networking and Computing in 5G Systems, IEEE ACCESS on Internet of Things in 5G Systems, IET COMMUNICATIONS on Evolution and Development of 5G Wireless Communication Systems, EURASIP on Radio Resource Management in 3G+ systems and on Fairness in Radio Resource Management for Wireless Networks, and MONET on Green Cognitive and Cooperative Communication and Networking. He served as TPC Co-Chair, IEEE VTC Fall 2017, TPC Co–Chair, IEEE INFOCOM'16: First International Workshop on Green and Sustainable Networking and Computing, IEEE Globecom15: SAC Green Communication and Computing, IEEE PIMRC11: Cognitive Radio and Spectrum Management. He served as Vice Chair, IEEE SIG on Green and Sustainable Networking and Computing with Cognition and Cooperation (2015–16), IEEE Canada Central Area Chair (2012-2014), IEEE Toronto Section Chair (2006-2007), ComSoc Toronto Chapter Chair (2004-2005), and IEEE Canada Professional Activities Committee Chair (2009-2011). He was the recipient of the Deans Teaching Award (2011), Faculty Scholastic, Research and Creativity Award (2010 and 2014), Faculty Service Award (2011 and 2013) from the Ryerson University. He was also the recipient of IEEE M.B. Broughton Central Canada Service Award (2016), Exemplary Editor Award from IEEE ComSoc (2013) and Editor-in-Chief Top10 Choice Award in Transactions on Emerging Telecommunications Technology (2012) and, a coauthor of a paper that received IEEE SPS Young Author Best Paper Award (2015). He is a Registered Professional Engineer in the province of Ontario, Canada and Fellow of the Institution of Engineering and Technology (FIET).