Mobility-Aware Offloading with SINR Prediction and Selective Replication in Fog-IoT Networks

Rajasekhar Dasari, Satyajit Mohapatra, Sanjeet Kumar Nayak
Department of CSE, IIITDM Kancheepuram, Chennai-600127, India
Email: {cs22d0003, cs23d0003, sanjeetn}@iiitdm.ac.in

***Abstract*—The rise of mobile Internet of Things (IoT) applications has made reliable, low-latency task offloading to nearby fog nodes essential. However, user mobility, time-varying wireless links, short-packet transmission errors, and fog-queue congestion make the execution of delay-sensitive tasks challenging. Existing schemes typically address reliability, replication, or offloading separately, without jointly considering mobility-aware link variation. To address this gap, this paper proposes a mobility-aware reliability-driven offloading framework for fog-enabled IoT networks. The proposed scheme combines a lightweight multilayer perceptron (MLP)-based SINR predictor with a Lyapunov drift-plus-penalty (LDPP) controller to select among local execution, single-fog offloading, and replicated-fog execution. Simulation results show that the proposed method reduces task delay, reliability violations, and energy consumption by 17%, 13%, and 41%, respectively as compared to existing scheme.**

***Index Terms*—SINR, Fog computing, multilayer perceptron, mobility, reliability.**

I. Introduction

Fog-enabled IoT has emerged as an effective paradigm for supporting latency-sensitive and mission-critical applications such as intelligent transportation, industrial monitoring, and mobile sensing. By bringing computation resources closer to end devices, fog computing reduces end-to-end delay and improves responsiveness compared with cloud-centric execution. However, ensuring both low latency and high reliability in fog-IoT systems with mobility remains challenging because task completion depends jointly on time-varying wireless links, fog queueing delay, and the dynamic availability of computing resources [1]. These challenges become more severe in finite-blocklength (FBL) communication and under user mobility [2]. Under FBL transmission, decoding error probability is non-negligible, and reducing transmission time too aggressively can degrade reliability. At the same time, mobility causes rapid channel variation, while fog congestion increases waiting time. This may lead to deadline violation even when communication succeeds. Although task replication can improve reliability, excessive or static replication wastes bandwidth, energy, and computing resources, which may further increase queue congestion. Therefore, the key problem is to design an online task offloading policy that can proactively choose local execution, single-fog offloading, or selective replication while accounting for both wireless uncertainty and fog congestion.

Existing reliability-aware mobile edge/fog computing studies based on FBL communication and computation-delay modelling provide rigorous analytical foundations for end-to-end reliability analysis [3], [4]. Nevertheless, their focus is mainly on communication/computation time allocation rather than mobility-aware execution-location selection or selective replication. Replication-based studies improve service reliability through redundant execution and adaptive redundancy control [2], [5], [6]. However, most of them rely on static replication, delay-sample-based adaptation, or scenario-specific reliability estimation rather than proactive Signal-to-Interference-plus-Noise Ratio (SINR) prediction integrated with fog-queue-aware online control. Learning-based reliable offloading has also been explored in UAV-assisted, vehicular, and multigateway edge systems, where dynamic adaptation is emphasised. However, these approaches are usually tailored to specific architectures and do not jointly consider mobility-aware SINR variation, finite-blocklength transmission reliability, and selective replication [7], [8], [9]. Therefore, a unified online framework that combines mobility awareness, proactive SINR prediction, selective replication, fog queue dynamics, and long-term reliability control remains largely unexplored.

Motivated by this gap, this paper proposes a mobility-aware reliability-driven task offloading framework for fog-enabled IoT networks. In particular, this paper makes three main contributions. First, we develop a mobility-aware fog-IoT system model that captures time-varying wireless links, short-packet transmission, queue-aware fog execution, and selective replication. Second, we formulate an online reliability-aware offloading problem and propose an ML-assisted LDPP control framework. This framework proactively predicts SINR, regulates long-term delay and reliability violations through physical and virtual queues. Third, we design a low-complexity greedy per-task algorithm that efficiently approximates the mixed-integer per-slot optimization by screening candidate fog nodes and activating replication only when needed.

II. System Model

A. System Architecture

We consider a mobility-aware fog-enabled IoT network with a set of IoT devices $\mathcal{I} = \{1, \ldots, |\mathcal{I}|\}$ and a set of fog nodes $\mathcal{F} = \{1, \ldots, |\mathcal{F}|\}$. Time is slotted as $t \in \{0, 1, 2, \ldots\}$ with slot duration Δ seconds. At most one task can arrive from device $i \in \mathcal{I}$ during a slot. Let $a_i(t) \in \{0, 1\}$ denote the task-arrival indicator of device i at slot t, where $\Pr\big(a_i(t) = 1\big) = \lambda_i \Delta,\ \lambda_i \Delta \leq 1$. Accordingly, the set of active devices at slot t is $\mathcal{A}(t) \triangleq \{i \in \mathcal{I} : a_i(t) = 1\}$. Device i follows a Gauss–Markov mobility model [10]. Let

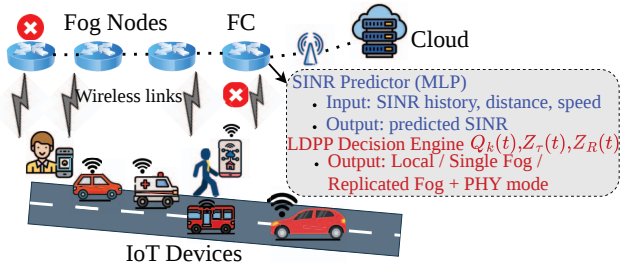


Fig. 1: Mobility-aware FC-IoT framework

$\mathbf{p}_i(t) \in \mathbb{R}^2$ and $\mathbf{p}_k \in \mathbb{R}^2$ denote the locations of device i and fog node k, respectively. The instantaneous distance between them is $d_{i,k}(t) = \|\mathbf{p}_i(t) - \mathbf{p}_k\|$. For each active device $i \in \mathcal{A}(t)$, the generated task is represented as $\mathcal{T}_i(t) \triangleq \big(B_i^{\text{in}}(t),\, B_i^{\text{out}}(t),\, C_i(t),\, \tau_i(t),\, R_i^{\min}(t)\big)$, where $B_i^{\text{in}}(t)$ and $B_i^{\text{out}}(t)$ denote the input and output sizes (bits), $C_i(t)$ is the required number of CPU cycles, $\tau_i(t)$ is the hard deadline (seconds), and $R_i^{\min}(t)$ is the minimum end-to-end reliability requirement. Each task may be executed locally, offloaded to a single fog node, or selectively replicated to two distinct fog nodes. Selective replication is used to improve reliability in delay-sensitive, reliability-critical fog-IoT applications. For replicated execution, timeliness is enforced through the deadline constraint, while reliability is satisfied when at least one fog node completes successfully. All decisions are made online based on the currently observed system state, including task attributes, queue backlogs, and predicted link qualities. The mobility-aware FC-IoT architecture is shown in Fig. 1.

B. Wireless Channel and SINR Model

We consider bandwidth W for IoT–fog communication. The complex baseband channel on link (i,k) at slot t is modeled as $h_{i,k}(t) = \sqrt{G_0 d_{i,k}(t)^{-\kappa}}\, g_{i,k}(t)$, where G_0 is the pathloss constant, κ is the pathloss exponent, and $g_{i,k}(t) \sim \mathcal{CN}(0,1)$ models Rayleigh fading [8].

Let P_i^{UL} denote the uplink radiated transmit power of device i. Then, the uplink SINR at fog node k is

$$\gamma_{i,k}(t) = \frac{P_i^{\text{UL}} |h_{i,k}(t)|^2}{\sigma^2 + \sum_{u \in \mathcal{I}\setminus\{i\}} b_u(t)\, P_u^{\text{UL}} |h_{u,k}(t)|^2}, \qquad (1)$$

where σ^2 is the noise power and $b_u(t) \in \{0,1\}$ indicates whether device u transmits on the same uplink resource at slot t [8]. The downlink SINR, denoted by $\gamma_{i,k}^{\text{DL}}(t)$, can be defined similarly.

C. Short-Packet Transmission with Capped Retransmissions

To model short-packet transmission under deadline and reliability requirements, we adopt the finite-blocklength (FBL) model [2]. For SINR γ, blocklength n (channel uses), and per-attempt decoding error probability ε, the achievable spectral efficiency (bits/channel-use) is approximated as $r(\gamma, \varepsilon, n) \approx \log_2(1+\gamma) - \sqrt{\frac{V(\gamma)}{n}}\, Q^{-1}(\varepsilon)$, where $Q^{-1}(\cdot)$ is the inverse Q-function and $V(\gamma) = \frac{\gamma(2+\gamma)}{(1+\gamma)^2}(\log_2 e)^2$ is the channel dispersion.

Each uplink/downlink transmission is allowed at most N_{tx} attempts. For a per-attempt decoding error probability ε, the expected number of attempts under a truncated geometric model is $\Xi(\varepsilon, N_{\text{tx}}) \triangleq \mathbb{E}[N] = \sum_{j=1}^{N_{\text{tx}}} \varepsilon^{j-1} = (1-\varepsilon^{N_{\text{tx}}})/(1-\varepsilon)$.

Let $(n_{i,k}(t), \varepsilon_{i,k}^{\text{UL}}(t), \varepsilon_{i,k}^{\text{DL}}(t))$ denote the PHY parameters used on link (i,k) at slot t. These parameters are selected by the controller as described in Sec. II-E. The expected uplink and downlink transmission times, including capped retransmissions are given by

$$T_{i,k}^{\text{UL}}(t) = \Xi\big(\varepsilon_{i,k}^{\text{UL}}(t), N_{\text{tx}}\big)\, \frac{B_i^{\text{in}}(t)}{W\, r\big(\gamma_{i,k}(t), \varepsilon_{i,k}^{\text{UL}}(t), n_{i,k}(t)\big)}, \quad (2)$$

$$T_{i,k}^{\text{DL}}(t) = \Xi\big(\varepsilon_{i,k}^{\text{DL}}(t), N_{\text{tx}}\big)\, \frac{B_i^{\text{out}}(t)}{W\, r\big(\gamma_{i,k}^{\text{DL}}(t), \varepsilon_{i,k}^{\text{DL}}(t), n_{i,k}(t)\big)}, \quad (3)$$

where $n_{i,k}(t) = \sum_{m \in \mathcal{M}} y_{i,k,m}(t)\, n_m$, $\varepsilon_{i,k}^{\text{UL}}(t) = \sum_{m \in \mathcal{M}} y_{i,k,m}(t)\, \varepsilon_m^{\text{UL}}$, $\varepsilon_{i,k}^{\text{DL}}(t) = \sum_{m \in \mathcal{M}} y_{i,k,m}(t)\, \varepsilon_m^{\text{DL}}$.

D. Fog Computing, Queueing and Reliability Model

Fog node k has CPU speed f_k (cycles/s). Let $Q_k(t)$ denote the queued workload (in CPU cycles) at the fog node k at the beginning of slot t. Using a slot-level workload approximation, the queueing delay experienced by task $\mathcal{T}_i(t)$ when assigned to fog node k is $T_{i,k}^{\text{q}}(t) = Q_k(t)/f_k$, and its processing time is $T_{i,k}^{\text{proc}}(t) = C_i(t)/f_k$. Hence, the total latency at fog is

$$T_{i,k}^{\text{fog}}(t) = T_{i,k}^{\text{UL}}(t) + T_{i,k}^{\text{q}}(t) + T_{i,k}^{\text{proc}}(t) + T_{i,k}^{\text{DL}}(t). \qquad (4)$$

The workload queue evolves according to

$$Q_k(t+1) = \max\{Q_k(t) - f_k \Delta,\, 0\} + \sum_{i \in \mathcal{I}} x_{i,k}(t)\, C_i(t), \quad (5)$$

where $x_{i,k}(t) \in \{0,1\}$ indicates whether device i assigns its task to fog node k at slot t. If replication is used, two such indicators become active, and the required CPU cycles are appended to both selected fog queues. In replicated execution, the two replicas run in parallel, and the task is considered complete when the first successful replica finishes. Hence, the effective latency is $T_{i,\{k,l\}}^{\text{fog}}(t) = \min\Big\{T_{i,k}^{\text{fog}}(t),\, T_{i,l}^{\text{fog}}(t)\Big\}$.

Reliability Model: Each uplink/downlink transmission is allowed at most N_{tx} attempts. With per-attempt decoding error probabilities $\varepsilon_{i,k}^{\text{UL}}(t)$ and $\varepsilon_{i,k}^{\text{DL}}(t)$, the corresponding uplink and downlink success probabilities are $R_{i,k}^{\text{UL}}(t) = 1 - \big(\varepsilon_{i,k}^{\text{UL}}(t)\big)^{N_{\text{tx}}}$, $R_{i,k}^{\text{DL}}(t) = 1 - \big(\varepsilon_{i,k}^{\text{DL}}(t)\big)^{N_{\text{tx}}}$. Since successful offloading requires both uplink and downlink transmissions to succeed, the wireless link success probability is $R_{i,k}^{\text{L}}(t) = R_{i,k}^{\text{UL}}(t)\, R_{i,k}^{\text{DL}}(t)$.

Fog node k alternates between operational and failed states with failure rate α_k and repair rate β_k. Its steady-state availability is given by $\overline{A}_k = \beta_k/(\alpha_k + \beta_k)$. Accordingly, the end-to-end reliability of offloading task $\mathcal{T}_i(t)$ to fog node k is

$$R_{i,k}^{\text{T}}(t) = \overline{A}_k\, R_{i,k}^{\text{L}}(t). \qquad (6)$$

Here, reliability captures successful communication and fog-node availability, whereas timeliness is enforced separately through the deadline constraint. Further, if $\mathcal{T}_i(t)$ is replicated, then the resulting end-to-end success reliability is $R_{i,\{k,l\}}^{\text{T}}(t) = 1 - \big(1 - R_{i,k}^{\text{T}}(t)\big)\big(1 - R_{i,l}^{\text{T}}(t)\big)$.

Energy consumption Model: For offloading to fog node k, the IoT-device communication energy includes both the power amplifier (PA) and circuit energy. Let $\eta_{\mathrm{pa}} \in (0,1]$ denote the PA efficiency, and let $P_{i,\mathrm{circ}}^{\mathrm{tx}}$ and $P_{i,\mathrm{circ}}^{\mathrm{rx}}$ denote the transmit and receive circuit powers, respectively [11]. Then, the communication energy is $E_{i,k}^{\mathrm{comm}}(t) = (P_i^{\mathrm{UL}}/\eta_{\mathrm{pa}} + P_{i,\mathrm{circ}}^{\mathrm{tx}})T_{i,k}^{\mathrm{UL}}(t) + P_{i,\mathrm{circ}}^{\mathrm{rx}}\, T_{i,k}^{\mathrm{DL}}(t)$. The fog-node processing energy is $E_{i,k}^{\mathrm{proc}}(t) = P_k^{\mathrm{act}}\, T_{i,k}^{\mathrm{proc}}(t)$, where P_k^{act} denotes the active processing power of fog node k. Therefore, the total energy associated with single-fog execution is

$$E_{i,k}^{\mathrm{fog}}(t) = E_{i,k}^{\mathrm{comm}}(t) + E_{i,k}^{\mathrm{proc}}(t). \tag{7}$$

For replicated execution over nodes $\{k,m\}$, we use the conservative accounting is $E_{i,\{k,l\}}^{\mathrm{fog}}(t) = E_{i,k}^{\mathrm{fog}}(t) + E_{i,l}^{\mathrm{fog}}(t)$.

E. Problem Formulation

Let $x_{i,0}(t) \in \{0,1\}$ denote whether task $\mathcal{T}_i(t)$ is executed locally, and let $x_{i,k}(t) \in \{0,1\}$, $k \in \mathcal{F}$, denote whether the task is assigned to fog node k. Selective replication is captured by allowing up to two fog indicators to be active. We define the selected fog set as $\mathcal{S}_i(t) \triangleq \{k \in \mathcal{F} : x_{i,k}(t) = 1\},\ |\mathcal{S}_i(t)| \in \{0,1,2\}$.

Let $\mathcal{M}$ denote the finite set of available PHY modes. Each mode $m \in \mathcal{M}$ specifies a tuple $(n_m, \varepsilon_m^{\mathrm{UL}}, \varepsilon_m^{\mathrm{DL}})$. We introduce binary variables $y_{i,k,m}(t) \in \{0,1\}$ indicating whether mode m is selected on link (i,k). Exactly one PHY mode is selected for each active fog link is $\sum_{m\in\mathcal{M}} y_{i,k,m}(t) = x_{i,k}(t),\ \forall i \in \mathcal{A}(t),\ \forall k \in \mathcal{F}$. Accordingly, the effective PHY parameters on link (i,k) are $n_{i,k}(t) = \sum_{m\in\mathcal{M}} y_{i,k,m}(t)\, n_m$, $\varepsilon_{i,k}^{\mathrm{UL}}(t) = \sum_{m\in\mathcal{M}} y_{i,k,m}(t)\, \varepsilon_m^{\mathrm{UL}}$, $\varepsilon_{i,k}^{\mathrm{DL}}(t) = \sum_{m\in\mathcal{M}} y_{i,k,m}(t)\, \varepsilon_m^{\mathrm{DL}}$. These effective PHY parameters are relevant only when $x_{i,k}(t) = 1$.

Local execution: Let f_i denote the local CPU speed of device i and P_i^{cpu} its local processing power. If task $\mathcal{T}_i(t)$ is executed locally, then the delay and energy can be calculated as $T_i^{\mathrm{loc}}(t) = C_i(t)/f_i$, $E_i^{\mathrm{loc}}(t) = P_i^{\mathrm{cpu}}\, T_i^{\mathrm{loc}}(t)$. We assume that local execution is failure-free from the reliability perspective, i.e., $R_i^{\mathrm{loc}}(t) = 1$.

Task-level metrics: The task completion delay $T_i(t)$, total energy consumption $E_i(t)$, and end-to-end reliability $R_i(t)$ are defined as follows.

$$T_i(t) = \begin{cases} T_i^{\mathrm{loc}}(t), & x_{i,0}(t) = 1, \\ T_{i,k}^{\mathrm{fog}}(t), & \mathcal{S}_i(t) = \{k\}, \\ \min\{T_{i,k}^{\mathrm{fog}}(t),\, T_{i,l}^{\mathrm{fog}}(t)\}, & \mathcal{S}_i(t) = \{k,l\}. \end{cases}$$

$$E_i(t) = \begin{cases} E_i^{\mathrm{loc}}(t), & x_{i,0}(t) = 1, \\ E_{i,k}^{\mathrm{fog}}(t), & \mathcal{S}_i(t) = \{k\}, \\ E_{i,k}^{\mathrm{fog}}(t) + E_{i,l}^{\mathrm{fog}}(t), & \mathcal{S}_i(t) = \{k,l\}. \end{cases}$$

$$R_i(t) = \begin{cases} R_i^{\mathrm{loc}}(t) = 1, & x_{i,0}(t) = 1, \\ R_{i,k}^{\mathrm{T}}(t), & \mathcal{S}_i(t) = \{k\}, \\ 1 - \big(1 - R_{i,k}^{\mathrm{T}}(t)\big)\big(1 - R_{i,l}^{\mathrm{T}}(t)\big), & \mathcal{S}_i(t) = \{k,l\} \end{cases}$$

Objective Function: The per-slot weighted system cost is defined as

$$p(t) = \sum\nolimits_{i\in\mathcal{A}(t)} \big(T_i(t) + \omega E_i(t)\big). \tag{8}$$

where $\omega \geq 0$ is the tradeoff parameter converting energy consumption into delay-equivalent cost.

Let π denote an online control policy that maps the observed system state at slot t to the decision variables $\{x_{i,0}(t), x_{i,k}(t), y_{i,k,m}(t)\}$. The optimization problem is formulated as

$$\min_{\pi} \quad \limsup_{T\to\infty} \frac{1}{T} \sum\nolimits_{t=0}^{T-1} \mathbb{E}_\pi\big[p(t)\big] \tag{9a}$$

$$\text{s.t.} \quad x_{i,0}(t) + \sum_{k\in\mathcal{F}} x_{i,k}(t) \geq 1,\ \forall i \in \mathcal{A}(t),\ \forall t, \tag{9b}$$

$$\sum_{k\in\mathcal{F}} x_{i,k}(t) \leq 2\big(1 - x_{i,0}(t)\big),\ \forall i \in \mathcal{A}(t),\ \forall t, \tag{9c}$$

$$\sum_{m\in\mathcal{M}} y_{i,k,m}(t) = x_{i,k}(t),\ \forall i \in \mathcal{A}(t),\ \forall k \in \mathcal{F}, \tag{9d}$$

$$T_i(t) \leq \tau_i(t),\ \forall i \in \mathcal{A}(t),\ \forall t, \tag{9e}$$

$$R_i(t) \geq R_i^{\min}(t),\ \forall i \in \mathcal{A}(t),\ \forall t. \tag{9f}$$

Constraint interpretation: Constraint (9b) ensures that each active task is assigned to at least one execution location. Constraint (9c) enforces local execution exclusivity and allows replication to at most two fog nodes when local execution is not selected. Constraint (9d) selects exactly one PHY mode for each active fog link. Constraint (9e) imposes the hard deadline requirement. Constraint (9f) enforces the minimum end-to-end reliability requirement.

III. LDPP With ML-Based SINR Prediction

To handle time-varying wireless conditions and long-term delay and reliability requirements, we combine a lightweight multilayer perceptron (MLP) based SINR predictor with a Lyapunov drift-plus-penalty (LDPP) controller [8] [2].

a) SINR Prediction and Queue Updates: At the beginning of slot t, the controller observes the fog node queue status $\{Q_k(t)\}_{k\in\mathcal{F}}$, the active task set $\mathcal{A}(t)$, and recent channel measurements (e.g., SINR history) as described in Sec. II-B. For each IoT to fog channel link (i,k), it constructs the feature vector $\mathbf{z}_{i,k}(t) = \big[\gamma_{i,k}(t-1), \gamma_{i,k}(t-2), \ldots, \gamma_{i,k}(t-L), d_{i,k}(t), v_i(t)\big]$, where L is the history length, $d_{i,k}(t)$ is the device–fog distance, and $v_i(t)$ is the device speed. Using a lightweight MLP regressor $f_{\boldsymbol{\theta}}(\cdot)$ [8], the predicted SINR is obtained as

$$\widehat{\gamma}_{i,k}(t) = f_{\boldsymbol{\theta}}\big(\mathbf{z}_{i,k}(t)\big), \tag{10}$$

where the model parameters $\boldsymbol{\theta}$ are trained using the mean-squared error loss $\mathcal{L}_{\mathrm{MSE}} = \sum_{(i,k),t} \big(\gamma_{i,k}(t) - f_{\boldsymbol{\theta}}(\mathbf{z}_{i,k}(t))\big)^2$.

Using $\widehat{\gamma}_{i,k}(t)$, the controller computes the predicted finite-blocklength rate and the corresponding task-level metrics $\widehat{T}_i(t)$, $\widehat{E}_i(t)$, and $\widehat{R}_i(t)$ for all feasible execution options. The physical fog-workload queues evolve according to Eq. (5). To enforce the long-term delay and reliability requirements, we introduce two virtual queues with aggregate violations $g_\tau(t) = \sum_{i\in\mathcal{A}(t)} \big[\widehat{T}_i(t) - \tau_i(t)\big]_+$, $g_R(t) = \sum_{i\in\mathcal{A}(t)} \big[R_i^{\min}(t) - \widehat{R}_i(t)\big]_+$, and updates

$$Z_\tau(t+1) = \max\big\{Z_\tau(t) + g_\tau(t), 0\big\}, \tag{11}$$

$$Z_R(t+1) = \max\big\{Z_R(t) + g_R(t), 0\big\}. \tag{12}$$

b) LDPP-Based Online Control: Let $\Theta(t) = \big(\{Q_k(t)\}_{k\in\mathcal{F}}, Z_\tau(t), Z_R(t)\big)$ denote the queue state [2]. The quadratic Lyapunov function is

$$L(\Theta(t)) = \frac{1}{2}\Big(\sum\nolimits_{k\in\mathcal{F}} Q_k^2(t) + Z_\tau^2(t) + Z_R^2(t)\Big). \tag{13}$$

The controller minimizes an upper bound on the drift-plus-penalty term $\Delta(t) + V\,\mathbb{E}[p(t) \mid \Theta(t)]$, where $V > 0$ controls the tradeoff between cost minimization and queue stability. After applying the standard LDPP bound and removing terms independent of the control action, the controller minimizes the following per-slot metric is

$$\begin{aligned}\Psi(t) = V \sum_{i\in\mathcal{A}(t)} \big(\widehat{T}_i(t) + \omega\widehat{E}_i(t)\big) + \sum_{k\in\mathcal{F}} Q_k(t) \sum_{i\in\mathcal{A}(t)} x_{i,k}(t)C_i(t) \\ + Z_\tau(t) \sum_{i\in\mathcal{A}(t)} [\widehat{T}_i(t) - \tau_i(t)]_+ + Z_R(t) \sum_{i\in\mathcal{A}(t)} [R_i^{\min}(t) - \widehat{R}_i(t)]_+\end{aligned} \tag{14}$$

At each slot, the controller selects the feasible decision variables that minimize the metric in Eq. (14). To implement this online, we use the greedy per-task procedure in Algorithm 1. The greedy step avoids exhaustive search over all fog-node pairs and PHY modes, while small-scale exhaustive-search validation confirms that the obtained decisions remain close to the exhaustive-search solution. Lines 1–5 evaluate local execution and initialize it as the default decision. Lines 6–19 compute the LDPP score for all fog-node and PHY-mode combinations and retain the two best single-fog candidates. Lines 20–21 compare the best single-fog option with the current decision, while lines 22–28 evaluate selective replication over the two best candidates. Lines 29–31 apply the selected decision and update the physical and virtual queues. The resulting time complexity of Algorithm 1 is $\mathcal{O}(|\mathcal{A}(t)|\,|\mathcal{F}|\,|\mathcal{M}|)$.

Algorithm 1: LDPP Offloading With SINR Prediction

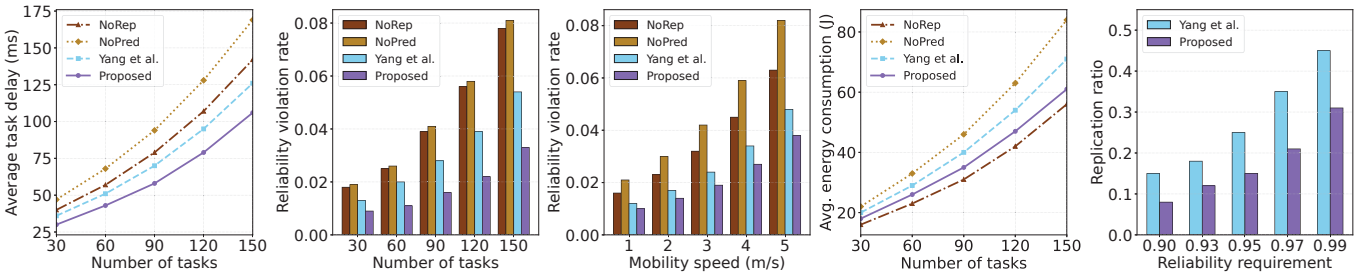
Input : $\{Q_k(t)\}_{k\in\mathcal{F}}$, $Z_\tau(t)$, $Z_R(t)$, $\mathcal{A}(t)$
Output: $\{x_{i,0}(t), x_{i,k}(t), y_{i,k,m}(t)\}$

1 **foreach** $i \in \mathcal{A}(t)$ **do**
2 $\quad\widehat{T}^\star \leftarrow T_i^{\mathrm{loc}}(t)$, $\widehat{E}^\star \leftarrow E_i^{\mathrm{loc}}(t)$, $\widehat{R}^\star \leftarrow 1$
3 $\quad S^\star \leftarrow V(\widehat{T}^\star + \omega\widehat{E}^\star) + Z_\tau(t)\big[\widehat{T}^\star - \tau_i(t)\big]_+ + Z_R(t)\big[R_i^{\min}(t) - \widehat{R}^\star\big]_+$
4 $\quad$Store decision: local execution
5 $\quad S_1 \leftarrow +\infty,\ S_2 \leftarrow +\infty$
6 $\quad$**foreach** $k \in \mathcal{F}$ **do**
7 $\quad\quad$Predict $\widehat{\gamma}_{i,k}(t)$ using Eq. (10)
8 $\quad\quad S_{i,k}^{\min} \leftarrow +\infty$
9 $\quad\quad$**foreach** $m \in \mathcal{M}$ **do**
10 $\quad\quad\quad$Compute predicted $\widehat{T}_{i,k}^{\mathrm{fog}}(t)$, $\widehat{E}_{i,k}^{\mathrm{fog}}(t)$, and $\widehat{R}_{i,k}^{\mathrm{T}}(t)$ using Eq. (4), Eq. (7), Eq. (6)
11 $\quad\quad\quad s \leftarrow V\Big(\widehat{T}_{i,k}^{\mathrm{fog}}(t) + \omega\widehat{E}_{i,k}^{\mathrm{fog}}(t)\Big) + Q_k(t)\,C_i(t)$
12 $\quad\quad\quad s \leftarrow s + Z_\tau(t)\big[\widehat{T}_{i,k}^{\mathrm{fog}}(t) - \tau_i(t)\big]_+ + Z_R(t)\big[R_i^{\min}(t) - \widehat{R}_{i,k}^{\mathrm{T}}(t)\big]_+$
13 $\quad\quad\quad$**if** $s < S_{i,k}^{\min}$ **then**
14 $\quad\quad\quad\quad S_{i,k}^{\min} \leftarrow s$, Store $(m_{i,k}^\star, \widehat{T}_{i,k}^\star, \widehat{E}_{i,k}^\star, \widehat{R}_{i,k}^\star)$
15 $\quad\quad$**if** $S_{i,k}^{\min} < S_1$ **then**
16 $\quad\quad\quad(S_2, k_2, m_2, \widehat{T}_2, \widehat{E}_2, \widehat{R}_2) \leftarrow (S_1, k_1, m_1, \widehat{T}_1, \widehat{E}_1, \widehat{R}_1)$
17 $\quad\quad\quad(S_1, k_1, m_1, \widehat{T}_1, \widehat{E}_1, \widehat{R}_1) \leftarrow (S_{i,k}^{\min}, k, m_{i,k}^\star, \widehat{T}_{i,k}^\star, \widehat{E}_{i,k}^\star, \widehat{R}_{i,k}^\star)$
18 $\quad\quad$**else if** $S_{i,k}^{\min} < S_2$ **then**
19 $\quad\quad\quad(S_2, k_2, m_2, \widehat{T}_2, \widehat{E}_2, \widehat{R}_2) \leftarrow (S_{i,k}^{\min}, k, m_{i,k}^\star, \widehat{T}_{i,k}^\star, \widehat{E}_{i,k}^\star, \widehat{R}_{i,k}^\star)$
20 $\quad$**if** $S_1 < S^\star$ **then**
21 $\quad\quad S^\star \leftarrow S_1$ and Store decision: execution on (k_1, m_1);
22 $\quad$**if** $S_2 < +\infty$ **then**
23 $\quad\quad\widehat{T}^{(2)} \leftarrow \min\{\widehat{T}_1, \widehat{T}_2\}$, $\widehat{E}^{(2)} \leftarrow \widehat{E}_1 + \widehat{E}_2$
24 $\quad\quad\widehat{R}^{(2)} \leftarrow 1 - (1 - \widehat{R}_1)(1 - \widehat{R}_2)$
25 $\quad\quad S^{(2)} \leftarrow V(\widehat{T}^{(2)} + \omega\widehat{E}^{(2)}) + Q_{k_1}(t)C_i(t) + Q_{k_2}(t)C_i(t)$
26 $\quad\quad S^{(2)} \leftarrow S^{(2)} + Z_\tau(t)\big[\widehat{T}^{(2)} - \tau_i(t)\big]_+ + Z_R(t)\big[R_i^{\min}(t) - \widehat{R}^{(2)}\big]_+$
27 $\quad\quad$**if** $S^{(2)} < S^\star$ **then**
28 $\quad\quad\quad S^\star \leftarrow S^{(2)}$, Store decision: $\mathrm{rep}(k_1, m_1, k_2, m_2)$
29 $\quad$Apply the stored decision for task i
30 Update the physical fog queues using Eq. (5)
31 Update the virtual queues using Eq. (11) and Eq. (12)

IV. Evaluation and Results

The proposed framework was implemented in Python using a slot-based simulator. Mobile IoT devices follow a Gauss-Markov mobility model and generate tasks according to Bernoulli arrivals, determined by the slot duration and workload setting. The main simulation parameters are shown in the Table I. We compared the proposed scheme with three baselines. Those are *NoRep*, which disables selective replication, and *NoPred*, which uses the most recently observed SINR as a last-value predictor instead of the MLP predictor. We also compare with the reliability-aware replication scheme of Yang *et al.* [5]. The MLP predictor is trained on simulation generated SINR traces, where each input consists of the previous $L = 5$ SINR samples. Additionally, the IoT device-to-fog distance, device speed, and the prediction target are generated slot-wise. For fair comparison, all baselines use the same task, mobility, channel, PHY modes, and system settings, with tunable parameters selected from the same candidate ranges.

Results and Discussion: Fig. 2(a) shows the average task delay versus workload size. As the number of tasks increases, delays across all schemes increase due to higher communication contention and a longer fog queue. The proposed scheme consistently achieves the lowest delay because SINR prediction enables more accurate link-aware offloading, while selective replication helps avoid poor execution paths under congestion and unreliable links. On average, the proposed method reduces delay by about 16.4%, 25.6%, and 37.8% compared with the Yang et al., NoRep, and NoPred schemes, respectively. Fig. 2(b) shows the reliability violation rate as the workload increases. For this evaluation, the Gauss–Markov mobility model is enabled with a fixed average device speed of 3 m/s. The violation rate increases for all schemes because a heavier load increases both wireless contention and fog congestion. However, the proposed method consistently achieves the lowest violation rate. Averaged over all workload points, the proposed method lowers the reliability violation rate by about 40.9% when compared with the Yang et al.



(a) Avg. Delay vs. #Tasks (b) Rel. Viol. vs #Tasks (c) Rel. Viol. vs Mob. Speed (d) Energy vs #Tasks (e) Replication vs Reliability

Fig. 2: Simulation Results of Proposed work

TABLE I: Key Simulation Parameters

Parameter	Value
Simulation area	300×300 m^2
No. of IoT devices; No. of fog nodes	30–150; 5–25
Slot duration Δ; Bandwidth W	0.05 s; 5–10 MHz
Pathloss exponent κ; Noise power σ^2	3; 10^{-10} W
Max. retransmissions N_{tx}; SINR history (L)	3; 5
PHY blocklengths n	$\{128, 256, 512\}$
Target decoding error rates	$\{10^{-2}, 10^{-3}, 10^{-4}\}$
MLP hidden layers; LDPP control par.(V)	(32, 16); 15
MLP data split train/test/validation	70:15:15
No. of independent runs	10

scheme, 57.9% when compared with NoRep, and 59.6% when compared with NoPred. Fig. 2(c) shows the results of reliability violation versus mobility speed, with the workload size fixed at 90 tasks. The results show that as mobility increases, channel conditions fluctuate more rapidly, making reliable task delivery more difficult for all schemes. Therefore, the proposed method reduces the reliability violation rate by about 20.0%, 39.7%, and 53.6% when compared with the Yang et al. scheme, NoRep, and NoPred, respectively. Fig. 2(d) shows the average energy consumption versus workload size. As expected, the energy consumption increases with the increase in number of tasks for all schemes. The no-replication scheme achieves the lowest energy because it avoids redundancy, but this comes at the cost of weaker reliability performance. In particular, the proposed scheme reduces energy consumption by about 12.6% compared to the Yang et al. scheme and 24.6% compared to NoPred. Fig. 2(e) illustrates the replication ratio under different minimum reliability requirements. Here, replication ratio is defined as $\rho_{\text{rep}} = N_{\text{rep}}/N_{\text{tot}}$, where N_{rep} is the number of replicated tasks and N_{tot} is the total number of processed tasks. As target reliability becomes stricter, both replication-enabled schemes increase their reliance on replication. However, the proposed method requires a lower replication ratio than the Yang et al scheme, about 37.0%. It shows that LDPP-based queue-aware control makes replication more selective and resource-efficient.

V. Conclusion

This paper presents a mobility-aware, reliable task offloading framework for fog-enabled IoT networks. The proposed method combined multilayer perceptron (MLP) based SINR prediction with LDPP-based online control. It selects either local execution, single-fog offloading, or selective replicated execution under dynamic wireless and queue conditions. The results showed that the proposed scheme reduces delay and reliability violations while achieving a better delay–reliability–energy tradeoff than the existing schemes. As future work, the proposed framework can be extended to multi-controller fog environments and heterogeneous task classes. It can also be validated on a real-world fog-IoT testbed.

References

[1] R. Dasari and S. K. Nayak, "Pr-fog: An efficient task priority-based reliable provisioning of resources in fog-enabled iot networks," *IEEE Transactions on Network and Service Management*, vol. 23, pp. 2543–2553, 2026.

[2] R. Huang, W. Wen, Z. Zhou, C. Dong, and X. Chen, "Mec-enabled task replication with resource allocation for reliability-sensitive services in 5g mmtc networks," *IEEE Transactions on Services Computing*, vol. 18, no. 1, pp. 253–269, 2024.

[3] J. Wang, Y. Zhu, Y. Hu, M. C. Gursoy, and A. Schmeink, "Average reliability-optimal offloading for mobile edge computing in low-latency industrial iot networks," *IEEE Transactions on Mobile Computing*, vol. 24, no. 7, pp. 5888–5902, 2025.

[4] J. Wang, Y. Hu, Y. Zhu, T. Wang, and A. Schmeink, "Reliability-optimal offloading for mobile edge-computing in low-latency industrial iot networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 10, pp. 12 765–12 781, 2024.

[5] L. Yang, A. Zhou, X. Ma, Y. Zhang, and S. Wang, "Reliability-aware task replication for mobile edge computing," *IEEE Internet of Things Journal*, vol. 11, no. 14, pp. 24 846–24 857, 2024.

[6] I. M. Amer, S. M. Oteafy, and H. S. Hassanein, "Task replication in unreliable edge networks," in *2022 IEEE 47th Conference on Local Computer Networks (LCN)*. IEEE, 2022, pp. 173–180.

[7] H. Hao, C. Xu, W. Zhang, X. Chen, S. Yang, and G.-M. Muntean, "Reliability-aware optimization of task offloading for uav-assisted edge computing," *IEEE Transactions on Computers*, vol. 74, no. 11, pp. 3832–3844, 2025.

[8] A. Kaur and A. Godara, "Machine learning empowered green task offloading for mobile edge computing in 5g networks," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 810–820, 2023.

[9] K. Ergun, R. Ayoub, P. Mercati, and T. Š. Rosing, "Dynamic reliability management of multigateway iot edge computing systems," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3864–3889, 2022.

[10] J. Dou, G. Xie, Z. Tian, L. Cui, and S. Yu, "Modeling and analyzing the spatial–temporal propagation of malware in mobile wearable iot networks," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2438–2452, 2024.

[11] F. Mahmood, E. Perrins, and L. Liu, "Energy-efficient wireless communications: From energy modeling to performance evaluation," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7643–7654, 2019.