Modeling and Analysis of Latencies in Multi-User, Multi-RAT Edge Computing

Rushabha Balaji
rubalaji@ucsd.edu
Electrical and Computer Engineering,
University of California San Diego
La Jolla, California, USA

Neelesh B. Mehta
nbmehta@iisc.ac.in
ECE,
Indian Institute of Science
Bengaluru, India

Chandramani Singh
chandra@iisc.ac.in
Electronic Systems Engineering,
Indian Institute of Science
Bengaluru, India

ABSTRACT

Multi-access edge computing enables resource-constrained handsets to offload their tasks to edge servers through multiple radio access technologies (RATs). However, RATs such as cellular and wireless local area network (WLAN) have different access protocols, throughputs, and latencies. We present novel models for accurate analysis of the average task computation delay, accounting for the physical and media access control layer throughputs and latencies. Our analysis considers the uplink latency encountered in offloading tasks to edge servers and the downlink latency in downloading completed tasks from the server to the users. It accounts for the contention-based channel access in a WLAN, in which the number of users that contend is a random process, and the orthogonal channelization based access of a cellular network. We show that WLAN contention delays and their variability have a significant impact on the overall delay. Through a tractable probabilistic offloading policy, we bring out the trade-offs between choosing the different RATs. We also benchmark the performance of this policy against an upper confidence bound (UCB)-based dynamic offloading policy.

CCS CONCEPTS

• **Networks** → *Network performance modeling*; **Network performance analysis**; **Wireless local area networks**; • **Mathematics of computing** → **Renewal theory**; **Markov processes**.

KEYWORDS

Multi-access edge computing, WLAN, cellular network, renewal theory, fixed-point analysis

ACM Reference Format:
Rushabha Balaji, Neelesh B. Mehta, and Chandramani Singh. 2023. Modeling and Analysis of Latencies in Multi-User, Multi-RAT Edge Computing. In *Proceedings of the 19th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '23), October 30-November 3, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3616391.3622761

1 INTRODUCTION

Multi-access edge computing (MEC) is an emerging paradigm for next generation communication systems. MEC provides a solution to meet the ever increasing demands of low latency and increased computation power by advanced applications such as augmented reality (AR) [1] and virtual reality (VR) [6]. MEC brings servers to the edge of the Internet, thereby saving on Internet's data transportation resources and decreasing network latency. Energy-constrained mobile devices can now offload their tasks to MEC servers, which have better storage and computation capabilities.

Q2SWinet '23, October 30-November 3, 2023, Montreal, QC, Canada
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0368-3/23/10.
https://doi.org/10.1145/3616391.3622761

Today's mobile devices can access more than one MEC server through different radio access technologies (RATs), such as wireless local area network (WLAN), 4th generation long-term evolution (LTE), and 5th generation new-radio (5G-NR) [7]. MEC performance over these RATs depends on their physical (PHY) and multiple access control (MAC) layer technologies. Different RATs employ different PHY and MAC layer technologies. For example, LTE and 5G-NR employ orthogonal frequency division multiplexing (OFDM) that offers a dedicated subchannel to the base station (BS) to every user admitted to the cellular network. On the other hand, legacy WLAN uses an IEEE 802.11 based random-access method in which the users and the access point (AP) contend to access the channel.

We analyze the performance and design a probabilistic MEC offloading algorithm in the presence of two MEC servers attached to a WLAN AP and to a cellular network BS, respectively. This warrants joint modeling of the MEC applications, the WLAN and the cellular network.

1.1 Related Literature

The literature on data offloading and network selection can be broadly classified into single-user and multiple-user models.

1.1.1 Single-User Models. In [8], a cellular network integrated with a single MEC server is considered. The transmit power of the handset and the computational resources at the MEC server are adapted to the channel variations. However, no local computation is considered. In [12], interface selection between a WLAN and a cellular network is considered. CPU resource allocation under latency and queue stability constraints is considered for a heterogeneous task model in which separate queues are used for each type of task. In [17], the WLAN is assumed to be the preferred RAT but its availability is intermittent. The packet computation delays and energy are optimized, but MEC is not considered. In the above works, MAC contention and delays do not arise since there is only one user in the system.

1.1.2 Multiple-User Models. For a WLAN equipped with multiple MEC servers, the energy incurred in allocating central processing unit (CPU) resources with task inter-dependencies is minimized in [4]. In [15], energy minimization under delay constraints for partial task offloading to a WLAN or a cellular network is considered.

Here the MAC contention and associated delays are not modeled. In [18], allocation of computing resources among multiple MEC servers in a cellular network with a fixed number of orthogonal channels is considered. Energy harvesting along with data offloading to the WLAN is considered in [14]. However, the MAC-level contention is ignored here also. In [13], data offloading to a WLAN and a cellular network for voice and video users is considered. While the MAC contention process is modeled, the average delay is not analyzed. More recently, deep learning and reinforcement learning (RL) techniques for RAT selection have been studied in [9].

1.2 Contributions

We consider an MEC system with multiple users who can compute their tasks locally or can offload these to the edge servers through either a WLAN or a cellular network. Each offloaded task is sent to the AP or to the BS as a packet and the processed data is also returned to the user as a packet. Task offloading results in a randomly varying number of users in the WLAN and the cellular network. Further, the uplink and downlink transmissions are coupled. We make several novel, well justified assumptions and employ Markov renewal theory to develop simplified yet accurate models.

We model the transmission times, processing times, queuing delays, and contention (if present) at the physical and MAC layers of the RATs. Our WLAN model with a varying number of users accurately captures the contention delays faced by the users and the AP. It is more comprehensive than those in [4, 14, 15], which either ignore the AP contention and queuing delays or simplistically model the WLAN as a network with a fixed number of channels.

We characterize the average latency of the probabilistic offloading policy. This tractable policy offers valuable insights into the trade-offs between different RATs. Our results show that ignoring MAC contention delays, as often done in the MEC literature, can markedly underestimate the average delay. This can result in suboptimal RAT selection for task offloading.

To benchmark the probabilistic offloading policy, we also consider a distributed, dynamic RAT selection policy based on the *upper confidence bound* (UCB) [2]. We show that the two policies have a similar performance. This leads us to believe that the dynamic offloading policy learns the optimal offloading probabilities.

1.2.1 Outline. We present the system model and the probabilistic task offloading policy in Section 2. We analyze the average latency seen by a user in Section 3. We introduce the UCB-based offloading policy in Section 4. Numerical results are presented in Section 5, and our conclusions follow in Section 6.

2 SYSTEM MODEL

We consider a system consisting of N users, a WLAN and a cellular network. Let $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$ denote the set of users. All the users run applications that generate identical tasks. Each task requires L_{UL} bits to be processed, and has processing density d cycles/bit, i.e., it takes d CPU cycles to process each bit [12]. The system evolves in discrete time steps of duration δ sec, called *slots*, i.e., all packet transmissions, task executions, and other MAC activities commence at the slot boundaries.

The users can either locally compute their tasks or can offload those to the AP of the WLAN or to the BS of the cellular network, both of which are connected to MEC servers as shown in Fig. 1. If a user chooses to offload a task, it packages the task into a packet of size L_{UL} bits. It sends the packet either to the AP or to the BS who then forwards it to the respective MEC server. After executing the task, the MEC server packages the result into a packet of size L_{DL} bits and sends the packet to the AP or to the BS, as applicable, which then sends it to the concerned user.

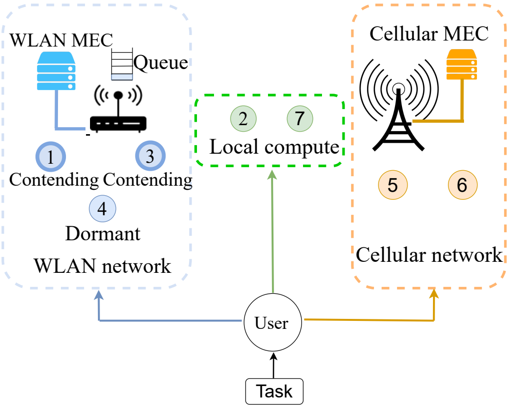


Figure 1: Illustration of the WLAN, cellular, and local networks with $N = 7$ users. In the cellular network, traffic is routed from BS to its MEC server. In the WLAN, traffic is routed through the AP to its MEC server. A user in the WLAN is either contending to transmit its packet to the AP or is dormant and is waiting for the AP to transmit its processed payload back.

We assume that each user's application generates a new task on receiving the result of the previous task. This corresponds to the users operating in the *saturated regime* but not having task queues. Below, we discuss local task computation and task offloading to the WLAN and the cellular network. In particular, we emphasize on the interplay of task offloading and the IEEE 802.11 MAC protocol.

2.1 Local Computation

We assume that the users' local compute devices have a CPU frequency f_{u} cycles/sec. Therefore, local computation of any task incurs a computation time $T_{\mathrm{loc}} = \frac{L_{\mathrm{UL}} d}{f_{\mathrm{u}}}$ sec. We use the term *local network* to refer to the set of users engaged in local computation.

2.2 Offloading to WLAN

The users who choose to offload their tasks to the WLAN and the AP use the IEEE 802.11 distributed coordination function (DCF) MAC protocol for channel access. A user transmits the packetized task to the AP, which forwards the packet to its MEC server. The server returns the processed packet to the AP, which queues it and transmits it back to the same user.

IEEE 802.11 MAC Protocol. All the WLAN users and the AP contend for channel access to transmit their packets. When a user or the AP has a packet to transmit, it sets its backoff timer by uniformly sampling a value from an initial contention window of size $\mathrm{CW}_{\mathrm{min}}$. It decrements the timer successively in all the slots in which it senses the channel idle. If it senses the channel busy, it freezes its backoff timer. If it again senses the channel idle for a DCF inter-frame space (DIFS) duration T_{DIFS} sec, it resumes decrementing the timer. Once the timer becomes zero, the user transmits to the AP a packet of duration $T_{\mathrm{w,p}}^{\mathrm{UL}} = \frac{L_{\mathrm{UL}}}{R_{\mathrm{w}}}$ sec, where R_{w} is the data rate of the WLAN in bits/sec.

If two or more nodes (users or the AP) start transmission in the same slot, a collision occurs. The channel remains busy for $T_{\mathrm{coll}} = T_{\mathrm{DIFS}} + T_{\mathrm{w,p}}^{\mathrm{UL}}$ sec. All the colliding nodes double their contention windows, subject to a maximum size of $\mathrm{CW}_{\max}$. These nodes then sample new backoff timers from their expanded contention windows and retransmit their packets when the respective timers expire.

After a user successfully sends its packet to the AP, it becomes *dormant*. The AP forwards the packet to the WLAN MEC server for processing. The server takes $T_{\mathrm{w,MEC}} = \frac{L_{\mathrm{UL}} d}{f_{\mathrm{w,MEC}}}$ sec to process the task, where $f_{\mathrm{w,MEC}}$ is its CPU frequency. The server sends packaged computation results to the AP where they are queued. Each time the AP wins channel access contention, it transmits a queued packet to the concerned user. The downlink packet transmission time equals $T_{\mathrm{w,p}}^{\mathrm{DL}} = \frac{L_{\mathrm{DL}}}{R_{\mathrm{w}}}$ sec. Once a user receives the computation result for a task, it comes out of the dormant state and takes up a new task.

2.3 Offloading to Cellular Network

The cellular network employs M orthogonal channels to serve its users, the number depending on the cellular bandwidth. When a user chooses to offload a task to the cellular network, it requests an orthogonal channel. The user is admitted to the network unless all the M channels are occupied. The user incurs an initial access delay of T_{access}. The delay models the time required by the user to request resources and for the BS to grant or refuse them. Once the channel has been allotted, the user takes a time $T_{\mathrm{c}}^{\mathrm{UL}} = \frac{L_{\mathrm{UL}}}{R_{\mathrm{c}}}$ sec to transmit its packet to the BS, where R_{c} is the data rate of the cellular network. The BS forwards the packet to its MEC server for processing.

The server incurs a computation time $T_{\mathrm{c,MEC}} = \frac{L_{\mathrm{UL}} d}{f_{\mathrm{c,MEC}}}$ sec, where $f_{\mathrm{c,MEC}}$ is its CPU frequency. Then it packetizes the result and sends it to the BS. We assume a frequency division duplex system where the uplink and the downlink channels are allotted in pairs to an admitted user. The BS sends the packet to the user on the corresponding downlink channel. This downlink transmission takes $T_{\mathrm{c}}^{\mathrm{DL}} = \frac{L_{\mathrm{DL}}}{R_{\mathrm{c}}}$ sec. The user computes the task locally in case the BS rejects it.

The delay incurred in transporting packets from the AP or the BS to their MEC servers and back is negligible compared to the contention delays and the transmission times [1].

2.4 Probabilistic Offloading Policy

Let a user offload each task to the WLAN with a probability p_{w}, to the cellular network with a probability p_{c}, or locally compute it with probability $1 - p_{\mathrm{c}} - p_{\mathrm{w}}$. This probabilistic offloading policy tractably captures the trade-offs between choosing the RATs because values of p_c and p_w affect the dynamics of the number of users that contend in the WLAN or access the cellular network and, hence, the average task computation delay (latency).

We analyze the conditional average task computation delay of a user given that it chooses a specific RAT as a function of p_{w} and p_{c}. Let $\overline{T}_{\mathrm{c}}$ and $\overline{T}_{\mathrm{w}}$ denote the average delays in the cellular network and the WLAN, respectively.

3 MEC LATENCY ANALYSIS

An exact analysis of the MEC system is intractable even for small values of N and M. We use decoupling approximations and Markov renewal process theory to develop a novel, simplified and accurate analysis [10]. We briefly outline our approach before elaborating it in the following subsections. Let q_{c}, q_{w}, and q_{loc} be the fraction of time a user spends in the cellular network, WLAN, and local network, respectively, having chosen to process tasks in different networks with probabilities p_{w}, p_{c}, and $1 - p_{\mathrm{c}} - p_{\mathrm{w}}$. Furthermore, let $q_{\mathrm{loc}}^{\mathrm{rej}}$ be the fraction of time a user spends in the local network after being rejected by the cellular network. This includes the initial access delay T_{access}. We derive expressions for the conditional average task computation delays $\overline{T}_{\mathrm{w}}$ and $\overline{T}_{\mathrm{c}}$ in terms of q_{c}, q_{w}, q_{loc}, and $q_{\mathrm{loc}}^{\mathrm{rej}}$. We also develop four fixed-point equations in q_{c}, q_{w}, q_{loc} and $q_{\mathrm{loc}}^{\mathrm{rej}}$ to obtain these. Then, the average latency of a user is

$$p_{\mathrm{c}} \overline{T}_{\mathrm{c}} + p_{\mathrm{w}} \overline{T}_{\mathrm{w}} + (1 - p_{\mathrm{c}} - p_{\mathrm{w}}) T_{\mathrm{loc}}. \tag{1}$$

3.1 WLAN Latency Analysis

We analyze the average latency of task execution in the steady-state regime in a WLAN consisting of n users. Due to the probabilistic offloading policy, the number of WLAN users is a random process. We subsequently compute the average latency with respect to the steady state distribution of this process.

For a given task offloaded to the WLAN, let $U(n)$, $W(n)$ and $S(n)$ be the times taken by the user to transmit the packet to the AP, for the downlink packet to reach the head-of-line position of the AP queue since arriving from the MEC server, and by the AP to transmit it to the user, respectively. Then, the total WLAN delay $T_{\mathrm{w}}(n)$ is

$$T_{\mathrm{w}}(n) = U(n) + W(n) + S(n). \tag{2}$$

We consider the following two extreme regimes and use the insights from these to address the general case in Section 3.1.3.

3.1.1 *Low WLAN Loading Regime Where $W(n) \approx T_{w,MEC}$.*

In this regime, the AP and the users face little contention for channel access. Moreover, once a packetized result arrives at the AP, it sees with high probability an empty AP queue. Consequently, the contention delays in the uplink and the downlink are negligible, and the MEC processing time dominates the overall delay. Hence, $T_{\mathrm{w}}(n) \approx U(n) + T_{\mathrm{w,MEC}} + S(n)$.

Since the contention delays are negligible, the contention windows of the AP and the users remain $\mathrm{CW}_{\min}$. Therefore, the average uplink delay is $\mathbf{E}[U(n)] = (\mathrm{CW}_{\min}/2) + T_{\mathrm{w,p}}^{\mathrm{UL}}$, and the average downlink delay is $\mathbf{E}[S(n)] = (\mathrm{CW}_{\min}/2) + T_{\mathrm{w,p}}^{\mathrm{DL}}$. Hence,

$$\mathbf{E}[T_{\mathrm{w}}(n)] = T_{\mathrm{w,p}}^{\mathrm{UL}} + T_{\mathrm{w,MEC}} + T_{\mathrm{w,p}}^{\mathrm{DL}} + \mathrm{CW}_{\min}. \tag{3}$$

3.1.2 *High WLAN Loading Regime Where $W(n) \gg T_{w,MEC}$.*

In this regime, the waiting time in the AP queue dominates the overall delay and the MEC processing time is relatively small. Hence, $T_{\mathrm{w}}(n) \approx U(n) + W(n) + S(n)$. Let $\Theta_{\mathrm{AP}}(n)$ be the AP's throughput in a WLAN with n users. All the users are statistically identical. For each packet offloaded by a user to the AP, the AP transmits a downlink packet to the user. Hence, the throughput of a user in the WLAN is $\Theta_{\mathrm{AP}}(n)/n$ packets/slot. Therefore, the average latency of a user is

$$\mathbf{E}[T_{\mathrm{w}}(n)] = \frac{n}{\Theta_{\mathrm{AP}}(n)}. \tag{4}$$

Expression for $\Theta_{AP}(n)$. At any instant, the n WLAN users fall into two categories:

(1) *Contending Users*: These users are contending for channel access to send their packets to the AP for processing.
(2) *Dormant Users*: These users are waiting to receive their processed task results back from the AP. Therefore, they are not contending for the channel.

The dynamics of the contending and dormant users can be modeled as a Markov renewal process [5, Chap. 5.4]. Let $T_0 = 0$, and T_i, for $i \geq 1$, be the instant when the i^{th} transmission, be it uplink or downlink, ends. Let G_i be the number of users contending at time T_i. Then, $(T_1, G_1), (T_2, G_2), \dots$ form a Markov renewal process.

Figure 2 shows the transition diagram of the Markov chain G_i for $i \geq 0$. Its state space is $\{0, 1, \dots, n\}$. For $n_{\text{cont}} \in \{0, 1, \dots, n-1\}$, a transition from state n_{cont} to state $n_{\text{cont}} + 1$ occurs when the AP wins contention, transmits a packet, and the corresponding dormant user stars contending again. This event occurs with probability $1/(n_{\text{cont}} + 1)$. Similarly, a transition from state n_{cont} to state $n_{\text{cont}} - 1$ occurs when any one of the contending users wins contention, which happens with probability $n_{\text{cont}}/(n_{\text{cont}} + 1)$. Note that when $n_{\text{cont}} = n$, the AP does not participate in contention since all the users are yet to transmit their packets to the AP. Hence, from state 0 a transition happens to state 1 with probability 1. The stationary distribution of the Markov chain, $\pi_{\text{w}}(.)$, can be shown to be

$$\pi_{\text{w}}(n_{\text{cont}}) = \begin{cases} \left(\sum_{n_{\text{cont}}=0}^{n-1} \frac{n_{\text{cont}}+1}{n_{\text{cont}}!} + \frac{1}{(n-1)!}\right)^{-1}, \text{if } n_{\text{cont}} = 0, \\ \frac{(n_{\text{cont}}+1)}{n_{\text{cont}}!}\pi_{\text{w}}(0), \text{ if } 1 \leq n_{\text{cont}} \leq n-1, \\ \frac{1}{(n_{\text{cont}}-1)!}\pi_{\text{w}}(0), \text{ if } n_{\text{cont}} = n. \end{cases} \tag{5}$$

We employ the following *decoupling approximation* originally proposed in [10]. Consider a WLAN with a saturated AP and n_{cont} saturated users, which always have packets to transmit and, thus, always contend. In the steady state, each user and the AP start the transmission of a packet in each backoff slot at a rate $\beta_{n_{\text{cont}}+1}$ attempts/slot, which is referred to as the attempt rate. *We assume that the backoff process of a given node is independent of the aggregate attempt process of the other* n_{cont} *nodes.* Hence, from the point of view of the given node, the number of attempts by the other nodes in successive slots are independent and identically distributed (i.i.d.) Binomial random variables with parameters n_{cont} and $\beta_{n_{\text{cont}}+1}$. The attempt rate $\beta_{n_{\text{cont}}+1}$ is obtained via a saturated network analysis [10].

Note that $\beta_{n_{\text{cont}}+1}$ is the steady state attempt rate in a WLAN with n_{cont} contending users (and the AP), whereas the number of contending users in our WLAN model constitutes a Markov chain. *As in [11], we also assume that whenever there are* n_{cont} *contending users, each of these users and the AP attempt in a backoff slot with probability* $\beta_{n_{cont}+1}$.

Using the above approximations we derive the following expression for $\Theta_{\text{AP}}(n)$.

Lemma 1. *For the WLAN with n users, the AP throughput is given by*

$$\Theta_{AP}(n) = \frac{\sum_{n_{cont}=0}^{n-1} \pi_{\text{w}}(n_{cont}) \frac{1}{n_{cont}+1}}{\sum_{n_{cont}=0}^{n} \pi_{\text{w}}(n_{cont}) \mathbf{E}\left[X(n_{cont})\right]}, \tag{6}$$

where $\mathbf{E}\left[X(n_{cont})\right]$ *is the average duration of the renewal cycle given that* n_{cont} *users are contending for channel access. For* $0 \leq n_{cont} \leq n-1$,

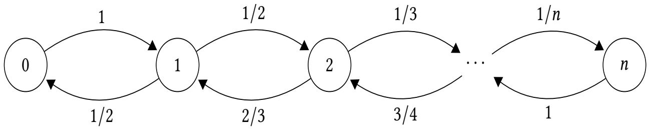


Figure 2: Markov chain for the number of contending users n_{cont} in the WLAN with n users.

$$\mathbf{E}\left[X(n_{cont})\right] = \frac{\begin{bmatrix} (1-\beta_{n_{cont}+1})^{n_{cont}+1} + \beta_{n_{cont}+1}(1-\beta_{n_{cont}+1})^{n_{cont}} T_{w,p}^{DL} \\ + P_{coll}^{(n_{cont}+1)} T_{coll} + n_{cont}\beta_{n_{cont}+1}(1-\beta_{n_{cont}+1})^{n_{cont}} T_{w,p}^{UL} \end{bmatrix}}{(n_{cont}+1)\beta_{n_{cont}+1}(1-\beta_{n_{cont}+1})^{n_{cont}}}, \tag{7}$$

where

$$P_{coll}^{(n_{cont}+1)} = 1 - (1-\beta_{n_{cont}+1})^{n_{cont}+1} - (n_{cont}+1)\beta_{n_{cont}+1}(1-\beta_{n_{cont}+1})^{n_{cont}}. \tag{8}$$

For $n_{cont} = n$,

$$\mathbf{E}\left[X(n)\right] = \frac{\begin{bmatrix} (1-\beta_n)^n + P_{coll}^{(n)} T_{coll} \\ + (n-1)\beta_n(1-\beta_n)^{n-1} T_{w,p}^{UL} \end{bmatrix}}{n\beta_n(1-\beta_n)^{n-1}}. \tag{9}$$

Proof. See Appendix A. □

3.1.3 General Case. The transmission of packets in the AP queue to the users and the processing of other tasks at the MEC server happen in parallel. The overall delay for these parallel processes is predominantly determined by the process with the greater delay. This motivates the following heuristic for the WLAN delay for arbitrary loading; it combines the results for the low and high loading regimes:

$$\mathbf{E}\left[T_{\text{w}}(n)\right] \approx \max\left\{\frac{n}{\Theta_{\text{AP}}(n)}, T_{\text{w,p}}^{\text{UL}} + T_{\text{w,MEC}} + T_{\text{w,p}}^{\text{DL}} + \text{CW}_{\text{min}}\right\}. \tag{10}$$

We assess the accuracy of the heuristic in Section 5.1.

We finally make another decoupling assumption that when a user offloads a task to the WLAN, the probability that the WLAN has $0 \leq n \leq N-1$ users is $\binom{N-1}{n} q_{\text{w}}^n (1-q_{\text{w}})^{N-1-n}$. Hence, averaging (10) over n, the average task computation delay in the WLAN becomes

$$\overline{T}_{\text{w}} = \sum_{n=0}^{N-1} \binom{N-1}{n} q_{\text{w}}^n (1-q_{\text{w}})^{N-1-n} \mathbf{E}\left[T_{\text{w}}(n+1)\right]. \tag{11}$$

3.2 Cellular Network Latency Analysis

Recall that a given user's request to offload a task to the cellular network is accepted if and only if one of the M channels is free. When it is accepted, the total time T_{cell} spent by the user in the cellular network is given by

$$T_{\text{cell}} = T_{\text{access}} + T_{\text{c}}^{\text{UL}} + T_{\text{c,MEC}} + T_{\text{c}}^{\text{DL}}. \tag{12}$$

Let η_{full} be the probability that the user finds all the M channels occupied. We now obtain the average latency seen by the user for the tasks that it chooses to offload to the cellular network in terms of η_{full}. *We assume that having offloaded a task to the cellular network, if*

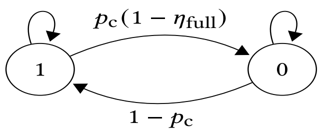


Figure 3: Markov chain for a user's task offloading indicator. The probabilities of self-loops are not shown to avoid clutter.

the user chooses to offload its next task also to the cellular network, it is admitted with probability 1. The rationale behind this assumption is that there will be one or more free channels on completion of a task, and the tagged user is unlikely to find as many contending users.

We define Y_i, for $i \geq 1$, to be the indicators of the tagged user's tasks being processed in the cellular network. More precisely, $Y_i = 1$ if the tagged user's i^{th} task is processed in the cellular network and $Y_i = 0$ otherwise. Then Y_i, for $i \geq 1$, is a Markov chain with a state transition diagram as shown in Figure 3. A transition from state 0 to 1 happens if the user chooses to offload the next task to the cellular network and is admitted. A transition from state 1 to 0 happens if the user does not choose the cellular network for the next task.

Lemma 2. *The average latency for the tasks that a user chooses to offload to the cellular network is given by*

$$\overline{T}_c = \frac{f_{cell}}{p_c} T_{cell} + \left(1 - \frac{f_{cell}}{p_c}\right)(T_{access} + T_{loc}). \tag{13}$$

Here, f_{cell} is the fraction of tasks a user offloads to the cellular network, and is given by

$$f_{cell} = \frac{p_c(1-\eta_{full})}{1 - p_c\eta_{full}}. \tag{14}$$

Proof. See Appendix B. □

Expression for η_{full}. We assume that the probability of two or more users completing their task computations in a slot is negligible because the slot duration is much smaller than the latencies of the WLAN or the cellular or local networks. Suppose there are k users in the cellular network. Let η_k^{exit} be the probability that exactly one user out of these k users exits the cellular network in a given slot. Assuming independence across the cellular users and across slots, we have $\eta_k^{\text{exit}} = \frac{k(1-p_c)}{T_{\text{cell}}}\left(1 - \frac{1-p_c}{T_{\text{cell}}}\right)^{k-1}$. Independence across slots strictly holds only if the sojourn times in the cellular network are *geometric* but is a reasonable assumption if sojourn times are much larger than the slot duration. Let η_k^{enter} be the probability of one WLAN or local network user choosing to offload a new task to the cellular network in a given slot.

Lemma 3. *The probability η_k^{enter} is given by*

$$\eta_k^{enter} = \sum_{n_w=0}^{N-k} \binom{N-k}{n_w} \frac{q_w^{n_w}\left(q_{loc} + q_{loc}^{rej}\right)^{n_l}}{(1-q_c)^{N-k}} \zeta(n_w, n_l), \tag{15}$$

where $n_l = N - k - n_w$ and $\zeta(n_w, n_l)$ is the probability that one user from the other networks chooses to offload a task to the cellular network in a slot given than the WLAN has n_w users and the local network has n_l users. Furthermore,

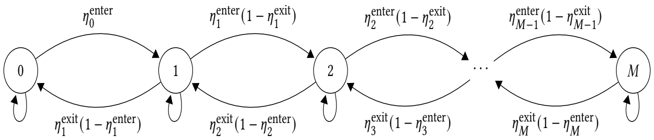

η_0^{enter}
$\eta_1^{\text{enter}}(1-\eta_1^{\text{exit}})$
$\eta_2^{\text{enter}}(1-\eta_2^{\text{exit}})$
$\eta_{M-1}^{\text{enter}}(1-\eta_{M-1}^{\text{exit}})$
0
1
2
$\cdots$
M
$\eta_1^{\text{exit}}(1-\eta_1^{\text{enter}})$
$\eta_2^{\text{exit}}(1-\eta_2^{\text{enter}})$
$\eta_3^{\text{exit}}(1-\eta_3^{\text{enter}})$
$\eta_M^{\text{exit}}(1-\eta_M^{\text{enter}})$

Figure 4: Markov chain for the number of users in the cellular network. The probabilities of the self-loops are not shown to avoid clutter.

$$\zeta(n_w, n_l) = \left(1 - \frac{p_c}{\mathbf{E}\left[T_w(n_w)\right]}\right)^{n_w-1}\left(1 - \frac{p_c}{T_{loc}}\right)^{n_l-1} \times \left[\frac{n_w p_c}{\mathbf{E}\left[T_w(n_w)\right]}\left(1 - \frac{p_c}{T_{loc}}\right) + \frac{n_l p_c}{T_{loc}}\left(1 - \frac{p_c}{\mathbf{E}\left[T_w(n_w)\right]}\right)\right], \tag{16}$$

where $\mathbf{E}\left[T_w(n_w)\right]$ *is given by* (10).

Proof. See Appendix C. □

The number of users in the cellular network in successive slots is an irreducible Markov chain for $p_c < 1$.[1] Let P_c be its transition probability matrix. A transition from state k to $k+1$ occurs when one user from the WLAN or the local network chooses to offload its next task to the cellular network and no cellular user exits in the same slot. A transition from state k to $k-1$ occurs when one cellular user exits and no user from the WLAN or the local network chooses to offload a task to the cellular network in the same slot. Thus,

$$\mathrm{P}_c(k, k+1) = \eta_k^{\text{enter}}\left(1 - \eta_k^{\text{exit}}\right), \text{ for } 0 \leq k \leq M-1, \tag{17}$$

$$\mathrm{P}_c(k, k-1) = \eta_k^{\text{exit}}\left(1 - \eta_k^{\text{enter}}\right), \text{ for } 1 \leq k \leq M. \tag{18}$$

Figure 4 illustrates this Markov chain. Let π_c be its stationary distribution. We can show that

$$\pi_c(M) = \frac{\prod_{h=0}^{M-1}\eta_h^{\text{enter}}\left(1-\eta_h^{\text{exit}}\right)}{\prod_{h=1}^{M}\eta_h^{\text{exit}}\left(1-\eta_h^{\text{enter}}\right)}\left(1 + \sum_{k=1}^{M}\frac{\prod_{h=0}^{k-1}\eta_h^{\text{enter}}\left(1-\eta_h^{\text{exit}}\right)}{\prod_{h=1}^{k}\eta_h^{\text{exit}}\left(1-\eta_h^{\text{enter}}\right)}\right)^{-1}.$$

Furthermore,

$$\eta_{\text{full}} = \frac{\pi_c(M)\eta_M^{\text{enter}}\left(1-\eta_M^{\text{exit}}\right)}{\sum_{k=0}^{M}\pi_c(k)\eta_k^{\text{enter}}}. \tag{19}$$

3.3 Fixed-Point Equations

The probability of finding a user in a network in an arbitrary slot is equal to the fraction of time spent by a user in that network. For instance, the average time per task a user spends in the WLAN is $p_w\overline{T}_w$. Thus, the total time spent by a user per task is $p_w\overline{T}_w + p_c\overline{T}_c + (1 - p_w - p_w)T_{\text{loc}}$. Hence,

$$q_w = \frac{p_w\overline{T}_w}{p_w\overline{T}_w + p_c\overline{T}_c + (1-p_c-p_w)T_{\text{loc}}}. \tag{20}$$

Similarly, we can show that

$$q_c = \frac{f_{\text{cell}}T_{\text{cell}}}{p_w\overline{T}_w + p_c\overline{T}_c + (1-p_c-p_w)T_{\text{loc}}}, \tag{21}$$

[1] Note that the extreme case of $p_c = 1$ is not of interest. If $p_c = 1$, once a user enters the cellular network, it never exits.

$$q_{\text{loc}}^{\text{rej}} = \frac{(p_{\text{c}} - f_{\text{cell}})(T_{\text{access}} + T_{\text{loc}})}{p_{\text{w}}\overline{T}_{\text{w}} + p_{\text{c}}\overline{T}_{\text{c}} + (1 - p_{\text{c}} - p_{\text{w}})T_{\text{loc}}}, \tag{22}$$

$$q_{\text{loc}} = \frac{(1 - p_{\text{w}} - p_{\text{c}})T_{\text{loc}}}{p_{\text{w}}\overline{T}_{\text{w}} + p_{\text{c}}\overline{T}_{\text{c}} + (1 - p_{\text{c}} - p_{\text{w}})T_{\text{loc}}}. \tag{23}$$

Equations (20), (21), (22), and (23) constitute four fixed-point equations in four variables. We solve these numerically to determine q_{w}, q_{c}, q_{loc}, and $q_{\text{loc}}^{\text{rej}}$. Substituting these solutions in (11) and (13) yields the average latency as a function of p_{c} and p_{w}. We then numerically find the optimal p_{c} and p_{w} and the optimal delay.

4 UCB-BASED OFFLOADING POLICY

The UCB algorithm is used to choose between different actions in the face of uncertainty in the multi-armed bandit (MAB) problem framework [2]. It has been used extensively to tackle the exploration-exploitation trade-off in a multi-RAT system [16]. We propose a distributed UCB-based policy for RAT selection, treating negative latencies of various networks as their rewards. The users do not need to know the system parameters to execute the policy.

Let us consider a tagged user. Let $a(k) \in \mathrm{I} \triangleq \{1, 2, 3\}$ be the network it uses for its k^{th} task, where 1, 2 and 3 represent the local network, WLAN and cellular network, respectively. Let L_k be the latency incurred by its k^{th} task. The user chooses the RATs to minimize the average latency across the tasks. Its offloading decision for a task depends on the delays incurred by its previous tasks. Let $c_i(k)$ be the number of tasks it has offloaded to network i up to the k^{th} task and $\overline{T}_i(k)$ be the average delay incurred by these $c_i(k)$ tasks.

The UCB-based offloading policy, inspired by [2], is shown in Algorithm 1. The policy is initialized by offloading the first three tasks to the local network, to the WLAN and to the cellular network, respectively, and updating the average delays $\overline{T}_i(3)$ for $i \in \mathrm{I}$. For $k \geq 4$, the k^{th} task is offloaded to the network with the highest $r_i(k)$. The exploration factor η controls how often a user explores networks other than the one it estimates to be the optimal.

Algorithm 1 UCB-Based Offloading Policy

```
for $k = 1 : 3$ do //initialization
    Choose $a(k) = k$
    Observe $L_k$
    Update $c_k(3) = 1,\ \overline{T}_k(3) = L_k$
end for
for $k \geq 4$ do
    Set $r_i(k) = -\overline{T}_i(k-1) + \eta\sqrt{\frac{\log(k-1)}{c_i(k-1)}}, \forall\ i \in \mathrm{I}$
    Choose $a(k) = \mathrm{argmax}_{i \in \mathrm{I}} \{r_i(k)\}$
    Observe $L_k$
    Update $c_i(k) = c_i(k-1) + \mathbb{1}(i = a(k))$,
        $\overline{T}_i(k) = \overline{T}_i(k-1) + \mathbb{1}(i = a(k))\frac{(L_k - \overline{T}_i(k-1))}{c_i(k)}$,
        $\forall\ i \in \mathrm{I}$
end for
```

5 NUMERICAL RESULTS

We now present simulation results to assess the accuracy of the analysis, and to evaluate the performance of the probabilistic and UCB-based offloading policies. The simulation parameters are listed in Table 1. The parameter values for L_{UL}, L_{DL}, and d correspond to an augmented reality application [3].

Table 1: Simulation parameters

Parameter	Value	Parameter	Value
δ	9 μsec	$f_{\text{c,MEC}}$	10 GHz
T_{DIFS}	34 μsec	T_{SIFS}	16 μsec
R_{w}	100 Mbps	R_{c}	10 Mbps
CW_{max}	1024	CW_{min}	16
T_{access}	4 msec	d	140 cycles/bit
L_{UL}	64 Kb	L_{DL}	12 Kb
f_{u}	2.2 GHz	$f_{\text{w,MEC}}$	10 GHz

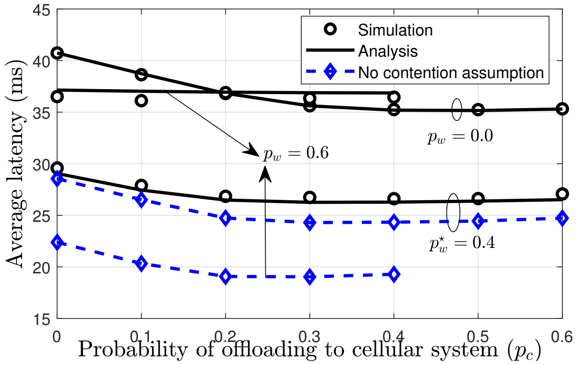


Figure 5: Average latency as a function of p_{c} for different values of p_{w} ($N = 50$ and $M = 10$).

5.1 Probabilistic Offloading Policy

Fig. 5 plots the average latency as a function of p_{c} for different p_{w}. Note that given p_{w}, p_{c} can vary from 0 to $1 - p_{\text{w}}$. For a given p_{w}, as p_{c} increases, the average latency first decreases and then gradually increases. This is because more users enter the cellular network until all the M channels are occupied. Beyond this, the BS rejects the users that attempt to enter the cellular network, which increases the average latency. For a given p_{c}, as p_{w} increases, the average latency decreases up to $p_{\text{w}} = 0.4$ and then increases. This is because the WLAN with its high data rate has a lower average latency than the cellular or local network when few users contend in it. However, the MAC contention delays increase as more users enter the WLAN. The analysis and simulation results match. We also show the results for the model where the uplink and downlink contention delays are ignored in the WLAN, as done in [9]. Neglecting the contention delays leads to an underestimation of the average latency, especially when p_w is large.

Fig. 6 plots the optimal delay as a function of the number of users for different values of M. For every value of N and M, the optimal delay that minimizes the average latency, which is given in (1), is determined numerically. For any M, the delay increases as N increases. For $N > M$, the delay decreases as M increases because more users can be served by the cellular network. For $N \leq M$, the delay is insensitive to M because most users choose to offload their tasks to the lightly-loaded WLAN due to its lower contention delays.

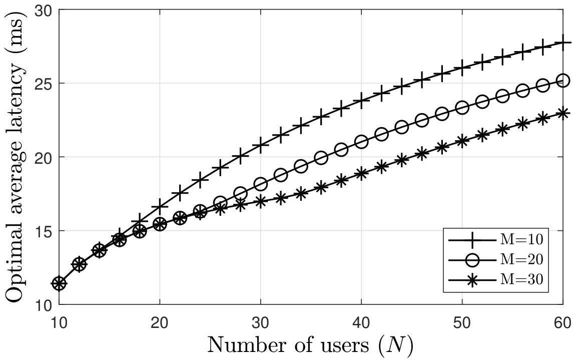
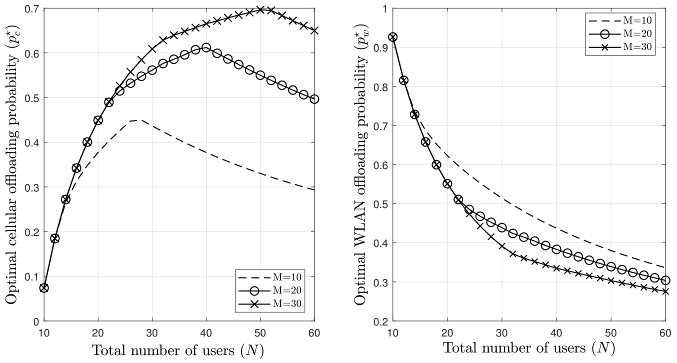


Figure 6: Optimal average latency as a function of the number of users for different values of M.



(a) Cellular offloading probabilities ($p_c^\star$). **(b) WLAN offloading probabilities ($p_w^\star$).**

Figure 7: Optimal offloading probabilities as a function of number of users for different number of cellular channels.

Fig. 7a plots the optimal cellular offloading probabilities $p_c^\star$ as a function of N for different values of M. Fig. 7b plots the corresponding optimal WLAN offloading probabilities $p_w^\star$. For given cellular network capacity M, $p_c^\star$ increases with N until the average number of users in the cellular network exceeds M. Thereafter, it decreases with N since rejections from the cellular network entail higher average delays. The trends are different for $p_w^\star$. For any M, $p_w^\star$ decreases monotonically as N increases because the contention delay increases. We see that $p_c^\star$ is more sensitive to changes in M than $p_w^\star$, especially for $N > M$. This is because the users that would have locally computed their tasks enter the cellular network as M increases.

5.2 UCB-Based Offloading Policy

Fig. 8 compares the optimal average delays of the UCB-based and probabilistic offloading policies for different values of the number of users N. For each N, the optimal average delay of the UCB-based policy is determined by sweeping the parameter η over the range $[0, 100]$ and then using the value that provides the lowest delay. As expected, the optimal delay of both policies increases as N increases. The UCB-based policy has a similar, albeit marginally lower, delay compared to the probabilistic policy. In classical MAB problems with a reward structure as described in Section 4, the UCB algorithm would give inferior rewards. However, in the RAT selection problem, the UCB-based policy adapts based on the previous task delays and exploits the correlation of rewards across the networks and the users. Unlike the static probabilistic offloading policy, it opportunistically offloads tasks to the networks, yielding a superior performance.

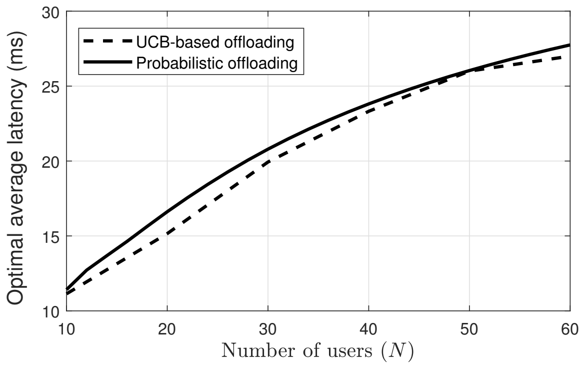


Figure 8: Comparison of the optimal average latency of the probabilistic and UCB-based offloading policies for different numbers of users.

6 CONCLUSIONS

We studied an MEC network in which the users offloaded their tasks to a WLAN or a cellular network, or could compute them locally. We developed a novel analytical framework to characterize the average latency. The policy brought out the interplay between the offloading probabilities and loading of the networks. The analysis accounted for the coupled uplink and downlink transmissions by the users and the AP, respectively, variations in the number of users served by each RAT, and the non-negligible AP queuing delays that arise due to contention. We saw that ignoring WLAN MAC contention could lead to underestimation of the average latency. We also proposed a UCB-based offloading policy. The average latencies of the two policies were similar though the UCB-based policy marginally outperformed.

ACKNOWLEDGMENTS

Chandramani Singh acknowledges the support from Aircel IISc Centre of Excellence in Telecommunications Project 39010C and the Centre for Networked Intelligence (a Cisco CSR Initiative) at IISc.

REFERENCES

[1] Jaewon Ahn, Joohyung Lee, Dusit Niyato, and Hong-Shik Park. 2020. Novel QoS-Guaranteed Orchestration Scheme for Energy-Efficient Mobile Augmented Reality Applications in Multi-Access Edge Computing. *IEEE Trans. Veh. Technol.* 69, 11 (Nov. 2020), 13631–13645.
[2] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning* 47, 2 (2002), 235–256.
[3] Xing Chen and Guizhong Liu. 2021. Energy-Efficient Task Offloading and Resource Allocation via Deep Reinforcement Learning for Augmented Reality in Mobile Edge Networks. *IEEE Internet Things J.* 8, 13 (Jul. 2021), 10843–10856.
[4] Boutheina Dab, Nadjib Aitsaadi, and Rami Langar. 2019. Joint Optimization of Offloading and Resource Allocation Scheme for Mobile Edge Computing. In *Proc. WCNC*. IEEE, Marrakesh, Morocco, 1–7.
[5] R.G. Gallager. 2013. *Stochastic Processes: Theory for Applications* (1 ed.). Cambridge Univ. Press, Cambridge.
[6] Fengxian Guo, F. Richard Yu, Heli Zhang, Hong Ji, Victor C. M. Leung, and Xi Li. 2020. An Adaptive Wireless Virtual Reality Framework in Future Wireless

Networks: A Distributed Learning Approach. *IEEE Trans. Veh. Technol.* 69, 8 (Aug. 2020), 8514–8528.

[7] Mohammad Asif Habibi, Meysam Nasimi, Bin Han, and Hans D. Schotten. 2019. A Comprehensive Survey of RAN Architectures Toward 5G Mobile Communication System. *IEEE Access* 7 (2019), 70371–70421.

[8] Di Han, Wei Chen, and Yuguang Fang. 2020. Joint Channel and Queue Aware Scheduling for Latency Sensitive Mobile Edge Computing With Power Constraints. *IEEE Trans. Wireless Commun.* 19, 6 (Jun. 2020), 3938–3951.

[9] Tai Manh Ho and Kim-Khoa Nguyen. 2022. Joint Server Selection, Cooperative Offloading and Handover in Multi-Access Edge Computing Wireless Network: A Deep Reinforcement Learning Approach. *IEEE Trans. Mob. Comput.* 21, 7 (Jul. 2022), 2421–2435.

[10] Anurag Kumar, Eitan Altman, Daniele Miorandi, and Munish Goyal. 2007. New Insights From a Fixed-Point Analysis of Single Cell IEEE 802.11 WLANs. *IEEE/ACM Trans. Netw.* 15, 3 (Jun. 2007), 588–601.

[11] G. Kuriakose, S. Harsha, A. Kumar, and V. Sharma. 2009. Analytical Models for Capacity Estimation of IEEE 802.11 WLANs using DCF for Internet Applications. *Wireless Netw.* 15, 2 (Feb. 2009), 259–277.

[12] Jeongho Kwak, Yeongjin Kim, Joohyun Lee, and Song Chong. 2015. DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems. *IEEE J. Sel. Areas Commun.* 33, 12 (Dec. 2015), 2510–2523.

[13] Arghyadip Roy, Prasanna Chaporkar, and Abhay Karandikar. 2018. Optimal Radio Access Technology Selection Algorithm for LTE-WiFi Network. *IEEE Trans. Veh. Technol.* 67, 7 (Jul. 2018), 6446–6460.

[14] Feng Wang, Jie Xu, Xin Wang, and Shuguang Cui. 2018. Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems. *IEEE Trans. Wireless Commun.* 17, 3 (Mar. 2018), 1784–1797.

[15] Fei Wang and Xi Zhang. 2018. Dynamic Interface-Selection and Resource Allocation over Heterogeneous Mobile Edge-Computing Wireless Networks with Energy Harvesting. In *Proc. INFOCOM*. IEEE, Honolulu, HI, USA, 190–195.

[16] Bochun Wu, Tianyi Chen, and Xin Wang. 2020. An MAB Approach for MEC-centric Task-offloading Control in Multi-RAT HetNets. In *Proc. Int. Conf. Commun. (ICC)*. IEEE, Dublin, Ireland, 1–6.

[17] Huaming Wu and Katinka Wolter. 2018. Stochastic Analysis of Delayed Mobile Offloading in Heterogeneous Networks. *IEEE Trans. Mob. Comput.* 17, 2 (Feb. 2018), 461–474.

[18] X. Yang, X. Yu, H. Huang, and H. Zhu. 2019. Energy Efficiency Based Joint Computation Offloading and Resource Allocation in Multi-Access MEC Systems. *IEEE Access* 7 (Aug. 2019), 117054–117062.

A PROOF OF LEMMA 1

In a renewal cycle of the Markov chain, let $R(n_{\text{cont}})$ denote the reward when n_{cont} users contend for channel access along with the AP. Using the Markov renewal reward theorem [5, Chap. 5.4], the AP throughput $\Theta_{\text{AP}}(n)$ is given by

$$\Theta_{\text{AP}}(n) = \frac{\mathbf{E}_{\mathbf{n}_{\text{cont}}}\left[\mathbf{R}(\mathbf{n}_{\text{cont}})\right]}{\mathbf{E}_{\mathbf{n}_{\text{cont}}}\left[\mathbf{E}\left[\mathbf{X}(\mathbf{n}_{\text{cont}})\right]\right]}. \tag{24}$$

a) Evaluating $\mathbf{E}_{\mathbf{n}_{cont}}\left[\mathbf{R}(\mathbf{n}_{cont})\right]$*:* The reward is 1 when the AP wins contention, which happens with a probability of $1/(n_{\text{cont}}+1)$, for $0 \le n_{\text{cont}} \le n-1$. It is 0 otherwise. When $n_{\text{cont}} = n$, the AP is not contending and the reward is 0. Hence, the average reward is given by $\sum_{n_{\text{cont}}=0}^{n-1} \pi_{\text{w}}(n_{\text{cont}})/(n_{\text{cont}}+1)$.

b) Evaluating $\mathbf{E}_{\mathbf{n}_{cont}}\left[\mathbf{E}\left[\mathbf{X}(\mathbf{n}_{cont})\right]\right]$*:* A renewal cycle ends with the completion of an uplink or downlink transmission. For $0 \le n_{\text{cont}} \le n-1$, the following four mutually exclusive events dictate the length of the renewal cycle: (i) Channel is idle and remains so for 1 slot. The probability of this event occurring is $(1-\beta_{n_{\text{cont}}+1})^{n_{\text{cont}}+1}$. The renewal cycle then continues. (ii) The channel is busy due to a transmission by the AP. The probability of this event is $\beta_{n_{\text{cont}}+1}(1-\beta_{n_{\text{cont}}+1})^{n_{\text{cont}}}$ and duration is $T_{\text{w,p}}^{\text{DL}}$. (iii) The channel is busy due to a transmission by a user. The probability of this event is $n_{\text{cont}}\beta_{n_{\text{cont}}+1}(1-\beta_{n_{\text{cont}}+1})^{n_{\text{cont}}}$ and duration is $T_{\text{w,p}}^{\text{UL}}$. (iv) The channel is busy due to a collision of duration T_{coll}. The probability of this event is $P_{\text{coll}}^{(n_{\text{cont}}+1)}$. Hence,

$$\begin{aligned}\mathbf{E}\left[X(n_{\text{cont}})\right] =& (1-\beta_{n_{\text{cont}}+1})^{n_{\text{cont}}+1}(1+\mathbf{E}\left[X(n_{\text{cont}})\right]) \\ &+ \beta_{n_{\text{cont}}+1}(1-\beta_{n_{\text{cont}}+1})^{n_{\text{cont}}}T_{\text{w,p}}^{\text{DL}} \\ &+ P_{\text{coll}}^{(n_{\text{cont}}+1)}(T_{\text{coll}}+\mathbf{E}\left[X(n_{\text{cont}})\right]) \\ &+ n_{\text{cont}}\beta_{n_{\text{cont}}+1}(1-\beta_{n_{\text{cont}}+1})^{n_{\text{cont}}}T_{\text{w,p}}^{\text{UL}}.\end{aligned}$$

Rearranging the terms yields (7).

For $n_{\text{cont}} = n$, the renewal cycle duration calculations are similar except that the AP does not contend.

B PROOF OF LEMMA 2

Let ν be the stationary distribution of the Markov chain $Y_1, Y_2, \ldots$. It can be shown that

$$\nu(1) = p_{\text{c}}(1-\eta_{\text{full}})/(1-p_{\text{c}}\eta_{\text{full}}). \tag{25}$$

Since $f_{\text{cell}} = \nu(1)$, (14) follows.

Next, let L_i be the latency incurred by the i^{th} task of the tagged user. Also, define $e(i)$, for $i \ge 1$, as follows: $e(i) = 1$ if the user chooses to offload the i^{th} task to the cellular network and $e(i) = 0$ otherwise. Then,

$$\overline{T}_{\text{c}} = \lim_{K\to\infty} \frac{\mathbf{E}\left[\sum_{\mathbf{i}=1}^{\mathbf{K}} \mathbf{L_i e(i)}\right]}{\mathbf{E}\left[\sum_{\mathbf{i}=1}^{\mathbf{K}} \mathbf{e(i)}\right]}. \tag{26}$$

Applying the Markov renewal reward theorem to both the numerator and denominator of (26) and simplifying, we can show that

$$\begin{aligned}\overline{T}_{\text{c}} =& \nu(1)T_{\text{cell}} \\ &+ \nu(0)\left[(1-\eta_{\text{full}})T_{\text{cell}} + \eta_{\text{full}}(T_{\text{access}}+T_{\text{loc}})\right],\end{aligned} \tag{27}$$

Since $\nu(1)+\nu(0)(1-\eta_{\text{full}}) = f_{\text{cell}}/p_{\text{c}}$ and $\nu(0)\eta_{\text{full}} = 1-(f_{\text{cell}}/p_{\text{c}})$, (13) follows.

C PROOF OF LEMMA 3

Each user that is in the WLAN finishes computing its task in a slot and chooses to offload its next task to the cellular network with probability $\frac{p_{\text{c}}}{\mathbf{E}[T_{\text{w}}(n_{\text{w}})]}$. Similarly, each user that is in the local network finishes computing its task in a slot and chooses to offload its next task to the cellular network with probability $\frac{p_{\text{c}}}{T_{\text{loc}}}$. Hence, given that there are n_{w} users in the WLAN and n_{l} users in the local network, the probability $\zeta(n_{\text{w}}, n_{\text{loc}})$ of exactly one of these users choosing to offload its task to the cellular network is

$$\begin{aligned}&\frac{n_{\text{w}}p_{\text{c}}}{\mathbf{E}\left[T_{\text{w}}(n_{\text{w}})\right]}\left(1-\frac{p_{\text{c}}}{\mathbf{E}\left[T_{\text{w}}(n_{\text{w}})\right]}\right)^{n_{\text{w}}-1}\left(1-\frac{p_{\text{c}}}{T_{\text{loc}}}\right)^{n_{\text{l}}} \\ &\quad + \frac{n_{\text{l}}p_{\text{c}}}{T_{\text{loc}}}\left(1-\frac{p_{\text{c}}}{T_{\text{loc}}}\right)^{n_{\text{l}}-1}\left(1-\frac{p_{\text{c}}}{\mathbf{E}\left[T_{\text{w}}(n_{\text{w}})\right]}\right)^{n_{\text{w}}}.\end{aligned}$$

This establishes (16).

Further, note that there are k users in the cellular network and $N-k$ in other networks. A user that is not in the cellular network is in the WLAN with probability $\frac{q_{\text{w}}}{1-q_{\text{c}}}$ and in the local network with probability $\frac{q_{\text{loc}}+q_{\text{loc}}^{\text{rej}}}{1-q_{\text{c}}}$. Hence, assuming independence across the non-cellular users and across slots, the number of users in the WLAN is a Binomial random variable with parameters $N-k$ and $\frac{q_{\text{w}}}{1-q_{\text{c}}}$. Therefore, unconditioning on the numbers of users in the WLAN and in the local network, we obtain the probability η_k^{enter} of one non-cellular user choosing to offload a new task to the cellular network in a given slot, as given in (15).