A multi-class channel access scheme for cognitive edge computing-based internet of things networks

Samuel D. Okegbile, *Student Member, IEEE,* Bodhaswar T. Maharaj, *Senior Member, IEEE,* and Attahiru S. Alfa

***Abstract*—Edge computing-based framework is capable of improving users' quality of experience in cognitive internet of things (IoT) networks. To explore the advantages of this edge computing-based framework, possible offloading and processing delay resulting from computation bottlenecks, and the offloading latency caused due to inter-cell interference must be properly considered. This paper thus considered a multi-class channel access mechanism for cognitive edge computing-based IoT networks where IoT users were categorized based on their quality of experience requirements. Essential IoT devices are permitted to offload to the edge server at any time following the hybrid channel access model, while delay-tolerant IoT devices are only permitted to offload to the server when the channel is idle following the overlay channel access model. Analyses were obtained for transmission rate and offloading delay to demonstrate the performance of the proposed mechanism, while important metrics such as total offloading latency and total offloading cost were investigated. The total offloading costs were formulated through the mixed strategy Nash equilibrium method. The obtained results showed that multi-class channel access mechanisms can reduce packet offloading delay in cognitive edge computing-based IoT networks.**

***Index Terms*—channel access, cognitive IoT, edge computing, data offloading, delay**

I. Introduction

The demand for wireless communication systems and applications continues to increase, and the need to improve the data rate of wireless communications has recently been attracting a lot of attention. Wireless applications are now been deployed in various aspects of lives under the framework of the internet of things (IoT) and the number of connected things is expected to keep increasing. As a result, channel access and availability remain a great threat to the continuous evolution of wireless technologies. Hence, enhancing the communication quality among connected users becomes an important topic to study. It is, therefore, unsurprising that enhancing the communication quality among heterogenous users has recently been the focus of much research.

One preliminary method of enhancing the communication quality among network users is through the adoption of relaying techniques [1], [2] where various relaying mechanism such as decode and forward, as well as amplify and forward relaying mechanisms have been exploited with the aim of improving users' transmission experience in the scarce spectral resources. The adoption of the cognitive radio network (CRN) also continues to be an interesting method of enhancing users' spectrum access demands. In cognitive IoT (C-IoT) networks, IoT devices are expected to intelligently access the spectral resources belonging to primary users (PUs) based on some predefined requirements as in CRN, to meet their channel access requirements for data offloading, while also ensuring that the activities of other users are not disrupted. However, due to the time required to offload data to the cloud for computation and processing, it may be difficult to satisfy the channel access demands of C-IoT users, especially in networks with dominant activities of PUs. As a result of this, the mobile edge-based computation (MEC) technique can be adopted, owing to its ability to reduce latency and energy consumption, while improving communication quality among IoT users [3].

Despite the availability of communication, computation and storage capacities at various IoT devices, most of the requirements of the next-generation systems cannot be met at the users' level. For instance, when deployed in a smart environment, e-health, interactive gaming and vehicular technology, data must be offloaded for efficient computation owing to the limited processing and storage capabilities available via local CPUs at the users' level. MEC provides a cloud computing capability at the edge to improve communications and task computation [4]. It offers faster access to computing and storage resources due to its advantages of closeness, diversity and resource-constrained. Through edge-based solutions, some data can be offloaded to the MEC server, while only data that require heavy computation are offloaded to the cloud.

Data offloading to the MEC server can, however, be very challenging, especially in large-scale IoT networks, where multiple data are offloaded simultaneously through the scarce spectrum resources. This multiple offloading can result in interference and increase the rate of data loss. It is hence imperative to consider an efficient offloading scheme to enhance the effective offloading process. This paper thus presents a cognitive edge computing-based approach, where offloading of essential data are prioritized to reduce the number of simultaneous offloading processes. We defined a cognitive edge computing-based IoT environment as an environment where cognitive IoT devices offload their data to edge servers for computation in an opportunistic manner on channels that are assigned to the licensed users. This is important to ensure that the offloading process of cognitive users is well coordinated

S.D. Okegbile was with Department of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa. He is currently with the Department of Electrical and Computer Engineering, Concordia University, Montreal, CA. E-mail: samuel.okegbile@concordia.com.

B.T. Maharaj is with Department of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa. E-mail: sunil.maharaj@up.ac.za.

A.S. Alfa is with Department of Electrical and Computer Engineering, University of Manitoba, Canada and Department of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa. E-mail: attahiru.alfa@up.ac.za.

to reduce interference and ensure a faster offloading process for delay-sensitive users. This will improve the offloading experience of IoT devices in large-scale IoT networks.

A. Related Work

To improve the offloading experience of IoT devices, many works have recently focused on improving communications and task computation experience among multiple IoT users. These efforts are necessary to reduce huge amounts of energy, time and bandwidth required to offload big data to the cloud [5]–[7]. In [7], an edge-based IoT (EdgeIoT) architecture was proposed to accommodate multiple connections and efficient transmissions. Similarly, a deep learning approach towards effective dynamic computation offloading was proposed in [8] for single-user single-cell MEC network. Since users' competition for channel resources often challenge the offloading process resulting in interference, interference was considered as an important element of offloading process in [9], [10]. Packets were offloaded in [10] based on the interference level and available computation resources.

A partial task offloading problem was formulated in [11] for the vehicular edge-based computing environment. Transmission rates were estimated based on some practical assumptions. Similarly, a cloud-based edge computing offloading framework was proposed in [12] for vehicular networks. Through the adopted Stackelberg game-theoretic approach, the authors were able to design an efficient multilevel offloading scheme. In [13], a multi-subtask and multi-server approach was proposed where different sub-tasks were offloaded to different servers for computation. The offloading process was achieved based on the required offloading distance and processing capacity. The offloading scheme was determined through the required energy consumption and bandwidth capacity in [14].

In a similar work, the authors in [15] formulated an offloading problem in the ultra-dense network as a mixed-integer nonlinear problem through the adoption of the software-defined network mechanism. The NP-hardness of the formulated offloading problem was proved. A collaborative offloading solution was proposed in [16] with a focus on supporting the coexistence of centralized cloud and multi-access edge computing. Similarly, a selective offloading scheme was proposed in [17], where cloud, edge, and local resources were integrated for improved performances. In [18], an offloading decision-making problem for a multi-user and multi-destination computation offloading scheme was formulated as a sequential game, while the Nash equilibrium of the game was shown to exist. A deep reinforcement learning-based approach was introduced in [19] to improve latency and reliability of the networks, while the authors in [20] proposed a Stackelberg game-based approach to improve computation offloading in cognitive industrial IoT.

It is worth noting that some of the existing edge-based solutions focused on single-user offloading problems [8], where interference is often neglected to avoid complicated analysis. However, in a multi-user environment, each user's operations can be significantly affected by interference from other users. Moreover, users are known to interact and compete with each other for computing resources. Hence, an independent offloading assumption is not sufficient.

Although the interference level and undesirable transmission delay [10] during offloading can be reduced through the local computation when possible [21], a large volume of packets is still expected to be offloaded in large-scale or ultra-dense networks for satisfactory computation. This will significantly increase the overall power consumption in the network. This excessive power consumption can be reduced when task offloading processes are achieved with a limited rate of re-transmission and packet delay, hence some IoT devices can desist from offloading until an appropriate channel is available to satisfy their channel access demands. This will also reduce inter-cell interference. It is, therefore, important to classify the enormous IoT devices based on their expected quality of experience (QoE) requirements, where channel access is granted following the priority class of each user, thus the justification for the adoption of the cognitive EdgeIoT in this paper. This ensures that the cognitive feature is integrated into edge services to make sure essential packets are offloaded with limited delay.

B. Contributions

We hence proposed a multi-class delay-constrained channel access mechanism capable of improving users' QoE in a cognitive edge computing-based network. To the best of our knowledge, a multi-class channel access mechanism for cognitive EdgeIoT networks has never been considered before. IoT devices were categorized based on their QoE requirements as shown in Fig. 1 to reduce the network interference and average delay. Essential IoT devices are permitted to share channels with high priority users following the hybrid channel access scheme, while delay-tolerant users offload following overlay channel access scheme (i.e. when all high priority users are not offloading). We assumed that class priorities are assigned by the operator depending on users' QoE requirements. Similar to [10], [21], we ignored the download transmission period since users' receiving power is much smaller than the transmission power, while the size of the output data is much smaller than the size of the input data. The contributions of this paper are thus summarized as follows:

- We modeled the proposed multi-class delay-constrained offloading scheme through an integrated hybrid channel access scheme and dynamic channel reservation techniques to ensure offloading retainability of CUs packets.
- To capture the relationship between the spatial geometry of wireless links and their temporal traffic dynamic, we integrated the techniques of stochastic geometry (SG) and queueing theory via a spatiotemporal characterization of the proposed cognitive Edge IoT networks.
- Analyses were then obtained for important metrics of interest to demonstrate the effectiveness of the proposed channel access mechanism on essential and delay-tolerant IoT devices.
- Based on the obtained analyses for the average departure rate and offloading delay, we obtained analysis for total offloading latency – a metric that captures the total

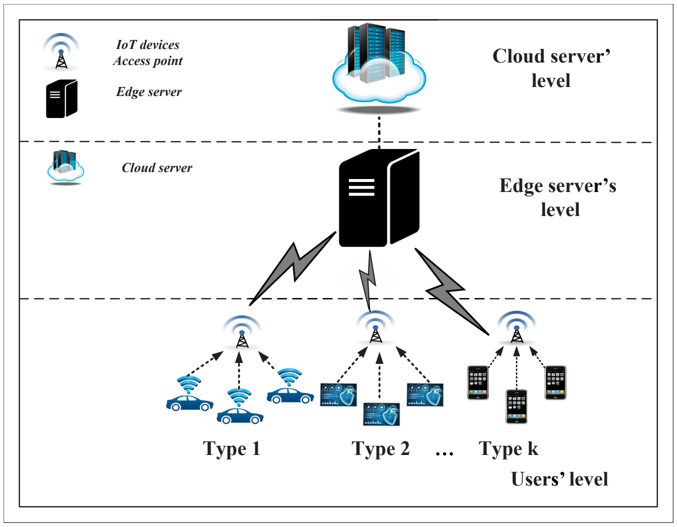

Fig. 1. Multi-class edge server access.

latency experienced by a packet from the time in which such a packet was generated to the time in which its computation was completed.

- Finally, we adopted the game theory approach via the mixed strategy Nash equilibrium to obtain the total offloading cost of each packet. We compared the total offloading cost of a packet with the cost of its local computation to demonstrate the advantages of packet offloading in the proposed cognitive EdgeIoT system. We demonstrated that the proposed channel access scheme can improve users' experience in large-scale networks.

We defined the departure rate as the rate at which packets are being successfully received at the edge server, while the offloading delay is the total time from generation to the successful reception of packets at any edge node.

C. Organization

The rest of this paper is structured as follows: The system model is presented in Section II, while the proposed multi-class channel access scheme is described in Section III. We then present the analyses of some selected performance metrics of interest in Section IV. Section V presents the details of the adopted mixed strategy Nash equilibrium – a game theoretic-based technique – while Section VI presents the outcomes of the numerical simulations and results. Section VII concludes the paper.

II. System model

In this section, we present the spatial and traffic models for the proposed cognitive Edge-IoT network and introduced the proposed integrated multi-class delay-constrained offloading scheme.

A. Spatial and Temporal Model

We considered a discrete time-based cognitive EdgeIoT network, where PUs and CUs are distributed following two independent homogeneous Poisson point processes (PPPs) Ψ_p and Ψ_c of intensities μ_p and μ_c respectively, given that $\mu_p \ll \mu_c$. CUs $\chi_i \in \Psi_c$ are C-IoT devices trying to satisfy their channel access demands for offloading processes. These C-IoT users were categorized as type k users (see Fig. 1) depending on their required QoE. For simplicity, we considered only type $k = \{1, 2\}$, where the distribution of type 1 and type 2 users also follow two independent PPPs Ψ_c^1 and Ψ_c^2 of intensities μ_c^1 and μ_c^2 respectively, with $\mu_c^1 \ll \mu_c^2, \forall \Psi_c = \Psi_c^1 \cup \Psi_c^2$. Type 1 CUs (also referred to as type 1 users) are essential IoT devices and enjoy priority over type 2 CUs (i.e. type 2 users). The proposed scheme can be extended to the case of $k > 2$ with little modifications. Note that PUs behaviors within the channel are known to be independent of the activities of C-IoT users since PUs have full privilege of offloading on their designated frequency bands without informing CUs. All users offload their data to the edge server for computation and only data that cannot be computed at the edge level are forwarded to the cloud server by the edge server.

To capture the temporal model, we adopt a discrete-time queueing model, where the time axis is segmented into slots of equal intervals. We then assumed that packets arrival and departure from the system occur around the slot boundaries. We considered packets' arrival rates of PUs, type 1 and type 2 CUs to follow independent Bernoulli processes α_p, α_1 and α_2 respectively. The buffer size at the edge server is considered to be infinite.

With this, a typical type 1 CU, $\chi_k^1 \in \Psi_c^1$, with non-empty buffer offloads to the edge server at any time slot following the offloading decision parameter $o_c \in \{0, 1\}$. The parameter $o_c = 0$ captures the event that a generated packet is computed locally. Similarly, a typical type 2 CU $\chi_k^2 \in \Psi_c^2$ offloads following the offloading decision parameter o_d. The parameter o_d depends on the joint activities of PU and type 1 CUs in such a time slot, with $o_d = 1$ representing the event that the band is available for type 2 CUs and $o_d = 0$ if otherwise. The IoT devices (type 1 and type 2 CUs) adopt the channel inversion power control [22], [23], where each user adjusts its transmit power, such that the received power level at the edge server equals a predetermined threshold P_o and P_u for overlay and underlay respectively. P_o depicts the offloading power of any CU (type 1 or 2) in the overlay channel access mode and P_u depicts the offloading power of type 1 CUs in the underlay channel access mode, with $P_o > P_u$.

B. Packet offloading with channel reservation scheme

The licensed channel of PUs is divided into M multiple cells or sub-channels of equal bandwidth where each user chooses a vacant cell uniformly randomly and independent of the location and the time slot (i.e., memoryless both spatially and temporally). These sub-channels can be occupied by the assigned PUs at any time. Thus, each CU can only opportunistically offload to the edge server through any licensed sub-channel, such that the activities of CUs do not generate disruptive interference in the primary network [24], [25]. The activities of PUs can, therefore, be represented as a simple queueing system, where the departure rate is only affected by negligible interference from the activities of CUs. We hence focus on CUs since most of these IoT devices are expected to access scarce spectrum resources as CUs rather than PUs.

When no PU is currently offloading, type 1 CUs offload with full offloading power as in the overlay channel access mode to enhance their throughput. Since PUs can appear at any time to offload on their assigned cells thereby forcing currently offloading type 1 CUs to terminate their offloading, a certain number of sub-channels R are reserved for CUs. This is important to ensure that type 1 CUs can continue to offload their packets with full power on the reserved sub-channels (R-SCH) when at least one PU is active on the non-reserved sub-channels (N-SCH). Owing to the pre-emptive nature of PUs, type 1 CUs offloading can also be interrupted in the R-SCH by PUs when a newly arriving PU finds no vacant sub-channel among the N-SCH. In such a case, type 1 CUs adjust their transmission powers and continue packet offloading following underlay channel access mode, provided that the interference generated at the nearest PU (I^n_{PU}) is less than the predefined interference temperature γ, defined as the total allowable interference level in a cell [26]. Otherwise, type 1 user's offloading is blocked. With this, the offloading of type 1 CUs follows the hybrid channel access scheme.

On the other hand, type 2 CUs are only permitted to offload on the N-SCH when no PU and type 1 CU is offloading following the overlay channel access mode. On the arrival of either PU or type 1 user, any currently offloading type 2 CU is moved to the R-SCH for continued offloading. Such a type 2 CU is interrupted and moved to a virtual queue when any PU or type 1 CU arrives on the R-SCH. Any interrupted type 2 CU repeats its entire data offload the next slot the channel is available. We assumed that interference within a cell is negligible, while the interference between N-SCH and R-SCH is also negligible. Similar to [9], [10], data offloading from each user located within each cell follows orthogonal frequency multiple access. We further assumed that edge server-to-cloud communication is achieved through error and interference-free bands. Hence, we focus on users-to-edge server offloading processes.

III. Multi-class channel access and preliminary derivations

In this section, we leverage the model and channel access scheme presented in the last section and describe the key algorithms of the proposed framework. We then present some preliminary results on the packet offloading and processing times.

One uniqueness of the proposed framework is its ability to provide continuous channel access to type 1 CUs when PUs arrive on the N-SCH through the reservation of a certain number of sub-channels R. This parameter R is dynamically adjusted depending on the ongoing channel occupancy status to ensure packets offloading retainability in the system. Let N_f and P_R represent the total number of idle N-SCH sub-channels during the current observation period and the predicted number of users that will move to R-SCH in the next period respectively, we used a simplified dynamic channel reservation method for the estimation of an average number of cells to be reserved for the next period R_{res}. This is given in Algorithm 1.

Algorithm 1: Simplified dynamic cell reservation

```
  **Input:** $R$, $N_f$, $P_R$
  **Output:** $R_{res}$
  Calculate $R_{res} = \min\Big\{\max(R, N_f), \max(N_f, P_R)\Big\}$
  Update $R_{res}$, $N_f$, $P_R$
```

Furthermore, let $n_{p,N}$, $n_{p,R}$, $n_{T1,N}$, $n_{T1,R}$, $n_{T2,N}$ and $n_{T2,R}$ represent the number of active PUs in N-SCH, active PUs in R-SCH, active type 1 CUs in N-SCH, active type 1 CUs in R-SCH, active type 2 CUs in N-SCH and active type 2 CUs in R-SCH respectively at any observation period. Ignoring channel or cell failure, the offloading process of type 1 and type 2 CUs is summarized in Algorithm 2. This is essential to the analyses presented in the next sections.

The performance of these algorithms was validated using independent Monte Carlo simulations to ensure that Algorithm 2 always produces expected outputs (i.e., priority-based channel access) for the range of users as long as channel access is required and terminates when no access is required. Similarly, Algorithm 2 provides reliable outcomes under stable conditions (i.e., provided that the total arrival rate of packets is less than the total departure rate in the system). For time complexity, we know that Algorithm 1 depends on the channel access request, hence its complexity is of order $O(N)$. For Algorithm 2, if the stability of the system is assumed, the time complexity depends on the set of inputs. To solve the complexity, we obtained the upper band and lower bound of the time complexity. The order of complexity is lower-bounded at $O(N)$ and upper-bounded at $O(N^2)$. Next, we describe the achievable offloading rate of both type 1 and type 2 CUs under the proposed scheme.

A. Achievable offloading rate

Without loss of generality, we considered a tagged edge server located at the origin o. The received signal to interference plus noise ratio (SINR) at the origin from any tagged type 1 CU $SINR_{\chi^1_k,t}$ is given in (1), located at the top of next page, where $P_c \in \{P_o, P_u\}$, $h_{a,b}$ is the channel gain between node a and b, σ^2 captures the thermal noise signal power, $o_p \in \{0,1\}$ captures the offloading decision parameter of each PU $\chi^p_i \in \Psi_p$ with transmission signal power P_p and $\mathbb{1}_{\{.\}}$ is the indicator function. Similarly, $||.||$ depicts the Euclidean distance, η is the path-loss exponent and $P_{c^!}$ is the received offloading power from other interfering type 1 CUs. We assumed that only one user can offload on a sub-channel at the same time, such that $P_{c^!} < P_c$ [22], [23].

The achievable offloading rate $R_{\chi_{k^1}}$ of a typical type 1 CU at time slot t is obtained from (1) as

$$R_{\chi^1_k,t} = \omega \log_2\Big(1 + SINR_{\chi^1_k,t}\Big), \tag{2}$$

where ω represents the channel bandwidth. For any typical type 2 CU, the inter-cell interference is only generated from other transmitting type 2 users within the considered frequency band, hence the SINR is given as,

$$SINR_{\chi^2_k,t} = \frac{P_o h_{\chi^2_k,o}}{\sigma^2 + \sum_{\chi^2_i \in \Psi^2_c \backslash \chi^2_k} \mathbb{1}_{\{o_c\}} P_{o^!} h_{\chi^2_i,o} ||\chi^2_i||^{-\eta}}, \tag{3}$$

$$SINR_{\chi_k^1,t} = \frac{P_c h_{\chi_k^1,o}}{\sigma^2 + \sum_{\chi_i^p \in \Psi_p} \mathbb{1}_{\{o_p\}} P_p h_{\chi_i^p,o} ||\chi_i^p||^{-\eta} + \sum_{\chi_i^1 \in \Psi_c^1 \setminus \chi_k^1} \mathbb{1}_{\{o_c\}} P_{c^!} h_{\chi_i^1,o} ||\chi_i^1||^{-\eta}}. \tag{1}$$

Algorithm 2: Offloading process for type 1 and 2 CUs

```
Initialization: Each user offloads at any observation
  period with probability between 0 and 1.
End Initialization
Input: $n_{p,N}$, $n_{p,R}$, $n_{T1,N}$, $n_{T1,R}$, $n_{T2,N}$ and $n_{T2,R}$
while channel access is required do
    if $n_{p,N} = 0$, $n_{p,R} = 0$ & $n_{T1,N} > 0$ then
        $\chi_i^1 \in \Psi_c^1$ offloads on N-SCH and R-SCH
          following the overlay channel access scheme;
    end
    else if $n_{p,N} > 0$, $n_{p,R} = 0$ & $n_{T1,N} > 0$ then
        $\chi_i^1 \in \Psi_c^1$ offloads on R-SCH following the
          overlay mode; $\chi_i^2 \in \Psi_c^2$ is blocked
    end
    else if $n_{p,N} > 0$, $n_{p,R} > 0$ & $n_{T1,N} > 0$ then
        Estimate $I_{PU}^n$
        For $I_{PU}^n < \gamma$:
        $\chi_i^1 \in \Psi_c^1$ offloads on N-SCH and R-SCH
          following the underlay channel access
          scheme; $\chi_i^2 \in \Psi_c^2$ is blocked
        end
    end
    else if $n_{p,N} = n_{p,R} = 0$, $n_{T1,N} = n_{T1,R} = 0$ &
      $n_{T2,N} > 0$ then
        $\chi_i^2 \in \Psi_c^2$ offloads on N-SCH and R-SCH
          following the overlay channel access scheme
    end
    else if $n_{p,N} = n_{p,R} = 0$, $n_{T1,N} > 0$, $n_{T1,R} = 0$ &
      $n_{T2,N} > 0$ then
        $\chi_i^2 \in \Psi_c^2$ offloads on R-SCH following the
          overlay channel access scheme
    end
    else if $n_{p,N} = n_{p,R} = 0$, $n_{T1,N} > 0$, $n_{T1,R} > 0$ &
      $n_{T2,N} > 0$ then
        $\chi_i^2 \in \Psi_c^2$ is blocked
    end
end
```

and its achievable transmission rate $R_{\chi_{k,2}}$ is expressed as

$$R_{\chi_k^2,t} = \omega \log_2 \Big(1 + SINR_{\chi_k^2,t}\Big), \tag{4}$$

where $P_{o^!} < P_o$. From (2) and (4), one may obtain the packet offloading time $t_{\chi_k,t}$ as

$$t_{\chi_k,t} = \begin{cases} \frac{d_{\chi_k}}{R_{\chi_k^1,t}} + \tau_{\chi_k}, & \text{if type 1 CU} \\ \frac{d_{\chi_k}}{R_{\chi_k^2,2}} + \tau_{\chi_k}, & \text{if type 2 CU,} \end{cases} \tag{5}$$

where d_{χ_k} is the input data size of any user χ_k and τ_{χ_k} represents the offloading time from the edge server to the cloud server when the data cannot be computed at the edge level. Owing to the assumed error-free communication channels between the edge and cloud servers, the transmission time between edge and cloud servers is assumed negligible, i.e., $\tau_{\chi_k} \approx 0$.

B. Processing time

Packet processing/computation time depicts the average time required to process a packet received from any test user χ_k. Let f^e and f^c represents the CPU-cycle frequency of edge and cloud servers (i.e. the processing capacity of edge and cloud servers) respectively, then the time required to process a data of size d_{χ_k} at the edge server is given as

$$T_e = \frac{c_{\chi_k}}{f^e}, \tag{6}$$

where $c_{\chi_k} \ll f^e \ll f^c$ is the required CPU cycles to process a task of size d_{χ_k}. Similarly, the time required to process any data of size d_{χ_k} at the cloud server is obtained as

$$T_c = \frac{c_{\chi_k}}{f^c}, \tag{7}$$

From (6) and (7), we know that the total processing time of any data of size d_{χ_k} can be given as

$$T_t = \varsigma_e T_e + \varsigma_c T_c, \forall \varsigma_e + \varsigma_c = 1, \tag{8}$$

where ς_e and ς_c are the probabilities that the packet is processed at the edge and cloud servers respectively. The total offloading latency – defined as the total time from the packet generation time to the time in which such a packet is successfully computed – can be obtained as

$$C_{total,t} = t_{\chi_k,t} + T_t. \tag{9}$$

The expression in (9) depends on $t_{\chi_k,t}$ which is difficult to obtain due to spatiotemporal correlation among interfering nodes. In the next section, we obtained an analysis for the average departure rate using the tools of SG and queueing theory. With this, we obtained an approximate analysis for the offloading time for any arbitrarily type 1 CU and type 2 CU.

IV. Analysis of average departure rate and offloading delay

The average departure rate is the rate at which generated packets are received at the edge's level. This depends on the average offloading success rate under each channel access scheme, obtained as the rate at which the received SINR exceeds the predefined threshold. This SINR is a random variable and is better characterized via distribution. We hence condition the process on the realization of point process $\Psi = \Psi_p \cup \Psi_c$ [23], [27], such that the conditional packet offloading success rates of type 1 CU and type 2 CU are generally obtained following

$$C_{1,t}^{\Psi} = P(SINR_{\chi_k^1,t} \geq \theta_1 | \Psi), \tag{10}$$

$$C_{2,t}^{\Psi} = P(SINR_{\chi_k^2,t} \geq \theta_2 | \Psi), \tag{11}$$

where θ_1 and θ_2 are the predefined SINR threshold for type 1and type 2 CUs respectively.

Assumption 1: To ensure tractability, we assumed that the packet departure rates are independent over different time slots, while the offloading powers among users are independent. This assumption follows the mean-field approximation [22], [23], [28]. Hence we assumed that the edge server experiences almost independent interference at each time slot.

Remarks: In [22], [23], [29], it was shown that the spatial correlation among the offloading power of interfering users is weak. Furthermore, when the full channel inversion power control scheme is adopted, the temporal SINR correlation has a negligible effect on the average departure rate. This approximation has also been validated in [22], [28], [29] and was further confirmed by the simulation results in Section VI of this paper. We hence drop the notation t.

A. Type 1 cognitive users

The average departure rate of a typical packet generated at a test type 1 user χ_k^1 is obtained as

$$\begin{aligned}\kappa_1 = p_q \bar{p_f} C_1^{\Psi,O} + \bar{p_q} p_m C_1^{\Psi,OI} + \bar{\rho}\beta[p_q p_f C_1^{\Psi,U} \\ + \bar{p_q} \bar{p_m} C_1^{\Psi,UI}],\end{aligned} \tag{12}$$

where $\bar{A} = 1 - A$. The parameter $C_1^{\Psi,O}$ represents the conditional success rate of type 1 CUs as in the conventional overlay channel access mode provided there was no misdetection, $C_1^{\Psi,U}$ is the conditional success rate of type 1 CUs as in the underlay channel access mode with a false alarm, $C_1^{\Psi,OI}$ and $C_1^{\Psi,UI}$ are the success rates in the overlay mode when there is misdetection and in the conventional underlay channel access mode when there is interference from PUs respectively, p_f and p_m represent the probability of false alarm and the probability of misdetection of PU signal by type 1 CU respectively, $p_q = P[o_p(t) = 0] = 1 - \alpha_p$, ρ is the penalty term due to periodic monitoring of the interference band in underlay mode and β is the type 1 CU switching rate between underlay and overlay modes. $\beta = 0$ indicates a fully or conventional overlay mode, while $\beta = 1$ indicates a fully underlay mode. $0 < \beta < 1$, however, indicates the hybrid model.

Next, we obtain analysis for each component of (12).

Lemma 1: The average departure rate of a tagged type 1 CUs χ_k^1 as in the conventional overlay channel access mode C_1^O can be obtained as the conditional offloading success probability following (10) as

$$\begin{aligned}C_1^{O,\Psi} &\overset{(a)}{=} P\left(\frac{P_o h_{\chi_k^1,o}}{\sigma^2 + \sum_{\chi_i^1 \in \Psi_c^1 \backslash \chi_k^1} \mathbb{1}_{\{o_c\}} P_{o^!} h_{\chi_i^1,o} ||\chi_i^1||^{-\eta}} > \theta_1 | \Psi\right) \\ &= \exp\left\{\frac{-\theta_1 r_o^{\eta} \sigma^2}{P_o}\right\} \prod_{\chi_i^1 \in \Psi_c^1 \backslash \chi_k^1} \left(\frac{o_{c,t}}{1 + \frac{\theta_1 r_o^{\eta}}{||\chi_i^1||^{\eta}} \frac{P_{o^!}}{P_o}} + 1 - o_{c,t}\right),\end{aligned} \tag{13}$$

where r_o is the distance between any tagged user and the edge server.

Proof: The proof is straightforward following the Rayleigh fading assumption on (a) with some algebraic manipulations [25], [27].

The exact derivation of the analysis presented in Lemma 1 is difficult to obtain. Thus we obtained its approximate using the b-th moment [30] as $\sigma^2 \rightarrow 0$. The b-th moment of Lemma 1 is given as

$$\begin{aligned}M_b &= E_{\chi_k^1}\{(C_1^{O,\Psi})^b\} \\ &= E_{r_o}\left\{\prod_{\chi_i^1 \in \Psi_c^1 \backslash \chi_k^1} \left(1 - \frac{E[o_{c,t}|\Psi]}{1 + \frac{||\chi_i^1||^{\eta}}{\theta_1} r_o^{-\eta} \frac{P_o}{P_{o^!}}}\right)^b\right\} \\ &\overset{(a)}{=} \exp\left\{ - \mu_1 \alpha_1 \theta_1^{\frac{2}{\eta}} r_o^2 \frac{2\pi^2}{\eta \sin\left(\frac{2\pi}{\eta}\right)} \sum_{k=1}^{\infty} \binom{b}{k} \binom{\delta - 1}{k - 1} a_1^k \right\}, \\ &\overset{(b)}{=} \exp\left\{ - \mu_1 \alpha_1 \theta_1^{\frac{2}{\eta}} r_o^2 \frac{2\pi^2}{\eta \sin\left(\frac{2\pi}{\eta}\right)} \frac{\Gamma(b + \delta)}{\Gamma(b)\Gamma(1 + \delta)} \right\},\end{aligned} \tag{14}$$

where $E[o_{c,t}|\Psi] = \lim_{t \rightarrow \infty} P(o_{c,t} = 1|\Psi)$ in (a) is the steady-state offloading probability of χ_k^1, $\delta = \frac{2}{\eta}$ and a_1 is the active probability of χ_k^1. For a special case of $a_1 = 1$, (b) is obtained. The resulting meta distribution [30] can be approximated with beta distribution as

$$P[C_1^{O,\Psi} \leq x] = I_x(\beta_1, \beta_2), \tag{15}$$

where $I_x(.,.)$ is generally known as the regularized incomplete beta function and

$$\begin{aligned}\beta_1 &= \frac{M_1(M_1 - M_2)}{M_2 - M_1^2}, \\ \beta_2 &= \frac{(M_1 - M_2)(1 - M_1)}{M_2 - M_1^2}.\end{aligned} \tag{16}$$

The parameters M_1 and M_2 represent the first and second moment of $C_1^{O,\Psi}$.

Similarly, the analysis for C_1^U, C_1^{OI} and C_1^{UI} in (12) can be obtained from (17), located at the top of next page.

Following Assumption 1, the closed-form expressions for C_1^U, C_1^{OI} and C_1^{UI} can be obtained as in Lemma 1. Substituting these into (12), the conditional average departure rate can be obtained for any test type 1 CU's packet. From this, the packet offloading time in (5) can be approximated as

$$t_{\chi_k} = E_{\Psi}\left\{\frac{1}{\kappa_1}\right\}. \tag{18}$$

Let Λ_E and Λ_C represent the offloading cost of a packet and the computation cost of such a packet per unit time at the edge server respectively, the total cost of offloading includes the cost of offloading and the cost of computation at the edge server[1]. Its analysis is straightforward from (9) and is given as

$$OC_{cost} = \Lambda_E * t_{\chi_k} + \Lambda_C * T_t. \tag{19}$$

B. Type 2 cognitive users

Note that any type 2 CU can only offload data to the edge server when all PUs and type 1 CUs within such a band are not offloading since offloading opportunity is unavailable to type

[1] The local computation cost depends only on the computation time and computation cost per unit time since packet offloading is not required in local computing.

$$\begin{aligned} C_1^{U,\Psi} &= P\Big(\frac{P_u h_{\chi_k^1}}{\sigma^2 + \sum_{\chi_i^1\in\Psi_c^1\backslash\chi_k^1} \mathbb{1}_{\{o_c\}} P_{u^!} h_{\chi_i^1} ||\chi_i^1||^{-\eta}} > \theta_1|\Psi\Big), \\ C_1^{OI,\Psi} &= P\Big(\frac{P_o h_{\chi_k^1}}{\sigma^2 + \sum_{\chi_i^p\in\Psi_p} \mathbb{1}_{\{o_p\}} P_p h_{\chi_i^p} ||\chi_i^p||^{-\eta} + \sum_{\chi_i^1\in\Psi_c^1\backslash\chi_k^1} \mathbb{1}_{\{o_c\}} P_{o^!} h_{\chi_i^1} ||\chi_i^1||^{-\eta}} > \theta_1|\Psi\Big), \\ C_1^{UI,\Psi} &= P\Big(\frac{P_u h_{\chi_k^1}}{\sigma^2 + \sum_{\chi_i^p\in\Psi_p} \mathbb{1}_{\{o_p\}} P_p h_{\chi_i^p} ||\chi_i^p||^{-\eta} + \sum_{\chi_i^1\in\Psi_c^1\backslash\chi_k^1} \mathbb{1}_{\{o_c\}} P_{u^!} h_{\chi_i^1} ||\chi_i^1||^{-\eta}} > \theta_1\Big). \end{aligned} \tag{17}$$

2 users when any primary or type 1 users are offloading; hence inter-cell interference is only generated from neighboring type 2 CUs. The average departure rate of a typical packet generated at a test type 2 user χ_k can be obtained as

$$\kappa_2 = (1-\alpha_h)C_2^{\Psi,O}, \forall \alpha_h = \alpha_p + \alpha_1. \tag{20}$$

Lemma 2: The average departure rate of any tagged type 2 CUs χ_k^2 can be obtained as the conditional offloading success probability following (11) as

$$\begin{aligned} C_2^{O,\Psi} &= P\left(\frac{P_o h_{\chi_k^2,o}}{\sigma^2 + \sum_{\chi_i^2\in\Psi_c^2\backslash\chi_k^2} \mathbb{1}_{\{o_d\}} P_{o^!} h_{\chi_i^2,o} ||\chi_i^2||^{-\eta}} > \theta_2|\Psi\right) \\ &= \exp\Big\{\frac{-\theta_2 r_o^\eta \sigma^2}{P_o}\Big\} \prod_{\chi_i^2\in\Psi_c^2\backslash\chi_k^2} \left(\frac{o_{d,t}}{1+\frac{\theta_2 r_o^\eta}{||\chi_i^2||^\eta}\frac{P_{o^!}}{P_o}} + 1 - o_{d,t}\right). \end{aligned} \tag{21}$$

Proof: From (11),

$$C_2^{\Psi,O} = \exp\Big(\frac{-\theta_2\sigma^2}{P_o}\Big)\mathcal{L}_{I_2}\Big(\frac{\theta_2}{P_o}\Big), \tag{22}$$

where $\mathcal{L}_{I_2}$ is the Laplace transform of interference from type 2 CUs except the test type 2 CU. The solution of $C_2^{\Psi,O}$ can be obtained similar to (14). The approximation also follows the same process as in Lemma 1 and is omitted to avoid unnecessary repetition.

If we substitute (21) into (20), we can obtain the average departure rate of packets belonging to a tagged type 2 user. From this, we obtained the packet offloading time.

C. Channel availability

Note that the active probability of each user a_1 depends on both the packet arrival rates (i.e., α_p, α_1 and α_2) and offloading decision parameters (i.e., o_p, o_c, o_d). This offloading decision parameters highly depend on channel availability rates. Thus, channel availability is characterized in this subsection following Algorithm 2. Recall that the dynamic channel reservation technique presented in the previous sections permits Type 1 CUs to switch to R-SCH when at least one PU arrives on the N-SCH to ensure offloading retainability of Type 1 CUs. Similarly, an active Type 2 CU can switch to R-SCH to ensure offloading retainability when at least one user belonging to higher priority classes arrives, while assuming negligible handover time. The channel availability indicator can thus be modeled through a discrete-time Markov chain (DTMC) model, where channel access (either in N-SCH or R-SCH) in each time slot is given to one class of users if there is no higher priority user on such a channel. The channel access rule in either the N-RCH or R-SCH is obtained as PU>Type 1 CU>Type 2 CU. It is worth noting that, channel reservations can enhance effective and efficient usage of the channel resources [31]–[34] and ensure that lower priority class users continue to offload with full offloading power on the unused part of the licensed channel to improve offloading throughput. Hence, channel reservation can improve channel availability for type 1 and type 2 users.

Without loss of generality, we assumed that each task is offloaded on a single sub-channel. Hence, the number of arriving tasks equals the number of needed cells or sub-channels. To characterize the channel availability indicator, we modeled the proposed multi-class channel access scheme using the DTMC, where the feasible states in the DTMC are represented as F with each state of the DTMC model corresponding to the multi-class channel access scheme given as $s = (n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}, n_{T2,R})$. The total number of unavailable sub-channels in the N-SCH and R-SCH for any typical state s is thus given as $B_N(s) = n_{p,N} + n_{T1,N} + n_{T2,N}$ and $B_R(s) = n_{p,R} + n_{T1,R} + n_{T2,R}$ respectively, while the total unavailable sub-channels are obtained as $B(s) = B_N(s) + B_R(s)$. From this, we know that the total number of available/vacant sub-channels in the state s is given as $M - B(s)$. Let the number of reserved sub-channels in the state s based on Algorithm 1 be given as $R(s)$, then the transition probability associated with different transition states can be obtained as a one-step transition matrix P. Using the transition probability matrix, the steady-state probabilities π_d can be obtained for each state $s \in F$.

If we assumed an event-centric environment where the system changes based on events (such as arrivals and departures) and not time, we can neglect the time notation for simplicity and focus on changes due to events. The transition rate matrix Q in Table I matches the continuous-time Markov chain (CTMC) model and can be used to obtain a uniformized DTMC [34], [35]. Let P represents the one-step transition probability matrix for the DTMC, while π_d and π_c represent the steady-state probability of the DTMC and CTMC respectively. By adopting uniformization (otherwise called randomization) technique, we know that the steady-state distributions π_d and π_c coincide [35], i.e.,

$$\pi_c(s) = \pi_d(s), \forall s \in F, \tag{23}$$

$$\sum_{s\in F}\pi_c(s) = \sum_{s\in F}\pi_d(s) = 1.$$

Given that $\pi(s)$ is the steady-state probability of being in state s, then the steady-state vector of the uniformized DTMC

TABLE I
TRANSITION RATE BETWEEN DIFFERENT STATES WITH $s = (n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}, n_{T2,R})$

Event	Destination state	Transition rate	Conditions
1. n number of PUs arrived. There are n vacant cells in N-SCH	$(n_{p,N}+n, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}, n_{T2,R})$	α_p	$B_N(s) < M - R(s)$; $B(s) < M$
2. n number of PUs arrived. n type 1 CUs perform handover to vacant cells in R-SCH	$(n_{p,N}+n, n_{T1,N}-n, n_{T2,N}, n_{p,R}, n_{T1,R}+n, n_{T2,R})$	α_p	$B_R(s) < R(s)$; $n_{p,R} = 0$
3. n number of PUs arrived. n type 1 CUs switch to underlay mode	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}+n, n_{T1,R}-n, n_{T2,R})$	α_p	$n_{T1,R} > 0$; $n_{T2,N} = n_{T2,R} = 0$
4. n number of PUs arrived. n type 2 CUs perform handover to vacant cells in R-SCH	$(n_{p,N}+n, n_{T1,N}, n_{T2,N}-n, n_{p,R}, n_{T1,R}, n_{T2,R}+n)$	α_p	$B_R(s) < R(s)$; $n_{p,R} = 0$ $n_{T1,N} = n_{T1,R} = 0$ $n_{T2,N} > 0$
5. n number of PUs arrived. n type 2 CUs are forced terminated in R-SCH	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}+n, n_{T1,R}, n_{T2,R}-n)$	α_p	$n_{T2,R} > 0$
6. n number of type 1 CUs arrived. There are n vacant cells in N-SCH	$(n_{p,N}, n_{T1,N}+n, n_{T2,N}, n_{p,R}, n_{T1,R}, n_{T2,R})$	α_1	$B_N(s) < M - R(s)$; $B(s) < M$; $n_{p,N} = 0$
7. n number of type 1 CUs arrived. n type 2 CUs perform handover to vacant cells in R-SCH	$(n_{p,N}, n_{T1,N}+n, n_{T2,N}-n, n_{p,R}, n_{T1,R}, n_{T2,R}+n)$	α_1	$B_R(s) < R(s)$; $n_{p,R} = 0$ $n_{T1,R} = 0$; $n_{T2,N} > 0$
8. n number of type 1 CUs arrived. n type 2 CUs are forced terminated in R-SCH	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}+n, n_{T2,R}-n)$	α_1	$n_{T2,R} > 0$
9. n type 2 CUs arrived. There are n vacant cells in N-SCH	$(n_{p,N}, n_{T1,N}, n_{T2,N}+n, n_{p,R}, n_{T1,R}, n_{T2,R})$	α_2	$B_N(s) < M - R(s)$; $B(s) < M$; $n_{p,N} = n_{T1,N} = 0$
10. n PUs depart from N-SCH^{++}	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}-n, n_{T1,R}, n_{T2,R})$	κ_p	$n_{p,N} = n_{p,R} > 0$
11. n PUs depart from N-SCH. No PU in R-SCH	$(n_{p,N}-n, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}, n_{T2,R})$	κ_p	$n_{p,N} > 0$; $n_{p,R} = 0$
12. n PUs depart from R-SCH.	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}-n, n_{T1,R}, n_{T2,R})$	κ_p	$n_{p,R} > 0$
13. n type 1 CUs depart from N-SCH^{++}	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}-n, n_{T2,R})$	κ_1	$n_{T1,N} = n_{T1,R} > 0$
14. n type 1 CUs depart from N-SCH. No type 1 CUs in R-SCH	$(n_{p,N}, n_{T1,N}-n, n_{T2,N}, n_{p,R}, n_{T1,R}, n_{T2,R})$	κ_1	$n_{T1,N} > 0$; $n_{T1,R} = 0$
15. n type 1 CUs depart from R-SCH.	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}-n, n_{T2,R})$	κ_1	$n_{T1,R} > 0$
16. n type 2 CUs depart from N-SCH^{++}	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}, n_{T2,R}-n)$	κ_2	$n_{T2,N} = n_{T2,R} > 0$
17. n type 2 CUs depart from N-SCH. No type 2 CUs in R-SCH	$(n_{p,N}, n_{T1,N}, n_{T2,N}-n, n_{p,R}, n_{T1,R}, n_{T2,R})$	κ_2	$n_{T2,N} > 0$; $n_{T2,R} = 0$
18. n type 2 CUs depart from R-SCH.	$(n_{p,N}, n_{T1,N}, n_{T2,N}, n_{p,R}, n_{T1,R}, n_{T2,R}-n)$	κ_2	$n_{T2,R} > 0$

++ – When PUs depart from N-SCH, active PUs in the R-SCH switch to the N-SCH. Type 1 or Type 2 CUs similarly switch from R-SCH to N-SCH when an appropriate sub-channel exists in N-SCH.

satisfies

$$\pi_d P = \pi_d, \qquad \pi_d 1 = 1. \tag{24}$$

From [31], the probability that a typical PU packet find an available sub-channel is given as

$$A_{PU} = 1 - \sum_{\substack{s\in F \\ B(s)=M;\ n_{T1,N}=n_{T1,R}=0; \\ n_{T2,N}=n_{T2,R}=0}} \pi(s). \tag{25}$$

Similarly, the probability that a typical type 1 CU finds an available sub-channel for overlay channel access is given as

$$A_{type1} = 1 - \sum_{\substack{s\in F \\ B(s)\leq M-R(s);\ n_{p,N}=n_{p,R}>0 \text{ OR} \\ B_N(s)=M;\ n_{p,N}=n_{p,R}=0;\ n_{T2,N}=n_{T2,R}=0}} \pi(s), \tag{26}$$

while the probability that a typical type 2 CU find an available sub-channel is given as

$$A_{type2} = 1 - \sum_{\substack{s\in F \\ B(s)\leq M-R(s);\ n_{p,N}=n_{p,R}>0 \text{ OR} \\ B(s)\leq M-R(s);\ n_{T1,N}=n_{T1,R}>0 \text{ OR} \\ B_N(s)=M;\ n_{p,N}=n_{p,R}=n_{T1,N}=n_{T1,R}=0}} \pi(s), \tag{27}$$

D. Offloading delay

Offloading delay depends on the channel availability period and is measured as the number of slots between the end of the slot in which a packet was generated at the users' level and the end of the slot in which such a packet was successfully received at the server. We assumed that packets belonging to the same class are offloaded following first come first serve. As previously mentioned, type 1 CUs packets are permitted to be offloaded anytime. Type 2 CUs' packets, however, are delay-tolerant and can only be offloaded when no delay-sensitive and PU traffic is present in the system. Also, type 2 packets in the system are interrupted at the beginning of each slot if any PU or type 1 CU packet arrived. Such an interrupted type 2 packet will only be re-offloaded in the next available slot. With this, PUs and type 1 CUs possess preemptive priority over type 2 CUs.

Since packets arrivals were considered to follow independent Bernoulli processes with inter-arrival times following geometric distributions, packets' departure rates provided in (12) and (22) can similarly be considered to follow independent general distributions. From this, we know that the offloading delay of any type 1 CU's packet can be expressed as

$$\Delta_1 = \frac{1-\alpha_1}{\kappa_1 - \alpha_1}. \tag{28}$$

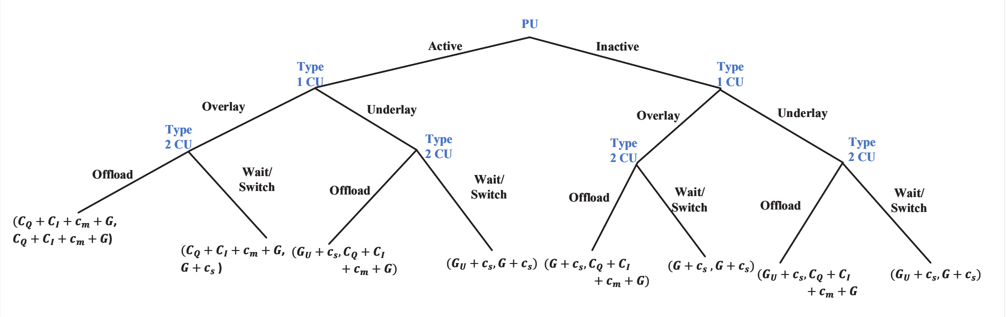
PU
Active
Inactive
Type 1 CU
Overlay
Underlay
Type 2 CU
Offload
Wait/ Switch
$(C_Q + C_I + c_m + G, C_Q + C_I + c_m + G)$
$(C_Q + C_I + c_m + G, G + c_s)$
$(G_U + c_s, C_Q + C_I + c_m + G)$
$(G_U + c_s, G + c_s)$
$(G + c_s, C_Q + C_I + c_m + G)$
$(G + c_s, G + c_s)$
$(G_U + c_s, C_Q + C_I + c_m + G$
$(G_U + c_s, G + c_s)$

Fig. 2. Game tree for the offloading cost.

For type 2 CUs, the arrival at the higher priority classes must be considered. Let the joint arrival rate at the higher priority classes be α_h and the average joint departure rate be κ_h, the offloading delay of any type 2 CU's packet is given as

$$\Delta_2 = \frac{1}{2}\Big(\frac{\kappa_{ef} V_{\alpha_2} + \alpha_2^2 V_{\kappa_{ef}}}{\alpha_2(1-\rho_{ef})} \\ +(1-\tau)\varsigma V_b + \kappa_{ef} - (1-\tau)(1-\tau\varsigma b)b\Big), \tag{29}$$

where $\kappa_{ef} = \frac{\tau}{\varsigma(1-\tau)}[\frac{1}{\kappa_2}(\frac{1}{\tau})-1]$ is the effective offloading time for preemptive repeat scheme, with τ being the probability that the channel is available for type 2 CU in two consecutive time slots. Furthermore, ς represents the fraction of time slots that the channel is available for type 2 user offloads. The parameter $\rho_{ef} < 1$ is the effective system load for a stable system, while b represents the busy period of high priority classes, which depends on α_h and κ_h. Finally, $V_{[*]}$ is the variance of $[*]$. The expression in (29) is derived following the analysis of the repeat after interruption Geo/G/1 queueing system. Its proof is straightforward from [36] and is omitted for brevity.

V. Game theory technique for offloading cost modeling

In this section, we obtain the offloading cost Λ_E given in (19) through the adoption of the game theory approach via the mixed strategy Nash equilibrium [37], [38]. Let C_Q and C_I represent the penalty for violating the offloading queueing constraints and the penalty for violating power control requirements respectively, while G and c_s represent the cost of offloading on a channel/sub-channel including the cost of channel sensing and cost of switching channels or waiting for appropriate offloading opportunity respectively. Similarly, let c_m and G_U depict the cost of violating the offloading channel access requirement and cost of offloading under appropriate power control scheme respectively with $G_U < G$, then the offloading cost of any generated packet at any tagged user can be formulated through game constraints given as

$$C_Q > C_I > G > c_s; c_m > G; G > G_U. \tag{30}$$

The condition $C_Q > C_I$ is necessary to ensure that violation of queueing constraints is discouraged, i.e., type 2 users do not pretend to be type 1 users, while the constraint $C_I > G$ ensures that penalty for violating the power control requirement is more than the cost of actual offloading - a parameter that is also greater than the cost of waiting or switching channel access mode. To minimize cost, a user will prefer to switch channels or channel access modes to violating offloading constraints. These constraints ensure that the cost of violating offloading channel access requirement is more than the cost of actual offloading thereby encouraging users to respect the channel access requirement. The game tree for the proposed offloading cost method is given in Fig. 2 and summarized in Table II.

From Table II, it is straightforward to obtain the upper and lower offloading costs. The upper offloading cost is obtained when the offloading cost is maximum. In such a case, the offloading process is achieved while all the required channel access constraints are violated. For instance, when at least one PU is active, a typical type 1 CU is expected to switch to R-SCH or offload its packet following the underlay channel access mode. A channel access constraint is violated when such a type 1 CU continues to offload in the overlay channel access mode. The situation is made worse when a type 2 CU also continues to offload in such a case. This increases the level of interference in the network thus increasing the cost of offloading to the maximum. Similarly, the lower offloading cost is obtained when the offloading cost is minimum. Such a case is obtained when all the constraints are satisfied (See Fig. 2). The offloading cost constraints are properly formulated through the game theory to ensure that users obey the offloading requirements thereby offloading with minimum offloading cost. When the offloading requirements are respected, packet offloading and processing times can be significantly reduced. This will ensure that the QoE requirements of essential and delay-tolerant applications are satisfied. We will demonstrate the performance of the proposed scheme in Section VI.

VI. Results and simulation

In this section, we evaluated the performance of the proposed channel access mechanism and validated the performance through independent Monte Carlo simulations. Except otherwise stated, the following parameters were used for simulations: $P_o = -32$ dB, $P_u = -36$ dB, $\sigma^2 = 10^{-9}$, $p_q = 0.9$, $p_f = 0.2$, $p_m = 0.2$, $\beta = 0.25$, $\rho = 0.2\beta$, $\theta_1 = \theta_2 = 9.5$ dB, $\alpha_1 = 0.15$, $\alpha_2 = 0.2$, $C_{\chi_k} = 1$ CPU cycle per task, $f^e = 10$ computations per time slot and $f^c = 1000$ computations per time slot.

TABLE II
PAYOFF MATRIX FOR THE COST OF OFFLOADING

	Active PU		Inactive PU	
Type 1 CU, Type 2 CU	Offload	Wait	Offload	Wait
Overlay	$C_Q + C_I + c_m + G,$ $G + C_Q + C_I + c_m$	$C_Q + C_I + c_m + G,$ $G + c_s$	$G + c_s,$ $G + C_Q + C_I + c_m$	$G + c_s,$ $G + c_s$
Underlay	$G_U + c_s,$ $G + c_Q + C_I + c_m$	$G_U + c_s,$ $G + c_s$	$G_U + c_s,$ $G + C_Q + C_I + c_m$	$G_U + c_s,$ $G + c_s$

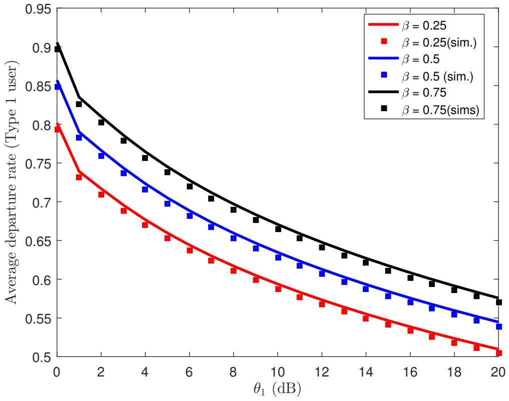

Fig. 3. Average departure rate for type 1 CUs, $\alpha_p = 0.1$.

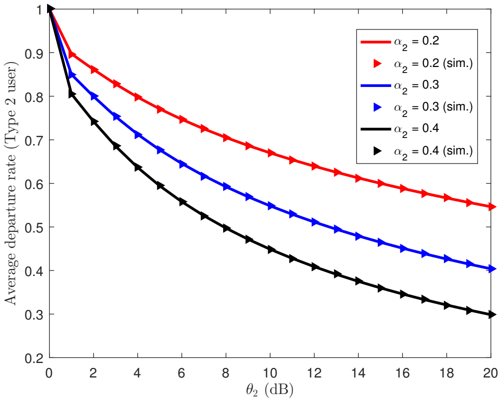

Fig. 4. Average departure rate of type 2 CUs, $p_m = p_f = 0$.

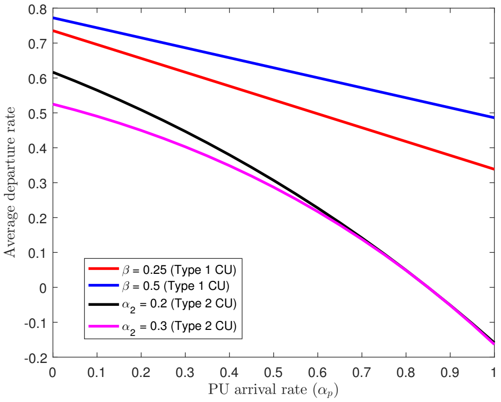

Fig. 5. Average departure rate.

A. Average departure rate and offloading delay

The average departure rate depends on the SINR threshold for each class of user. As shown in Fig. 3, the departure rate reduces with θ_1, while a frequent switching rate between modes by type 1 CUs further reduces the departure rate of type 1 CUs' packets owing to the reduced offloading power when offloading in the underlay mode (i.e. when sharing channel with at least on PU). Notwithstanding this, essential packets can experience better departure rates under the proposed scheme, since the packet departure rates are observed to be lower than the arrival rate even at a higher SINR threshold. Similar results were obtained for type 2 users as shown in Fig. 4 at $\alpha_p = 0.1$ and $\alpha_1 = 0.15$. As expected, the higher arrival of packets further reduces the departure rate. The lower departure rate at $\alpha_2 = 0.4$ shows the impact of arrivals of the higher priority class packets, which makes the channel unavailable for type 2 CU offloading.

The impacts of PU packet arrivals on the average departure rates of both type 1 and type 2 CUs are presented in Fig. 5. The departure rate of type 1 CU decreases as the PU arrival rate increases. This is necessary since a higher arrival of PU packets causes type 1 CUs to switch often to underlay channel access mode, which reduces the overall throughput. Similarly, type 2 CUs receive lower offloading opportunities when PU and type 1 CU packets are generated hence the justification for the observed lower departure rate.

The offloading delay of type 1 users is presented in Fig. 6. This metric increases with an increase in packets arrival rate since more packets will have to wait for offloading opportunity when higher number of packets are generated at the same time. Interestingly, the offloading delay of CUs increases with activities of PUs as also suggested from Algorithm 2. With an increase in the required SINR threshold for successful reception of type 2 users' packets at the edge server, offloading delay increases provided that other parameters remain unchanged. Also, an increase in the arrival rate of type 2 packets as shown in Fig. 7 further increases the offloading delay of type 2 packets, since the average departure rate of type 2 packets depends on the departure rates at the higher priority queues. Interestingly, a lesser delay is experienced at a reduced SINR threshold.

The offloading delay is also shown to be affected by the probability of false alarm in Fig. 8. An increase in the false alarm rate in the system implies that type 1 CUs offload in the underlay channel access mode with lower throughput instead of the overlay channel access mode that enhances higher

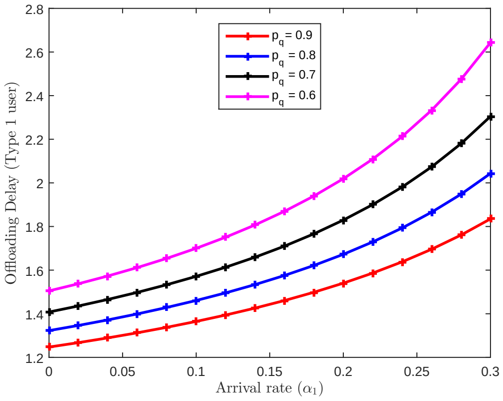

Fig. 6. Offloading delay experienced by type 1 CUs.

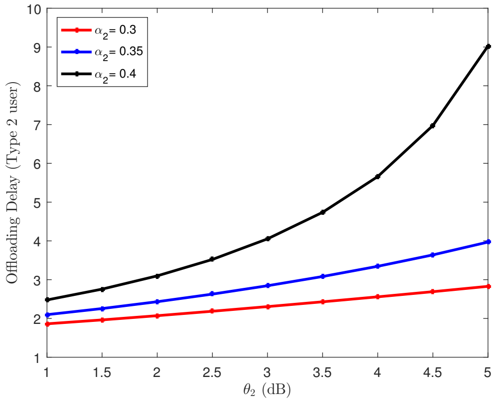

Fig. 7. Offloading delay for type 2 packets.

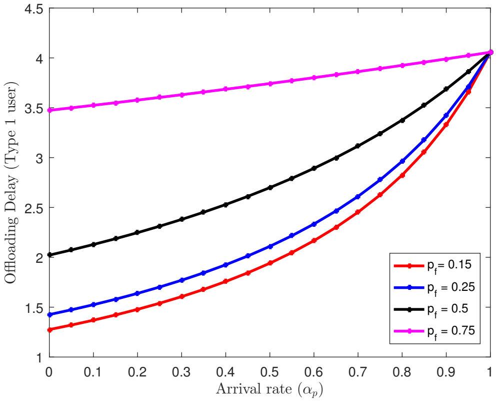

Fig. 8. Offloading delay.

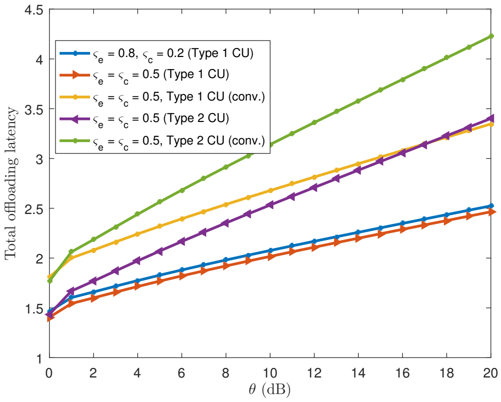

Fig. 9. Packet offloading latency, $d_{\chi_k} = 2$.

overall throughput. It is then important to ensure accurate detection of PU signals to ensure effective offloading of packets to the edge server.

B. Comparision with existing approach

The total offloading latency captures both the offloading and computation delay. As shown in Fig. 9, the total offloading latency depends on the fraction of packets computed at the edge and cloud servers. The total offloading latency increases when the fraction of packets computed at the cloud server increases under the assumption that the transmission latency between the edge and the cloud servers is negligible. For type 2 CU, the increase in the total offloading latency is due to the activities of PUs and type 1 CU. Reduced offloading latency is achieved for type 2 users with lower arrival rates of higher priority packets. To demonstrate the performance of the proposed channel access scheme over the existing conventional priority-based schemes, we compared the performance of the existing schemes [5]–[7] for type 1 and type 2 users with the proposed scheme. As presented in Fig. 9, the proposed scheme achieves lower total offloading latencies for both type 1 and type 2 CUs. This demonstrates the ability of the proposed scheme to satisfy lower priority users' channel access demands in the presence of higher-priority users as described in Algorithm 2.

Finally, the offloading cost of type 1 and type 2 packets is investigated in Fig. 10. The presented result was achieved at $C_Q = 6.5$, $C_I = 6$, $G = 2$, $c_m = 2.5$, $c_s = 1.5$, $G_U = 1.5$. The computation costs at the edge server and local CPU were set to 1, while the capacity of the local CPU was set to be 0.03 computation per slot. The cost is upper bounded when there is sensing inaccuracy which resulted in higher interference hence a lower delivery rate in the network. The offloading cost is lower bounded when the sensing is more accurate. Since there is lower available computation capacity at the local CPU, offloading to the edge server can improve the computation experience of CUs. This performance can be improved further with a more reliable channel sensing technique. When compared with the conventional approach, the proposed approach showed a reduced offloading cost as in Fig. 11.

Generally, an increase in the average departure rate reduces the average transmission time and offloading cost, while a decrease in offloading delay will improve users' QoE. The proposed channel access mechanism can ensure that the offloading process of essential packets is prioritized while also improving the offloading efficiency for all users.

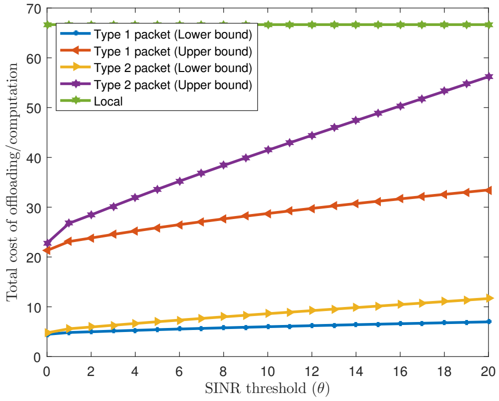

Fig. 10. Packet offloading cost, $d_{\chi_k} = 2$.

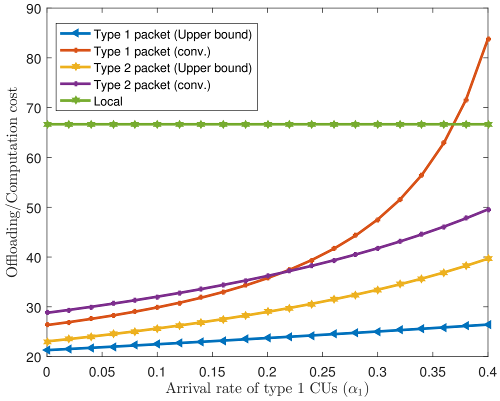

Fig. 11. Offloading/Computation cost, $d_{\chi_k} = 2$.

VII. Conclusion and Future Work

Tremendous emerging issues are associated with the deployment of IoT-based systems in various environments. For instance, a large amount of data traffic is expected to be produced in IoT industries through the deployment of extensive IoT-based wireless area networks. This, of course, will require efficient radio resource management. This paper presents a multi-class delay-constrained channel access mechanism in cognitive EdgeIoT networks. Essential IoT devices were granted priority over delay-tolerant users. We showed that such a multi-class channel access mechanism is required to meet the spectrum access demands of numerous EdgeIoT users. The proposed mechanism improves the channel access experience of essential IoT users, while also ensuring that channel access demands of delay-tolerant IoT users are met with limited offloading delay.

The proposed approach considered two classes of cognitive users – essentials and delay-tolerant users. This proposed approach can be extended to accommodate more classes of users with different priority levels, where each of these classes seeks to satisfy its offloading requirement on PUs licensed channels. To improve the computation model, the effects of parallel computing in multi-server systems may be captured in the computation cost analysis. The proposed multi-class channel access scheme provides an insight into how the QoE of essentials applications can be met in cognitive edge-based IoT networks. This is important towards the deployment of next-generation networks such as 5G and beyond 5G, where multiple users are expected to compete for channel resources to satisfy their offloading requirement when accessing computation opportunities at the edge server. We believe that the multi-class channel access scheme presented in this paper can be significant towards improving channel access demands in more complex emerging large-scale communication systems such as multi-hop reconfigurable intelligent surfaces-based communications systems [39], [40].

References

[1] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, “Relaying techniques based outage analysis for mobile users in cognitive radio networks,” in IEEE Vehicular Technology Conference, Antwerp, May 2020, pp. 1–5.

[2] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, “Outage and throughput analysis of cognitive users in underlay cognitive radio networks with handover,” IEEE Access, Accepted, Oct. 2020.

[3] R. Zhao, X. Wang, J. Xia, and L. Fan, “Deep reinforcement learning based mobile edge computing for intelligent internet of things,” Physical Communication, vol. 43, pp. 101184, Aug. 2020.

[4] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” IEEE Transactions on Networking, vol. 24, no. 5, pp. 2795–2808, Oct. 2015.

[5] A. Naouri, H. Wu, N. A. Nouri, S. Dhelim, and H. Ning, “A Novel Framework for Mobile-Edge Computing by Optimizing Task Offloading,” IEEE Internet of Things Journal, vol. 8, no. 16, pp. 13065–13076, Aug. 2021.

[6] S. Luo, H. Li, Z. Wen, B. Qian, G. Morgan, A. Longo, O. Ranjan, and R. Ranjan, “Blockchain-Based Task Offloading in Drone-Aided Mobile Edge Computing,” IEEE Network, vol. 35, no. 1, pp. 124–129, Feb. 2021.

[7] S. Li, N. Zhang, S. Lin, L. Kong, A. Katangur, M. Khan, M. Ni, and G. Zhu, “Joint admission control and resource allocation in edge computing for internet of things,” IEEE Network, vol. 32, no. 1, pp. 72–79, Jan. 2018.

[8] S. Yu, X. Wang, and R. Langar, “Computation offloading for mobile edge computing: A deep learning approach,” in IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications, Montreal, Oct. 2017, pp. 1–6.

[9] Z. Ning, P. Dong, X. Kong, and F. Xia, “A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things,” IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4804–4814, Sept. 2018.

[10] M. Deng, H. Tian, and X. Lyu, “Adaptive sequential offloading game for multi-cell mobile edge computing,” in IEEE International Conference on Telecommunications, Thessaloniki, May 2016, pp. 1–5.

[11] S. Raza, W. Liu, M. Ahmed, M. R. Anwar, M. A. Mirza, Q. Sun, and S. Wang, “An efficient task offloading scheme in vehicular edge computing,” Journal of Cloud Computing, vol. 9, pp. 1–14, Dec. 2020.

[12] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, “Optimal delay constrained offloading for vehicular edge computing networks,” in IEEE International Conference on Communications, Paris, May 2017, pp. 1–6.

[13] J. Wang, W. Wu, Z. Liao, A. K. Sangaiah, and R. S. Sherratt, “An energy-efficient off-loading scheme for low latency in collaborative edge computing,” IEEE Access, vol. 7, pp. 149182–149190, Oct. 2019.

[14] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, “Performance guaranteed computation offloading for mobile-edge cloud computing,” IEEE Wireless Communications Letters, vol. 6, no. 6, pp. 774–777, Aug. 2017.

[15] M. Chen, and Y. Hao, “Task offloading for mobile edge computing in software defined ultra-dense network,” IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[16] H. Guo, and J. Liu, “Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks,” IEEE Transactions on Vehicular Technology, vol. 67, no. 5, pp. 4514–4526, Jan. 2018.

[17] X. Lyu, H. Tian, L. Jiang, A. Vinel, S. Maharjan, S. Gjessing, and Y. Zhang, “Selective offloading in mobile edge computing for the green internet of things,” IEEE Network, vol. 32, no. 1, pp. 54–60, Jan. 2018.

[18] G. Hu, Y. Jia, and Z. Chen, "Multi-user computation offloading with d2d for mobile edge computing," in IEEE Global Communications Conference, Abu Dhabi, Dec. 2018, pp. 1–6.

[19] B. Gu, X. Zhang, Z. Lin, and M. Alazab, "Deep multiagent reinforcement-learning-based resource allocation for Internet of controllable things," IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3066–3074, Mar. 2021.

[20] F. Li, H. Yao, J. Du, C. Jiang, and Y. Qian, "Stackelberg game-based computation offloading in social and cognitive industrial Internet of Things," IEEE Transactions on Industrial Informatics, vol. 16, no. 8, pp. 5444–5455, Aug. 2020.

[21] J. Zhang, X. Hu, Z. Ning, E. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2633–2645, Dec. 2017.

[22] M. Gharbieh, H. ElSawy, A. Bader, and M. S. Alouini, "Spatiotemporal stochastic modeling of IoT enabled cellular networks: Scalability and stability analysis," IEEE Transactions on Communications, vol. 65, no. 8, pp. 3585–3600, May 2017.

[23] H. ElSawy, and E. Hossain, "On stochastic geometry modeling of cellular uplink transmission with truncated channel inversion power control," IEEE Transactions on Wireless Communications, vol. 13, no. 8, pp. 4454–4469, April 2014.

[24] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, "Interference characterization in underlay cognitive networks with intra-network and inter-network dependence," IEEE Transactions on Mobile Computing, May 2020, DOI: 10.1109/TMC.2020.2993408.

[25] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, "Spatiotemporal Characterization of Users' Experience in Massive Cognitive Radio Networks," IEEE Access, vol. 8, pp. 57114–57125, Mar. 2020.

[26] Y. Xing, C. N. Mathur, M. A. Haleem, R. Chandramouli, and K. P. Subbalakshmi, "Dynamic spectrum access with QoS and interference temperature constraints," IEEE Transactions on mobile computing, vol. 6, no. 4, pp. 423–433, April 2007.

[27] H. H. Yang, and T. Q. Quek, "Spatio-temporal analysis for SINR coverage in small cell networks," IEEE Transactions on Communications, vol. 67, no. 8, pp. 5520–5531, Aug. 2019.

[28] H. H. Yang, A. Arafa, T. Q. Quek, and H. V. Poor, "Optimizing information freshness in wireless networks: A stochastic geometry approach," IEEE Transactions on Mobile Computing, vol. 20, no. 6, pp. 2269–2280, Jun. 2021.

[29] G. Chisci, H. ElSawy, A. Conti, M. S. Alouini, and M. Z. Win, "Uncoordinated massive wireless networks: Spatiotemporal models and multiaccess strategies," IEEE/ACM Transactions on Networking, vol. 27, no. 3, pp. 918–931, Jun. 2019.

[30] M. Haenggi, "The meta distribution of the SIR in Poisson bipolar and cellular networks," IEEE Transactions Wireless Communications, vol. 15, no. 4, pp. 2577–2589, Apr. 2016.

[31] I. A. Balapuwaduge, F. Y. Li, and V. Pla, "Dynamic spectrum reservation for CR networks in the presence of channel failures: Channel allocation and reliability analysis," IEEE Transactions on Wireless Communications, vol. 17, no. 2, pp. 882–898, Nov. 2017.

[32] G. Abbas, Z. H. Abbas, M. Waqas, and A. K. Hassan, "Spectrum utilization efficiency in the cognitive radio enabled 5G-based IoT," Journal of Network and Computer Applications, vol. 164, pp. 102686, Aug. 2020.

[33] G. Abbas, Z. H. Abbas, T. Baker, and M. Waqas, "Spectrum efficiency in CRNs using hybrid dynamic channel reservation and enhanced dynamic spectrum access," Ad Hoc Networks, vol. 107, pp. 102246, Oct. 2020.

[34] R. Kulshrestha, "Channel allocation and ultra-reliable communication in CRNs with heterogeneous traffic and retrials: A dependability theory-based analysis," Computer Communications, vol. 158, pp. 51–63, May 2020.

[35] N. M. van Dijk, S. P. van Brummelen, and R.J. Boucherie, "Uniformization: Basics, extensions and applications," Performance evaluation, vol. 118, pp. 8-32, Feb. 2018.

[36] J. Walraevens, D. Fiems, and H. Bruneel, "Performance analysis of priority queueing systems in discrete time," in Network Performance Engineering, Berlin, Germany: Springer, 2011, pp. 203–232.

[37] Y. Tan, S. Sengupta, and K. P. Subbalakshmi, "Primary user emulation attack in dynamic spectrum access networks: a game-theoretic approach," IET Communications, vol. 6, no. 8, pp. 964–973, May 2012.

[38] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa,"Malicious users control and management in cognitive radio networks with priority queues," in IEEE Vehicular Technology Conference, Victoria, pp. 1–7, Nov. 2020.

[39] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," IEEE Transactions on Wireless Communications, vol. 18, no. 8, pp. 4157–4170, Jun. 2019.

[40] C. Huang, Z. Yang, G. C. Alexandropoulos, K. Xiong, L. Wei, C. Yuen, Z. Zhang, and M. Debbah, "Multi-hop RIS-empowered terahertz communications: A DRL-based hybrid beamforming design," IEEE Journal on Selected Areas in Communications, vol. 39, vol. 6, pp. 1663–1677, Apr. 2021.

S.D. Okegbile received B.Tech (Hons.) degree in Computer Engineering (First class division) from Ladoke Akintola University of Technology, Ogbomoso, Nigeria, in 2011 and M.Sc. degree in Computer Science (Distinction) from Obafemi Awolowo University, Ile-Ife, Nigeria, in 2016. He also received Ph.D. degree in Computer Engineering from the University of Pretoria, South Africa in 2021. His research interests are in the area of pervasive and mobile computing which includes various interesting topics on cognitive radio networks, internet of things and wireless sensor networks.

B. T. Maharaj received the Ph.D. degree in wireless communications from the University of Pretoria, South Africa. He is currently a Professor and holds the position of Sentech Chair of Broadband Wireless Multimedia Communications with the Department of Electrical, Electronic, and Computer Engineering, University of Pretoria. His research interests are in MIMO channel modelling, OFDM-MIMO systems, and cognitive radio for rural broadband.

Attahiru S. Alfa is Professor Emeritus at the University of Manitoba, Department of Electrical and Computer Engineering and Extraordinary Professor at the University of Pretoria, Department of Electrical, Electronic and Computer Engineering. Dr. Alfa's most recent research focus covers age of information, wireless sensor networks, cognitive radio networks, network restoration tools for wireless sensor networks, and the role of 5G on IoT, with specific interest in the mathematical modeling of those systems. His general research covers, but not limited to, the following areas: queueing theory and applications, optimization, performance analysis and resource allocation in telecommunication systems, modeling of communication networks, analysis of cognitive radio networks, modeling and analysis of wireless sensor networks, and smart cities. Some of his previous works include developing efficient decoding algorithms for LDPC codes, channel modeling, traffic estimation for the Internet, and cross layer analysis. Dr. Alfa also works in the application of queueing theory to other areas such as transportation systems, manufacturing systems and healthcare systems. He has authored two books, "Queueing Theory for Telecommunications: Discrete Time Modelling of a Single Node System", published by Springer in 2010, and "Applied Discrete-Time Queue" published in 2015, also by Springer, as a second edition of the first book; and also co-authored (with Dr. Ibrahim) a book titled: "Optimization Methods for User Admissions and Radio Resource Allocation for Multicasting over High Altitude Platforms" published by River Publishers in 2019.