Parallel Computing at the Extreme Edge: Spatiotemporal Analysis

Mahmoud Abdelhadi*, Sameh Sorour*, Hesham ElSawy*, Sara A. Elsayed*, and Hossam Hassanein*
*School of Computing, Queen's University, Kingston, Ontario, Canada
{m.abdelhadi, sameh.sorour, hesham.elsawy}@queensu.ca, {selsayed, hossam}@cs.queensu.ca

***Abstract*—Multi-access Edge Computing (MEC) is a revolutionary computing paradigm that facilitates delay-sensitive and/or data-intensive applications associated with the Internet of Things (IoT). Harvesting copious yet underutilized computational resources of the Extreme Edge Devices (EEDs) is foreseen as a promising endeavor. Such EEDs offer a unique opportunity to bring the computing service closer to IoT devices to curtail delay. However, the efficacy of extreme-edge parallel computing paradigm is profoundly impacted by i) wireless device-to-device communication performance, that is required for task offloading; and ii) computing capabilities of the EEDs, that governs the execution time of each task. In this context, we propose a novel spatiotemporal framework that employs stochastic geometry and continuous time Markov chains to jointly analyze the interwoven communication and computation performance of extreme edge computing systems. Based on the incorporated framework, we study the influence of various system parameters on the task response delay. Our findings reveal the existence of an optimal number of EEDs that need to be recruited in order to minimize the task response delay. Moreover, we show that in some cases, our model can outperform the normal MEC offloading systems.**

***Keywords*-Multi-access Edge Computing, Stochastic Geometry, Continuous Time Markov Chain, Computation offloading.**

I. Introduction

With the advent of the Internet of Things (IoT), it is anticipated that 75.44 billion IoT devices will be connected to the Internet by 2025 [1]. In addition, it is expected that the IoT market size will rise up to $15 trillion by the same year [1]. This can trigger a broad spectrum of latency-sensitive IoT applications with strenuous Quality of Service (QoS) requirements [2]. Such requirements cannot be adequately satisfied by cloud computing, due to the distant geographical location of cloud data centers, as well as the huge traffic influx imposed at backhaul links [2]. Multi-access Edge Computing (MEC) has emerged as a propitious computing paradigm that can bring the computing service within close proximity to end devices, thus significantly reducing the delay and successfully satisfying the soaring demands of IoT applications [3].

In MEC, efficient task offloading decisions are pivotal to achieve promising performance gains. Most existing MEC platforms depend on the availability of computationally capable Base Stations (BSs) to perform the offloaded computational tasks [3]. Recently, various research efforts [4]–[6] have explored the potential of leveraging the drastic surge in IoT devices, also referred to as Extreme Edge Devices (EEDs) [7], and exploiting their collective processing capabilities to further improve the performance.

978-1-6654-3540-6/22/$31.00 © 2022 Crown

Harvesting abundant yet underutilized computational resources at EEDs can break the monopoly caused by the fact that most Edge Computing (EC) paradigms, including MEC, are controlled solely by cloud service providers and/or network operators. Breaking this monopoly can democratize the edge and allow additional participants to build and manage their own edge cloud. In addition, in EED-enabled computing environments, EEDs are recruited to amplify the compute resource pool, perform parallel computing, and enhance the task offloading service, which can enable further improvement of the delay. However, achieving network objectives, such as low latency, reliable communication, and efficient computing, relies heavily on effective network design, analysis, and optimization, where a combined communication and computation perspective must be taken into consideration.

Task offloading in MEC environments is largely dependent on the availability and reachability of the available resources, as well as their resilience to failures [8]. Service interruptions triggered by failures of physical machines (PMs) and virtual machines (VMs) are addressed in [9]. In [10], the scalability of the network is explored in wireless-based task offloading, and both the communication and computation performance bounds are determined. The work in [11] analyzes task offloading under heterogeneous computational resources by estimating the network-wide outage probability. To perform energy-efficient task offloading, the spatial and temporal network parameters are considered in [12].

The aforementioned body of works either adopt a dependability perspective of the network [8], [9], or a spatiotemporal one [10]–[12]. A combined view of both perspectives is provided by the spatiotemporal framework presented in [13], where the authors consider the joint limitation of network interference and parallel computing by multiple failure-prone VMs that reside on the same edge server. However, feasible and dependable task execution that accounts for the joint device-to-device (D2D) communications under network-wide interference as well as the parallel computing at the EEDs has been overlooked.

Motivated by the above, we propose a spatiotemporal analysis that investigates the total task response delay at EEDs. Consider a computational task that can be divided into smaller slices called jobs to be offloaded at EEDs for faster execution

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10001321

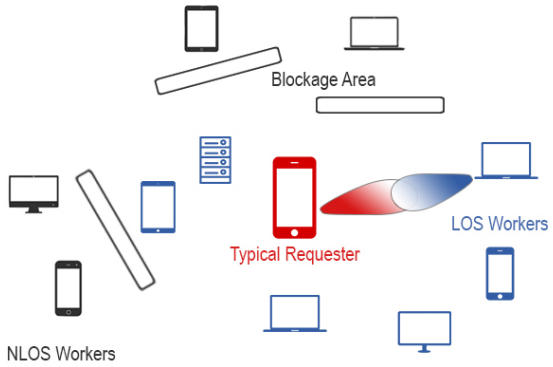


Fig. 1: System illustration

and less computational delay at each device. Consequently, the response delay includes a) the D2D communications delay to recruit and offload the jobs to the EEDs, and b) the computation delay to execute the offloaded jobs at the recruited EEDs. To this end, an absorbing continuous time Markov chain (ACTMC) is constructed to track the sequential recruitment of the EEDs via D2D communications as well as the parallel task execution at the recruited EEDs. To capture the interwoven communication and computation delays, the recruitment rate of the ACTMC is computed via tools from stochastic geometry to account for the D2D communications success probability. The results reveal the existence of an optimal number n^* of EEDs that minimizes the response delay. Going beyond n^* leads to an overwhelming communication latency that dominates the reduced computation latency.

II. System Model

The computationally capable EEDs, denoted hereafter as workers, are modeled via a Poisson point process (PPP) $\Phi_w \subset \mathbb{R}^2$ with intensity ν_w. The EEDs offer computational services to resource-constrained devices (e.g., IoT), which hereafter are denoted as requesters. The requesters are spatially distributed according to an independent PPP $\Phi_r \subset \mathbb{R}^2$ with intensity ν_r. There is an *edge orchestrator*, which can be a base station or an access point, which organizes the transactions between workers and requesters. In particular, the EEDs register their availability at the edge orchestrator, which in turns informs each requester about the availability of proximate EEDs. It is assumed that $\nu_w \gg \nu_r$, and hence, the edge orchestrator avoids conflicting requests to the same worker. To utilize parallel computing and reduce response delay, the requester divides each computational task into n smaller and equivalent jobs to be offloaded and executed at n different EEDs. Due to the heterogeneity of the computational powers of the EEDs, the finishing time of each job is exponentially distributed with mean $1/n\mu_f$, where μ_f is the task execution rate if computed at a single worker.

In compliance with 5G and beyond systems, the requesters utilize millimetre wave (mmW) for D2D communications in order to offload jobs to their proximate workers. The high vulnerability of mmW communications to blockage is taken into consideration via the general line of sight (LOS) ball blockage model [14], [15]. That is, devices within the distance of R_L from the requester are LOS devices, and otherwise, are considered non LOS (NLOS) devices. Distance dependent power-law path-loss is considered with exponents α_L and α_N for LoS and NLOS devices, respectively. All transmissions experience Nagakami multi-path fading. Hence, the channel power gains have independent and identical gamma distribution parameter N_L for LoS devices and parameter N_L for NLOS devices. We also ignore the fading in the frequency selective, as measurements show that the delay spread is generally small [16]. Also, results indicated that small-scale fading at mmWave is less severe than that in conventional systems when narrow beam antennas are used [16]. Thus, we can use a large Nakagami parameter N_L to approximate the small-variance fading as found in the LOS case.

Universal frequency reuse and constant transmit power P is utilized via all requesters. The requester and workers deploy antenna arrays for mmW beamforming. The array patterns are approximated by the sectored antenna model with main lobe gain of M_x, side lobe gain of m_x, and 3 dB beamwidth of θ_x, where the subscript $x \in \{w, r\}$ to differentiate between the antenna patterns of the requesters and workers. Without loss of generality, we consider that the requester is located at the origin and can establish D2D links with proximate LOS EEDs only. Perfect antenna alignment is considered for the intended D2D link and uniform random antenna alignment is considered for the interfering links. A pictorial illustration of the system model is shown in Figure 1.

The requester is assumed to have one task that is sliced into n smaller and equal tasks called $jobs$. The jobs are encapsulated into n packets that are transmitted via D2D communications to n different proximate workers. Since a single mmW interface is available at the requester, the workers are recruited one after another in a sequential fashion. The workers are selected randomly among the list of available LOS EEDs provided by the edge orchestrator. The communication between the requester and worker is subject to errors, and hence, the requester may need several attempts to successfully deliver the job packet and recruit a worker. The time required for each packet transmission attempt via D2D communications is τ_c seconds. The worker starts the computation immediately upon the successful reception of the job. The requester is then notified to proceed with offloading the pending jobs to other available LOS workers. All notification are assumed to be sent over perfect feedback channel.

III. Spatiotemporal Analysis

The job is correctly received at the worker if the signal-to-interference-plus-noise ratio (SINR) is above a given threshold ξ. Otherwise, the job has to be re-transmitted to another worker. Hence, the first step to investigate the response delay is to find the D2D communication success probability to recruit a randomly selected LOS worker. Such probability is then utilized within an ACTMC to find the total response delay.

TABLE I: Directivity gain probability and value

k	1	2	3
a_k	M_w^2	$M_w m_w$	m_w^2
b_k	c^2	$2c(1-c)$	$(1-c_r)^2$

The successful D2D transmission of the job can be expressed as

$$p_s = \mathbb{P}\left\{\text{SINR} > \xi\right\} = \mathbb{P}\left\{\frac{Ph_0 M_r M_w C_L r_0^{-\alpha_L}}{\sigma^2 + I_N + I_L} > \xi\right\}, \quad (1)$$

where h_0 is the intended channel power gain, C_L is the intercept of the LOS channel, r_0 is the distance between the requester and the intended LOS worker, I_L is the aggregate interference from other active LOS requesters, I_N is the aggregate interference from other active NLOS requesters, and σ^2 is the ambient noise power. Let $\Phi_L \subset \Phi_r$ and $\Phi_N = \Phi_r \setminus \{(\Phi_L) \cup (0,0)\}$ be the point process of the LOS and NLOS requesters seen from the origin, respectively. Then, the LOS and NLOS interference terms are expressed as

$$I_L = \sum_{i>0:\mathbf{x_i}\in\mathbf{\Phi_L}} h_i D_i C_L \left\|\mathbf{x_i}\right\|^{-\alpha_L}, \quad (2)$$

and

$$I_N = \sum_{i>0:\mathbf{y_i}\in\mathbf{\Phi_N}} g_i D_i C_N \left\|\mathbf{y_i}\right\|^{-\alpha_N}, \quad (3)$$

where h_i is the i^{th} LOS interfering link channel power gain, g_i is the i^{th} NLOS interfering link channel power gain, C_N is the intercept of the NLOS channel, $\|\cdot\|$ is the Euclidean norm, and D_i is the antenna gain width for the i^{th} interfering requester in Φ_L or Φ_N. Due to the sectored antenna model along with the uniformly random antenna alignment, D_i is a discrete random variable with the probability distribution defined as $\mathbb{P}\{D_i = a_k\} = b_k$ with k $\in$ {1, 2, 3}, where a_k and b_k are constants defined in Table I and $c = \theta_r/2\pi$.

The D2D transmission success probability given in (1) is characterized in the following lemma.

Lemma 1: The spatially averaged successful recruitment probability via mmW D2D communications for a randomly selected LOS worker out of Φ_w is given by

$$p_s = \int_0^{R_L} \sum_{n=1}^{N_L} \binom{N_L}{n} \frac{2r(-1)^{n+1} e^{M_n(\xi)\sigma^2 - W_n(\xi) - Z_n(\xi)}}{R_L^2} dr \quad (4)$$

where $M_n(\xi) = -\frac{\eta_L n r_0^{\alpha_L}\xi}{C_L M_r M_w}$, while $W_n(\xi)$ and $Z_n(\xi)$ are given by

$$W_n(\xi) = 2\pi\nu_r b_k \int_0^{R_L} \left(1 - \frac{1}{\left(1 + \frac{\eta_L \bar{a}_k n \xi (\frac{r_0}{x})^{\alpha_L}}{N_L}\right)^{N_L}}\right) x dx,$$

and

$$Z_n(\xi) = 2\pi\nu_r b_k \int_{R_L}^{\infty} \left(1 - \frac{1}{\left(1 + \frac{n_L \bar{a}_k n \xi C_N r_0^{\alpha_L}}{C_L x^{\alpha_N} N_N}\right)^{N_N}}\right) x dx.$$

Proof: This lemma can be proved by following the systematic stochastic geometry analysis through the probability generating functional of the PPP as in [14], [15]. ■

The D2D successful recruitment probability given in Lemma 1 is a core building block of the ACTMC that tracks the jobs offloading and execution. The states of the ACTMC is $\mathcal{S} = \{\mathbf{z} = (x_f, x_c) | \sum_j x_j \leq n; j \in \{f, c\}\}$, where $x_f \in \{0, 1, 2, \cdots, n\}$ denotes the number of workers that have finished their assignment and $x_c \in \{0, 1, 2, \cdots, n\}$ denotes the number of recruited workers that are actively executing the job. For each task, the ACTMC starts at the state $\mathbf{z} = (0, 0)$ where the requester has a task that is sliced to n jobs but has not yet recruited any worker. Each time the requested succeeds to recruit a LOS worker via mmW D2D transmission, a transition from state $\mathbf{z}_i = (x_f, x_c)$ to $\mathbf{z}_j = (x_f, x_c+1)$ occurs. Each time a worker is retired because of a job completion, a transition from state $\mathbf{z}_i = (x_f, x_c)$ to $\mathbf{z}_j = (x_f + 1, x_c - 1)$ occurs. Since the requester needs only n workers, then $x_c + x_f \leq n$ and $\mathbf{z} = (n, 0)$ is the absorbing state that implies the termination of the ACTMC. Following the aforementioned criterion, jobs offloading and execution at the EEDs can be tracked with an ACTMC with the following two-level hierarchical generator matrix

$$\mathbf{Q} = \begin{array}{c} \\ x_f \\ 0 \\ 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & \cdots & n \end{array} \\ \begin{pmatrix} \mathbf{K_0} & \mathbf{H_{0,1}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{K_1} & \mathbf{H_{1,2}} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K_2} & \mathbf{H_{2,3}} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{K_{n-1}} & \mathbf{H_{n-1,n}} \\ 0 & \cdots & 0 & 0 & 0 & 0 \end{pmatrix} \end{array},$$

where $\mathbf{Q}$ is a block matrix of size $(n+1) \times (n+1)$ that tracks the number of finished workers x_f. Since the task is finished upon the completion of the n jobs, then the state $x_f = n$ is the absorbing state that indicates the termination of the edge computing. Within each level of $\mathbf{Q}$, the sub-matrices $\mathbf{K}_m$ and $\mathbf{H}_{m,m+1}$ track the number of recruited workers x_c. Exploiting that fact that $x_c + x_f \leq n$, the matrix $\mathbf{H}_{m,m+1}$ is of size $(n-m) \times (n-m-1)$ that tracks x_c due to the completion of a job by any of the workers. Let $\mathbf{H}_{m,m+1}(i,j)$, with $i \in \{0, 1, 2, \cdots, n-m\}$ and $j \in \{0, 1, 2, \cdots, n-m-1\}$, denote that (i,j)-th element of the matrix $\mathbf{H}_{m,m+1}$. Then, due to the parallelized computing at the EEDs along with the fact that only one worker can finish at a given instance, the matrix $\mathbf{H}_{m,m+1}$ is given by

$$\mathbf{H}_{m,m+1}(i,j) = \begin{cases} i\mu_f, & i = j+1 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Using similar argument, the matrix $\mathbf{K}_m$ is of size $(n-m+1) \times (n-m+1)$ that tracks x_c upon recruiting new workers. Let $\mathbf{K}_m(i,j)$, with $i,j \in \{0, 1, 2, \cdots, n-m\}$ denote that (i,j)-th element of the matrix $\mathbf{K}_m$. Then, due to the sequential worker

recruitment, we have

$$\mathbf{K}_m(i,j) = \begin{cases} -(\lambda_h + i\mu_f), & i = j \ \ \& \ \ i < n-m \\ \lambda_h, & i = j-1 \ \ \& \ \ i < n-m \\ -(n-m)\mu_f, & i = j = n-m \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\lambda_h = p_s/\tau_c$ is the recruiting rate, p_s is the D2D transmission success probability given in (4), and τ_c is the time required for each D2D transmission attempt.

The task response time cannot be directly obtained for the matrix $\mathbf{Q}$. Instead, we first need to obtain the embedded discrete time Markov chain (EDTMC) of $\mathbf{Q}$ and the average sojourn time at each state. The EDTMC of $\mathbf{Q}$ is given by

$$\mathbf{P} = \begin{pmatrix} \mathcal{K}_{\mathbf{0}} & \mathcal{H}_{\mathbf{0,1}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathcal{K}_{\mathbf{1}} & \mathcal{H}_{\mathbf{1,2}} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{K}_{\mathbf{2}} & \mathcal{H}_{\mathbf{2,3}} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathcal{K}_{\mathbf{n-1}} & \mathcal{H}_{\mathbf{n-1,n}} \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{7}$$

where $\mathcal{K}_m$ and $\mathcal{H}_{m,m+1}$ track the transition probabilities due to worker recruitment and job completion, respectively. The matrices $\mathcal{K}_m$ and $\mathcal{H}_m$ are given by

$$\mathcal{K}_m(i,j) = \begin{cases} \frac{\lambda_h}{\lambda_h + i\mu_f} & i = j-1 \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

and

$$\mathcal{H}_{m,m+1}(i,j) = \begin{cases} \frac{i\mu_f}{\lambda_h + i\mu_f}, & i = j+1 \ \ \& \ \ i < n-m \\ 1, & i = n-m, j = n-m-1 \\ 0, & \text{otherwise} \end{cases}. \tag{9}$$

Let $x_{c_i} \in \mathbf{z}_i$ be the number of recruited workers in state $\mathbf{z}_i$, then the average sojourn time $t_{\mathbf{z}_i,\mathbf{z}_j}$ is given by

$$t_{\mathbf{z}_i,\mathbf{z}_j} = \begin{cases} \frac{1}{x_{c_i}\mu_f}, & \text{if the transition from } \mathbf{z}_i \text{ to } \mathbf{z}_j \\ & \text{is due to job completion} \\ \frac{1}{\lambda_h}, & \text{if the transition from } \mathbf{z}_i \text{ to } \mathbf{z}_j \\ & \text{is due to worker recruitment} \end{cases} \tag{10}$$

Equipped with $\mathbf{P}$ and $t_{\mathbf{z}_i,\mathbf{z}_j}$, the task response time is given in the following theorem.

Theorem 1: The task response time in the extreme edge computing networks with mmW D2D communications and n workers is given by

$$T_A = \alpha(\mathbf{I} - \mathbf{P}_T)^{-1}\mathbf{w}, \tag{11}$$

TABLE II: Simulation Parameters

Parameter	Value
Workers Intensity (ν_w) / 10 km^2	$7 * 10^{-4}$
Requester Intensity (ν_r) / 10 km^2	$1 * 10^{-4}$
LOS and NLOS path loss exponent (α_L, α_L)	2, 4
Fading values for LOS and NLOS (N_L, N_N)	3, 2
Noise (σ^2)	-114 dBm
Maximum radius for LOS devices (R_L)	100m
SINR coverage probability threshold (ξ)	-10 dB
D2D communication time (τ_c)	1 second
Task finishing rate (μ_f)	0.02 task/second
Number of task slices (n)	5 slices

where $\alpha = [1, 0, 0, \cdots, 0]$, $\mathbf{I}$ is the identity matrix, $\mathbf{P}_T$ is the transient state of $\mathbf{P}$ given in (7), which is obtained by excluding the last row and column of $\mathbf{P}$. The column vector $\mathbf{w}$ contains the average sojourn time in states $\mathbf{z}_i$ that is given by $w_{\mathbf{z}_i} = \sum_{\mathbf{z}_j} \mathbf{P}(\mathbf{z}_i, \mathbf{z}_j) t_{\mathbf{z}_i,\mathbf{z}_j}$, where $\mathbf{P}(\mathbf{z}_i, \mathbf{z}_j)$ is the transition probability from state $\mathbf{z}_i$ to $\mathbf{z}_i$.[1]

Proof: Let $\mathcal{S}_A = \mathcal{S} \setminus (n, 0)$ be the entire state space of the ACTMC excluding the absorbing state. Then, the time to absorption from a state $\mathbf{z}_i \in \mathcal{S}_A$ is given by

$$T_{\mathbf{z}_i} = \sum_{\mathbf{z}_j \in \mathcal{S}_A} \mathbf{P}(\mathbf{z}_i, \mathbf{z}_j)(t_{\mathbf{z}_i,\mathbf{z}_j} + T_{\mathbf{z}_j}). \tag{12}$$

After some manipulations, the expression in (12) can be written in the vector form for all states $\mathbf{z}_i \in \mathcal{S}_A$ as follows

$$\mathbf{T} = (\mathbf{I} - \mathbf{P}_T)^{-1}\mathbf{w}, \tag{13}$$

where $\mathbf{I}$ is the identity matrix. Given that the task execution starts at state $\mathbf{z}_0 = (0, 0)$, the task response delay is the first element in $\mathbf{T}$, which can be obtained by multiplying (13) with α as given in (11). ■

IV. Numerical Results

This section provides numerical and simulations results to validate the developed spatiotemporal model and illustrate the trade-off between the computation value and communication cost in extreme edge computing networks. Unless otherwise specified, the list of underlying network parameters utilized in this section is summarized in Table II. The Monte Carlo simulation are conducted over an area of 10 km^2. In each simulation run, a requester in the origin utilizes D2D communication to recruiter proximate LOS workers and the successful recruitment probabilities as well as the task response delay are recorded. The simulation results are then averaged over 10^5 runs.

Figure 2 shows the successful recruitment probability p_x as a function of the desired SINR threshold ξ for different values of the radius R_L that encloses LoS devices. The close

[1]In consistence with the hierarchical structure of $\mathbf{P}$, we utilize two dimensional indexing for its elements. Particularly, $\mathbf{P}(\mathbf{z}_i, \mathbf{z}_i) = \mathbf{P}\big((x_{f_i}, x_{c_i}), (x_{f_j}, x_{c_j})\big)$ is the (x_{c_i}, x_{c_j}) element of the matrix (x_{f_i}, x_{f_j}) sub-matrix in $\mathbf{P}$.

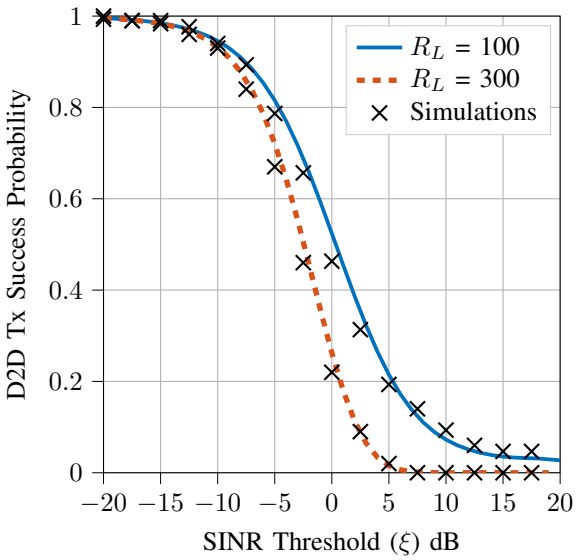


Fig. 2: D2D Tx success probability vs SINR threshold ξ

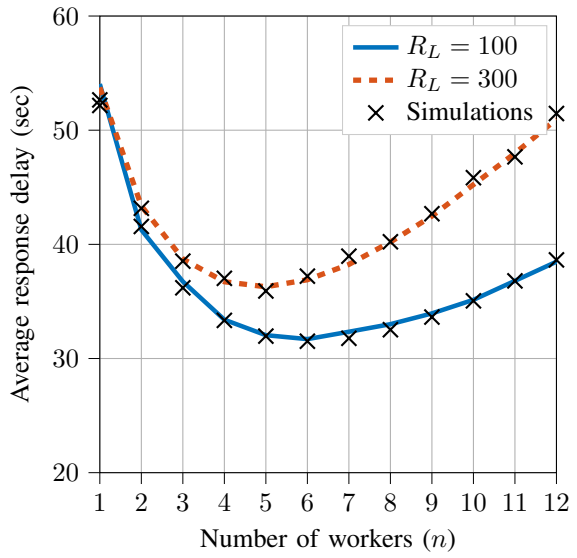


Fig. 3: Task response time vs the number of recruited workers

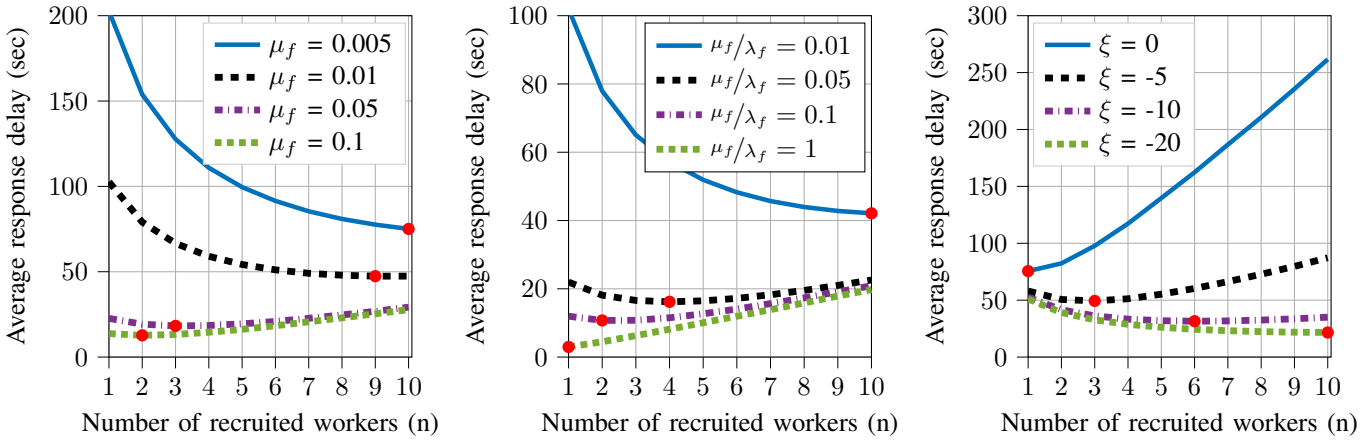
match demonstrated between the simulation and the proposed analytical framework validates Lemma 1. The figure shows that the successful recruitment probability p_x is inversely proportional to ξ due the increased link quality requirement. Hence, increasing x_i leads to higher number of attempts to successfully offload a task to a worker, which increases the communication cost. The figure also shows that a larger R_L also increases the communication cost due to i) the higher probability of longer D2D distance between the requested and the workers; and ii) the increases interference from other LOS requesters.

Figure 3 depicts the system performance in terms of the average task response delay over varying number of recruited workers given different values of R_L. The simulation and the proposed analytical framework closely coincide, which validates Theorem 1. The figure reveals an important trade-off between the communication cost and the computation value in extreme edge systems. As the number of recruited workers increases, the total communication time increases, whereas the total computation time decreases. This is due to the increase in the number of collaborating devices among which the task is divided, and with which communication occurs. As a result, the average task response delay continues to decrease as long as the reduction in computation delay is significantly predominant. This persists until reaching a point beyond which the increase in communication delay becomes too intense that it dominates the reduction in computation delay, thus causing the task response delay to start increasing. This indicates that there is an optimal number of recruited workers that minimizes the task response delay.

Figure 4 shows that the optimal number of recruited workers significantly varies with the network parameters. Note that the red dots show the minimum task response time, which corresponds to the time at which the system reaches the optimal number of recruited workers. In this context, Figure 4(a) demonstrates that the optimal number of workers depends on the task size, which is depicted via the rate μ_f. Lower finishing rates implies larger tasks, and hence, more workers need to be involved. Figure 4(b) illustrates that the optimal number of worker depends on the relative values of λ_f and μ_f. The case of $\mu_f/\lambda_f = 1$ implies that the communication delay is equal to the computational delay, and hence, one worker is sufficient to do the task. A lower ratio μ_f/λ_f implies faster recruitment rate when compared to the computation rate, and hence, more workers can be recruited to decrease the task response delay. Figure 4(c) shows that increasing the threshold ξ increases the communication cost, and hence, less number of workers is preferred due to the dominating communication delay.

Figure 6 and 5 show the average delay from our model, along side the response delay obtained by offloading the task to the an MEC PM. The task used here is $\mu_f = 0.007$ and $\nu_r = 2{*}10^{-4}$ (the rest of the parameters are specified in Table II). The requester will not divide the task before sending it, as it will be computed in the PM as a hall. The PM computational power is ten times more than the EEDs computational power.

Figure 5 shows the average response delay using three different PM congestion cases: (a) the PM is currently not serving any other users ($\nu_{r_{MEC}} = 0$), in that case, the total task delay will be less than the optimal value when the task is offloaded to the EEDs, due to the availability in the computational resources at the PM. (b) the number of served requesters is half of the available requesters ($\nu_{r_{MEC}} = \nu_r$), but the PM is not fully congested, this will lead to increasing the total task response delay, which happened because of the increase in the computational time due to the increase of the demand on the PM. (c) the PM is surfing all the available requesters ($\nu_{r_{MEC}} = 2\nu_r$) and the PM is fully congested, in that case, offloading the task to the EEDs will be better than



(a) Over varying task finishing rate (μ_f), (b) Over varying finishing and recruiting ratio (μ_f/λ_f), (c) Over varying SINR threshold (ξ),

Fig. 4: Task response time (T_A) vs the number of recruited workers (n) for different system parameters.

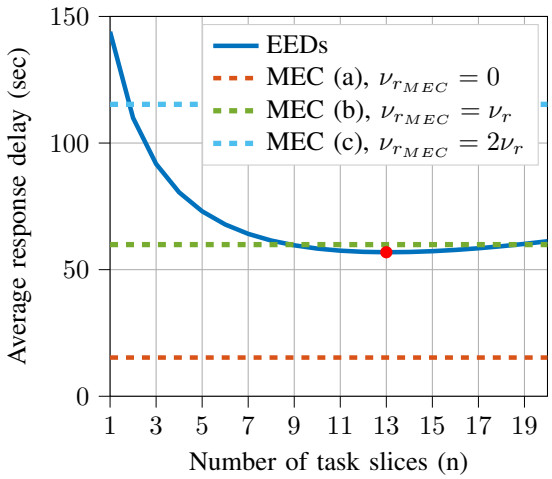


Fig. 5: MEC and EEDs average response delay using varying BS congestion parameters

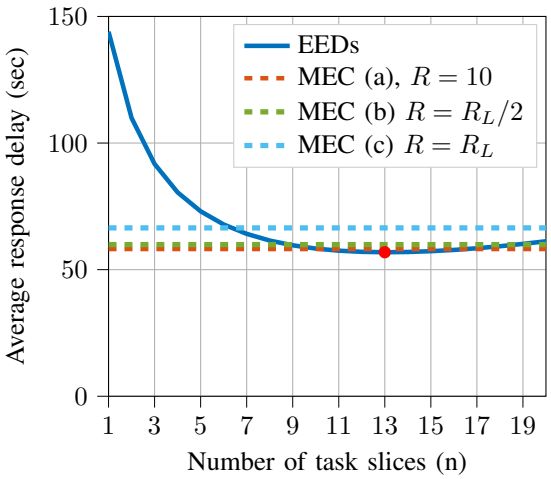


Fig. 6: MEC and EEDs average response delay using varying BS distance parameters

offloading it to the MEC, due to the response delay the MEC will take compared to the delay from the EEDs.

Figure 6 shows the average response delay using three different distance value with $\nu_{r_{MEC}} = \nu_r$: (a) the distance between the requester and the BS $R = 10$, so the communication time will be at its best, as the BS is close to the requester. (b) the distance between the BS and the requester $R = R_L/2$, this will increase the communication time a little due to the increase in the distance, which aggravates the impact of fading and interference. Finally, (c) the BS is located on the farthest LoS point from the requester $R = R_L$, the communication time will be at its worst due to the low successful recruitment probability. Combining the three cases, we see that the distance does really increase the average response delay that much, and the communication time in the MEC case does not have a big effect.

V. Conclusion and Future Work

This paper presents a novel spatiotemporal framework that characterizes the task response time in extreme edge computing networks. The developed model accounts for the interwoven communication and computation delays by constructing an absorbing continuous time Markov chain (ACTMC) to track the sequential offloading and parallel execution of tasks at extreme edge devices (EEDs), where the offloading rate is obtained via stochastic geometry analysis. The former is used to capture the sequential recruitment of EEDs via D2D communications, along with the process of parallel task execution at such devices. The latter is adopted to model the D2D communications success probability. Numerical results validate the analysis and reveal the existence of an optimal number (n^*) of recruited EEDs that minimizes the underlying task response time. Operating below n^* leads to underutilized

edge computational resources, and hence, prolongs the task response delay. Exceeding n^* leads to a dominating communication delay that also prolongs the response delay. To this end, it is demonstrated that the optimal number of recruited EEDs significantly varies with the network parameters. In the future, we plan on handling service interruptions caused by failure. We also plan to come up with a new hiring technique, where EEDs hiring will depend on the distance of the devices.

VI. Aknowlodgment

This research is supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant number: ALLRP 549919-20.

References

[1] J. Steward, "21 internet of things statistics, facts & trends for 2021„" 2021. [Online]. Available: https://findstack.com/internet-of-things-statistics/

[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[3] L. Peterson, T. Anderson, S. Katti, N. McKeown, G. Parulkar, J. Rexford, M. Satyanarayanan, O. Sunay, and A. Vahdat, "Democratizing the network edge," *SIGCOMM Comput. Commun. Rev.*, vol. 49, no. 2, p. 31–36, may 2019. [Online]. Available: https://doi.org/10.1145/3336937.3336942

[4] R. Olaniyan, O. Fadahunsi, M. Maheswaran, and M. F. Zhani, "Opportunistic edge computing: Concepts, opportunities and research challenges," *Future Generation Computer Systems*, vol. 89, pp. 633–645, 2018.

[5] Y. Sahni, J. Cao, S. Zhang, and L. Yang, "Edge mesh: A new paradigm to enable distributed intelligence in internet of things," *IEEE Access*, vol. 5, pp. 16 441–16 458, 2017.

[6] W. ZHANG, H. Flores, and P. HUI, "Towards collaborative multi-device computing," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2018, pp. 22–27.

[7] J. Portilla, G. Mujica, J.-S. Lee, and T. Riesgo, "The extreme edge at the bottom of the internet of things: A review," *IEEE Sensors Journal*, vol. 19, no. 9, pp. 3179–3190, 2019.

[8] S. Bagchi, M.-B. Siddiqui, P. Wood, and H. Zhang, "Dependability in edge computing," *Commun. ACM*, vol. 63, no. 1, p. 58–66, dec 2019. [Online]. Available: https://doi.org/10.1145/3362068

[9] R. Birke, I. Giurgiu, L. Y. Chen, D. Wiesmann, and T. Engbersen, "Failure analysis of virtual and physical machines: Patterns, causes and characteristics," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2014, pp. 1–12.

[10] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5225–5240, 2018.

[11] H.-S. Lee and J.-W. Lee, "Task offloading in heterogeneous mobile cloud computing: Modeling, analysis, and cloudlet deployment," *IEEE Access*, vol. 6, pp. 14 908–14 925, 2018.

[12] H. Ko, J. Lee, and S. Pack, "Spatial and temporal computation offloading decision algorithm in edge cloud-enabled heterogeneous networks," *IEEE Access*, vol. 6, pp. 18 920–18 932, 2018.

[13] M. Emara, H. ElSawy, M. C. Filippou, and G. Bauch, "Spatiotemporal dependable task execution services in mec-enabled wireless systems," *IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 211–215, 2021.

[14] S. Singh, M. N. Kulkarni, A. Ghosh, and J. G. Andrews, "Tractable model for rate in self-backhauled millimeter wave cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2196–2211, 2015.

[15] N. Deng, M. Haenggi, and Y. Sun, "Millimeter-wave device-to-device networks with heterogeneous antenna arrays," *IEEE Transactions on Communications*, vol. 66, no. 9, pp. 4271–4285, 2018.

[16] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5g cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.