Freshness-Aware Inference Services in Edge Computing via Offloading or Local Processing

Xuan Ai
Department of Computer Science
City University of Hong Kong, Hong Kong, China
xuanai2-c@my.cityu.edu.hk

Weifa Liang
Department of Computer Science
City University of Hong Kong, Hong Kong, China
weifa.liang@cityu.edu.hk

2025 IEEE 50th Conference on Local Computer Networks (LCN) | 979-8-3315-3703-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/LCN65610.2025.11146061

***Abstract*—The collaboration between end devices and edge servers has been extensively investigated to enable adaptive service provisioning, particularly in scenarios requiring trade-offs between differential model accuracies and heterogeneous resource consumption costs. In this paper, we propose freshness-aware hierarchical inference service provisioning in a Mobile Edge Computing (MEC) network, where inference models are trained and maintained in edge servers to address resource limitations on devices. Devices dynamically download updated models to maintain local inference fidelity, mitigating performance degradation caused by model staleness. We formulate a freshness and cost minimization problem that maximizes overall inference fidelity while minimizing total costs, subject to long-term average energy budgets on devices and computing capacities on cloudlets. We design an online algorithm with provable competitive ratio, by leveraging Lyapunov optimization and randomized rounding techniques. We conduct simulations to evaluate the performance of the proposed online algorithm. Simulation results demonstrate that the proposed algorithm is promising.**

I. Introduction

The rapid proliferation of Internet of Things (IoT)-enabled applications facilitates the adoption of Edge Intelligence (EI), which supports distributed Model-as-a-Service (MaaS) for real-time inference with high fidelity at the network edge [1], [7], [19]. For freshness-aware EI applications, service fidelity exhibits strong temporal dependence on model freshness [7], [10]. Their performance degrades dramatically as model staleness increases, which necessitates periodic model updates, such as the road condition monitoring system [2].

In this paper, we propose freshness-aware hierarchical service provisioning in an MEC network, which consists of two tiers: a lower tier of energy-constrained end devices capable of local inference using deployed pre-trained models with low fidelity; a higher tier of edge servers (cloudlets) that deliver inference services and maintain continuously retrained high-fidelity models. Fig. 1 is an illustrative example of hierarchical service provisioning in an MEC network, which allows for flexible request processing through the following two complementary mechanisms. 1) A portion of inference requests is processed by end devices locally, which delivers inference results with lower fidelity, but prioritizes low-latency responses for delay-sensitive tasks at reduced costs. 2) Inference requests can be alternatively offloaded to nearby edge servers for execution, where they benefit from freshly retrained models with high accuracy, at the expense of higher latency and costs [3].

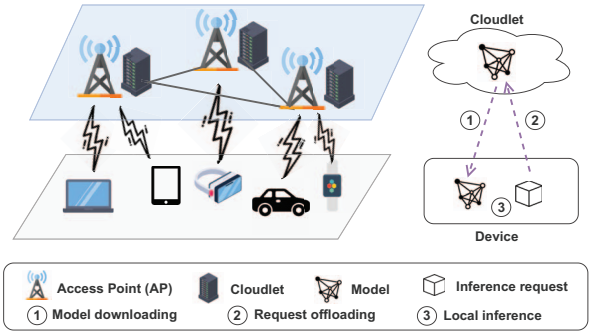


Fig. 1: An illustrative example of freshness-aware hierarchical service provisioning in an MEC network.

However, with limited computing and energy capacities, it is impractical for end devices to perform computationally intensive model retraining to address issues such as model staleness. To overcome this limitation, model retraining is conducted in cloudlets, and devices then decide whether to download these updated service models for local inference. Although energy consumption is non-negligible during the downloading process due to the large size of service models [17], it ensures inference benefits from updated models with superior fidelity and enhances overall service quality.

Although the collaboration between energy-constrained end devices and edge servers balances trades-off between service costs, resource efficiency, and service fidelity, by leveraging their strengths to optimize system performance, it poses several challenges. First, how to measure model freshness for inference fidelity? Second, when should energy-constrained devices download the updated service models to enhance local inference, without information of future request arrivals? Third, when and where should devices offload requests to balance a trade-off between inference fidelity and costs, subject to the energy budgets of devices and computing capacities on cloudlets?

There are several studies devoted to hierarchical inference systems with the collaboration between devices and MECs [3], [15]. For example, Moothedath *et al.* [15] considered an application scenario where an end device is equipped with a small model with low accuracy, while a large model with high accuracy is available at an edge server. They proposed a threshold offloading policy based on the confidence of the

small model. Beytur *et al.* [3] further considered a hierarchical inference system where multiple clients connect to a server. Clients can make offloading decisions based on local inference results to minimize inference errors, subject to the resource allocation among them. There are a few studies that considered the model updating on end devices for local service quality improvement [7], [14]. For instance, Han *et al.* [7] assumed that clients can update local models by downloading the updated models from the server. They proposed a novel metric, Age of Model (AoM), to measure the service quality on clients. Luo *et al.* [14] considered a network where end devices simultaneously perform federated learning (FL) model training to update local models. An online algorithm is proposed to control model training for a trade-off between inference performance and resource costs. Different from the aforementioned investigations, in this paper, we will study freshness-aware hierarchical inference service provisioning in an MEC network, by considering both the service levels with differential accuracies and heterogeneous costs, as well as local model updates on devices.

The main contributions of this paper are presented as follows. We study a freshness-aware hierarchical service provisioning in an MEC network, which not only enhances costs and resource efficiency but also ensures adaptability and service fidelity through introducing the Age of Information (AoI) metric. We propose a novel optimization problem for inference request admissions, with the aim to maximize overall inference fidelity while minimizing total cost. We devise a performance-guaranteed online algorithm for the problem, by leveraging the Lyapunov optimization and randomized rounding techniques. We finally conduct simulations to evaluate the performance of the proposed online algorithm.

II. Prelimnaries

A. System model

We consider an MEC network represented by an undirected graph $G = (N, E)$, where N is a set of access points (APs), and E is a set of links connecting the APs. Each AP connects to its co-located cloudlet through a high-speed optical cable, and the communication delay between them is negligible. Thus, denote by $j \in N$ an AP or its co-located cloudlet for simplicity if no confusion arises. Let B_j be the bandwidth of AP j, and Cap_j be the computing capacity on the co-located cloudlet of AP j. For each link $e \in E$, denote by μ_e the transmission cost per unit data along link e.

There is a set $\mathcal{M}$ of service models (e.g., DNN models) stored in cloudlets for inference service provisioning, and they are frequently updated through continual retraining to maintain service fidelity, which is closely related to model freshness, as defined in detail later. Let $comp_m$ be the amount of computing resource demanded by an instance of each model $M_m \in \mathcal{M}$ for request execution in cloudlets. Denote by S_m the parameter size of each model $M_m \in \mathcal{M}$. There is a set $\mathcal{V}$ of devices moving across different geographic locations in the network, which continuously issues inference requests. For each device $v_i \in \mathcal{V}$, let $\overline{\mathcal{E}}_i$ be its average energy budget per time slot, and denote by $\mathcal{M}_i \subset \mathcal{M}$ a set of service models deployed on it. Assume that each request issued by device v_i requests service from a model in $\mathcal{M}_i$. We assume that the network G runs over a time horizon $\mathbb{T}$, and $\mathbb{T}$ is discretized into $|\mathbb{T}|$ equal time slots with $\mathbb{T} = \{1, \ldots, |\mathbb{T}|\}$.

B. Model freshness

Considering the freshness-aware nature of service models, service fidelity critically depends on the freshness of service models. Inspired by the Age of Model (AoM) introduced by [7], model freshness is proposed as a novel metric to measure service fidelity based on the time elapsed since the last model retraining. A small value of model freshness implies that the model is fresh and can achieve high inference accuracy. Otherwise, a large value of model freshness indicates that the model is out of date, leading to reduced inference accuracy. For the freshness-aware hierarchical service provisioning, there always exists a difference between the performance of the service models deployed on cloudlets and devices, as it is challenging for devices to consistently maintain synchronized updates through model downloading. In the following, we quantify the model freshness of service models deployed on cloudlets and devices, respectively.

Model freshness on cloudlets: Service models stored in cloudlets are periodically updated through continual retraining. For each model $M_m \in \mathcal{M}$, let $F^{cloud}_{m,t}$ be its freshness at time slot t, which is defined as $F^{cloud}_{m,t} = t - t'$, where t' is the time slot of its last retraining with $t' \leq t$. That is, model freshness grows linearly with time if no retraining occurs, but drops to 0 immediately after retraining.

Model freshness on devices: As the retaining of service models consumes large amounts of computing resource, it is unrealistic that model retraining can be supported by devices, while each device v_i can determine whether to upgrade its local models at the beginning of each time slot t through downloading the updated models from cloudlets.

Similar to the definition of $F^{cloud}_{m,t}$, for each model $M_m \in \mathcal{M}_i$, let $F^{loc}_{i,m,t}$ be its freshness on device v_i at time slot t. Then,

$$F^{loc}_{i,m,t} = \begin{cases} F^{cloud}_{m,t}, \text{ if } v_i \text{ downloads } M_m \text{ at } t \\ F^{loc}_{i,m,t-1} + 1, \text{ otherwise.} \end{cases} \tag{1}$$

That is, the model freshness on each device v_i depends on both its downloading decisions and the model freshness on cloudlets. If device v_i decides to download the updated version of model M_m at time slot t, the model version on device v_i synchronizes with that on cloudlets, and its freshness $F^{loc}_{i,m,t}$ decreases to $F^{cloud}_{m,t}$. Otherwise, $F^{loc}_{i,m,t}$ increases by 1 than $F^{loc}_{i,m,t-1}$ since the local model M_m will be older.

C. Local model update

The energy consumption and costs of model downloading by each device $v_i \in \mathcal{V}$ are quantified as follows.

Given time slot t, assume that the nearest AP of device v_i is indexed by l_i, and denote by $R^{down}_{i,t}$ the downlink rate of device v_i when downloading models from AP l_i.

The energy consumption of device v_i for downloading the updated version of model $M_m \in \mathcal{M}_i$ at time slot t is

$$\epsilon_{i,m,t}^{down} = \frac{S_m}{R_{i,t}^{down}} \cdot p_i^{down}, \tag{2}$$

where S_m is the parameter size of model M_m.

The download cost of device v_i for downloading M_m is

$$cost_{i,m,t}^{down} = \epsilon_{i,m,t}^{down} \cdot \zeta_i, \tag{3}$$

where ζ_i is the cost of device v_i per unit energy consumption.

D. Local inference vs. request offloading

Inference requests are stochastically issued by devices during the time horizon $\mathbb{T}$. Given time slot t, denote by $\mathcal{V}_t$ the set of devices that issues inference requests. For each device $v_i \in \mathcal{V}_t$, let $vol_{i,t}$ be the volume of its request, and m_i^t be the index of its requested model with $M_{m_i^t} \in \mathcal{M}_i$. The inference request issued by device v_i will be either processed locally or offloaded to a cloudlet for execution. For the former, it will be served by model $M_{m_i^t}$ with freshness $F_{i,m_i^t,t}^{loc}$. For the latter, it will be served by the latest version of $M_{m_i^t}$, of which the freshness is $F_{m_i^t,t}^{cloud}$. The energy consumption and costs for the two cases are as follows.

Local inference: If the request issued by device v_i at time slot t is processed locally on v_i, the incurred cost is attributed to the energy consumption of v_i during the processing.

The energy consumption of device v_i for processing the request locally is

$$\epsilon_{i,t}^{loc} = \xi \cdot vol_{i,t} \cdot f_i^2, \tag{4}$$

where ξ is a coefficient, $vol_{i,t}$ is the volume of the request, and f_i is the CPU frequency of device v_i.

The local processing cost attributed to the energy consumption of device v_i is

$$cost_{i,t}^{(0)} = \epsilon_{i,t}^{loc} \cdot \zeta_i, \tag{5}$$

where ζ_i is the cost of device v_i per unit energy consumption.

Request offloading: If the request is offloaded to cloudlet j, the cost consists of three parts: the energy consumption cost of v_i to upload the request, the transmission cost for transferring the request to cloudlet j, and the computing cost.

The uploading rate of device v_i to upload the request to its nearest AP j_i^t at time slot t is

$$R_{i,t}^{up} = B_{j_i^t} \log_2(1 + \frac{p_i \cdot h_{i,j_i^t}^{up}}{\sigma^2}), \tag{6}$$

where p_i is the transmission power of device v_i, $h_{i,j_i^t}^{up}$ is the channel gain, and σ is the Gaussian white noise.

The energy consumption of device v_i to upload the input data of the request at time slot t is

$$\epsilon_{i,t}^{up} = \frac{vol_{i,t}}{R_{i,t}^{up}} \cdot p_i. \tag{7}$$

The cost of offloading request and transferring it to cloudlet j for execution by device v_i at time slot t is

$$cost_{i,t}^{(j)} = \epsilon_{i,t}^{up} \cdot \zeta_i + \sum_{e \in P_{j,j_i^t}} vol_{i,t} \cdot \mu_e + comp_{m_i^t} \cdot \eta, \tag{8}$$

where the first term in the RHS of Eq. (8) is the uploading cost of device v_i, the second term is the transmission cost of transferring request from its uploading location to cloudlet j, where P_{j,j_i^t} is a shortest path in G between cloudlets j and j_i^t, and μ_e is the transmission cost per unit data along link e, and the third term is the request processing cost against model $M_{m_i^t}$ in cloudlet j, where $comp_{m_i^t}$ is the amount of computing resource demanded by an instance of model $M_{m_i^t}$ to process a request, and η is the cost per unit computing resource.

E. Problem definition

Definition 1: Given an MEC network represented by $G = (N, E)$, there is a set $\mathcal{M}$ of service models updated frequently through continual training and a set $\mathcal{V}$ of devices. Each device $v_i \in \mathcal{V}$ has an average energy budget $\overline{\mathcal{E}}_i$ and accommodates a set $\mathcal{M}_i$ of inference models. Let $\mathcal{V}_t$ be the set of devices that issues requests at time slot t. *The freshness and cost minimization problem* in G is to minimize the weighted sum of the long-term freshness of inference results and resource consumption costs over time horizon $\mathbb{T}$, by dynamically scheduling model updates in devices and request offloading from devices, subject to computing capacities on cloudlets and energy budgets on devices.

We define the following binary variables for later problem formulation. Let $x_{i,t}^{(j)}$ be a binary variable. When $1 \le j \le |N|$, $x_{i,t}^{(j)} = 1$ indicates that the request issued by device $v_i \in \mathcal{V}_t$ is assigned to cloudlet j for processing, and $x_{i,t}^{(j)} = 0$ otherwise. When $j = 0$, $x_{i,t}^{(0)} = 1$ implies that the request is processed locally on device v_i, and $x_{i,t}^{(0)} = 0$ otherwise. Obviously, for each device $v_i \in \mathcal{V} \backslash \mathcal{V}_t$, $x_{i,t}^{(j)} = 0$ for $0 \le j \le |N|$. Let $y_{i,m,t}$ be a binary variable, where $y_{i,m,t} = 1$ indicates that device v_i downloads the updated model $M_m \in \mathcal{M}_i$ at time slot t, and $y_{i,m,t} = 0$ otherwise.

Freshness of inference results: Given time slot t, the freshness $F_{i,m,t}^{loc}$ of model $M_m \in \mathcal{M}_i$ deployed on device v_i, which is defined by Eq. (1), can be presented using binary variable $y_{i,m,t}$, i.e.,

$$F_{i,m,t}^{loc} = F_{m,t}^{cloud} \cdot y_{i,m,t} + (F_{i,m,t-1}^{loc} + 1) \cdot (1 - y_{i,m,t}). \tag{9}$$

As a request will either be processed locally on device or be offloaded to a cloudlet for execution, the total freshness F_t of inference results at time slot t is defined as follows.

$$\begin{aligned} F_t &= \sum_{v_i \in \mathcal{V}} \sum_{j=1}^{|N|} F_{m_i^t,t}^{cloud} \cdot x_{i,t}^{(j)} + F_{i,m_i^t,t}^{loc} \cdot x_{i,t}^{(0)} \\ &= \sum_{v_i \in \mathcal{V}} \sum_{j=1}^{|N|} F_{m_i^t,t}^{cloud} \cdot x_{i,t}^{(j)} + (F_{i,m_i^t,t-1}^{loc} + 1) \cdot x_{i,t}^{(0)} \\ &\quad - (F_{i,m_i^t,t-1}^{loc} + 1 - F_{m_i^t,t}^{cloud}) \cdot x_{i,t}^{(0)} \cdot y_{i,m_i^t,t}, \end{aligned} \tag{10}$$

Energy consumption of each device: Let $\mathcal{E}_{i,t}$ be the energy consumption of each device v_i at time slot t. Then,

$$\mathcal{E}_{i,t} = \sum_{M_m \in \mathcal{M}} \epsilon^{down}_{i,m,t} \cdot y_{i,m,t} + \sum_{j=1}^{|N|} \epsilon^{up}_{i,t} \cdot x^{(j)}_{i,t} + \epsilon^{loc}_{i,t} \cdot x^{(0)}_{i,t}. \quad (11)$$

The expected energy consumption of each device $v_i \in \mathcal{V}$ per time slot must meet

$$\lim_{|\mathbb{T}|\to\infty} \frac{1}{|\mathbb{T}|} \sum_{t=1}^{|\mathbb{T}|} \mathcal{E}_{i,t} \leq \overline{\mathcal{E}}_i. \quad (12)$$

Total cost: Denote by $cost_t$ the cost in the network at time slot t, we have

$$cost_t = \sum_{v_i \in \mathcal{V}} \sum_{M_m \in \mathcal{M}} cost^{down}_{i,m,t} \cdot y_{i,m,t} + \sum_{v_i \in \mathcal{V}} \sum_{j=0}^{|N|} cost^{(j)}_{i,t} \cdot x^{(j)}_{i,t}. \quad (13)$$

In the RHS of Eq. (13), the first term is the cost of local model updates, and the second term is the cost of request processing, either on devices or on cloudlets.

Objective function: The objective of the freshness and cost minimization problem is the long-term average of the weighted sum of inference freshness and costs. Denote by W_t the weighted sum of inference freshness and costs at each time slot t, we have

$$W_t = \alpha \cdot F_t + (1-\alpha) \cdot cost_t, \quad (14)$$

where α is a positive coefficient that balances the weights between the freshness and the cost with $0 \leq \alpha \leq 1$.

F. ILP formulation

Due to the non-linearity of F_t by Eq. (10), we formulate the freshness and cost minimization problem as an Integer Nonlinear Programming (INP) as follows.

$$\textbf{P1:} \quad \underset{x^{(j)}_{i,t}, y_{i,m,t}}{\text{Minimize}} \lim_{|\mathbb{T}|\to\infty} \frac{1}{|\mathbb{T}|} \sum_{t=1}^{|\mathbb{T}|} W_t \quad (15)$$

subject to: from (1) to (14),

$$\lim_{|\mathbb{T}|\to\infty} \frac{1}{|\mathbb{T}|} \sum_{t=1}^{|\mathbb{T}|} \mathcal{E}_{i,t} \leq \overline{\mathcal{E}}_i, \forall v_i \in \mathcal{V} \quad (16)$$

$$\sum_{v_i \in \mathcal{V}_t} comp_{m^t_i} \cdot x^{(j)}_{i,t} \leq Cap_j, \forall j \in N, 1 \leq t \leq |\mathbb{T}| \quad (17)$$

$$\sum_{j=0}^{|N|} x^{(j)}_{i,t} = 1, \forall v_i \in \mathcal{V}_t, 1 \leq t \leq |\mathbb{T}| \quad (18)$$

$$x^{(j)}_{i,t} \in \{0,1\}, 0 \leq j \leq |N|, \forall v_i \in \mathcal{V}, 1 \leq t \leq |\mathbb{T}| \quad (19)$$

$$y_{i,m,t} \in \{0,1\}, \forall v_i \in \mathcal{V}, \forall M_m \in \mathcal{M}, 1 \leq t \leq |\mathbb{T}| \quad (20)$$

$$x^{(j)}_{i,t} = 0, 0 \leq j \leq |N|, \forall v_i \in \mathcal{V} \backslash \mathcal{V}_t, 1 \leq t \leq |\mathbb{T}| \quad (21)$$

$$y_{i,m,t} = 0, \forall v_i \in \mathcal{V}, \forall M_m \in \mathcal{M} \backslash \mathcal{M}_i, 1 \leq t \leq |\mathbb{T}| \quad (22)$$

where Objective (15) is the long-term average of the weighted sum of inference freshness and costs. Constraint (16) ensures energy constraints on devices. Constraint (17) ensures the capacity constraints on cloudlets. Constraint (18) guarantees that the inference request of each device $v_i \in \mathcal{V}_t$ will either be processed locally on device v_i or be offloaded to a cloudlet.

Theorem 1: The freshness and cost minimization problem in an MEC network $G = (N, E)$ is NP-hard.

Proof Due to space limitation, the proof is omitted.

III. Online algorithm for the freshness and cost minimization problem

A. Lyapunov optimization formulation

Virtual queues for long-term energy consumption: Due to the dynamic property of the system, a primal challenge of **P1** is how to maintain energy efficiency on each device without the knowledge of future request arrivals overtime. The key idea of Lyapunov optimization is to construct virtual queues for long-term energy consumption on each energy-constrained device and to optimize the problem while keeping the energy consumption queues stable. For each device $v_i \in \mathcal{V}$, a virtual queue as a historical measure of energy consumption exceeding its average energy budget is defined as follows.

$$Q_i(t+1) = \max(Q_i(t) + \mathcal{E}_{i,t} - \overline{\mathcal{E}}_i, 0), \quad (23)$$

where $Q_i(t)$ is the virtual queue length at time slot t, which is the exceeded energy consumption on device v_i by the end of time slot t, and the initial queue backlog is assumed to be 0 (i.e., $Q_i(0) = 0$). A large $Q_i(t)$ implies that the average of cumulative energy consumption of device v_i has significantly exceeded its long-term average budget $\overline{\mathcal{E}}_i$, while a small $Q_i(t)$ suggests compliance with the energy budget constraint.

Virtual queue stability: To ensure the stability of the virtual queues, we define a *quadratic Lyapunov function* and a *Lyapunov drift function*, respectively, as follows. The defined quadratic Lyapunov function is

$$L(t) = \frac{1}{2} \sum_{v_i \in \mathcal{V}} Q_i(t)^2. \quad (24)$$

A policy guarantees the stability of virtual queues if it consistently pushes the quadratic Lyapunov function toward a bounded level. Hence, we introduce the following one-step Lyapunov drift function.

$$\Delta L(t) = L(t+1) - L(t), \quad (25)$$

which describes the evolution of virtual queues in the quadratic Lyapunov function over one time slot.

Joint Lyapunov drift and freshness and cost minimization: By the construction of virtual energy consumption queues, the original problem can be decomposed into a series of one-time-slot optimization problems. A policy can be proposed by achieving a trade-off between the virtual queue stability and the optimization objective of minimizing freshness and cost. A *Lyapunov drift-plus-penalty function* for the one-time-slot optimization problem is defined as

$$\Delta L(t) + V \cdot W_t, \quad (26)$$

Algorithm 1 Online algorithm for the freshness and cost minimization problem

Input: Given an MEC network $G = (N, E)$, a set $\mathcal{M}$ of service models, a set $\mathcal{V}$ of devices that either offloads requests to MEC or processes requests locally, and a subset $\mathcal{V}_t$ of devices that issues requests at time slot t.

Output: Each device v_i makes decisions for model downloading and request offloading, with the objective to maximize overall inference fidelity while minimizing total costs.

1: **for** each device $v_i \in \mathcal{V}$ **do**
2: $Q_i(0) \leftarrow 0$ /* initialize a virtual queue Q_i */
3: **end for**;
4: $t \leftarrow 1$;
5: **while** $t \leq |\mathbb{T}|$ **do**
6: A sequence of arrived requests issued by $\mathcal{V}_t$;
7: Formulate a Lyapunov drift-plus-penalty function by (26);
8: Find an approximate solution $\mathcal{S}_t$ that minimizes the upper bound of the objective function in (29), by invoking the randomized algorithm `Algorithm 2`;
9: **for** each device $v_i \in \mathcal{V}$ **do**
10: $Q_i(t+1) \leftarrow \max(Q_i(t) + \mathcal{E}_{i,t} - \overline{\mathcal{E}}_i, 0)$ /*update the virtual queue $Q_i(t)$*/
11: **end for**;
12: $\mathcal{S} \leftarrow \mathcal{S} \cup \langle \mathcal{S}_t \rangle$;
13: $t \leftarrow t + 1$;
14: **end while**;
15: **return** A feasible solution $\mathcal{S}$ for the freshness and cost minimization problem with high probability.

where V is a non-negative control parameter that adjusts the trade-off between the energy consumption queue backlogs and the weighted sum of freshness and costs.

Lemma 1: For all possible solutions delivered by any feasible policy over all time slots in $\mathbb{T}$, we have

$$\Delta L(t) + V \cdot W_t \leq B + \sum_{v_i \in \mathcal{V}} Q_i(t) \cdot (\mathcal{E}_{i,t} - \overline{\mathcal{E}}_i) + V \cdot W_t, \tag{27}$$

where $B = \frac{1}{2} \sum_{v_i \in \mathcal{V}} (\mathcal{E}_i^{max})^2 + \overline{\mathcal{E}}_i^2$ is a constant for all time slots, $\mathcal{E}_i^{max} = \max\{\mathcal{E}_{i,t} \mid \forall t \in \mathbb{T}\}$, and W_t is defined in (14).

Proof The detailed proof is shown in the APPENDIX.

B. Online algorithm

Having transformed the original optimization problem into the Lyapunov optimization problem, we now decompose **P1** in (15) to a series of one-time-slot drift-plus-penalty upper bound minimization problems. By Lemma 1, we can approximate the upper bound closely by minimizing the RHS of Ineq. (27) at each time slot $t \in \mathbb{T}$. However, it can be shown that this minimization problem is still NP-hard. We instead develop a randomized algorithm, `Algorithm 2`, for it later. The online algorithm for the freshness and cost minimization problem is detailed in `Algorithm 1`.

C. Randomized algorithm

To minimize the upper bound of the Lyapunov drift-plus-penalty function at each time slot t, we only consider the remaining terms in the RHS of Ineq. (27). We have

$$\sum_{v_i \in \mathcal{V}} Q_i(t) \cdot (\mathcal{E}_{i,t} - \overline{\mathcal{E}}_i) + V \cdot W_t \leq \sum_{v_i \in \mathcal{V}} Q_i(t) \mathcal{E}_{i,t} + V \cdot W_t. \tag{28}$$

Hence, the objective of the one-time-slot optimization problem is to minimize $\sum_{v_i \in \mathcal{V}} Q_i(t) \mathcal{E}_{i,t} + V \cdot W_t$ at time slot t, which can be rewritten using the aforementioned parameter definitions by Eq. (10) – (14) as follows.

$$\begin{aligned} \text{Minimize} \quad & \sum_{v_i \in \mathcal{V}} Q_i(t) \mathcal{E}_{i,t} + V \cdot W_t \\ & = \sum_{v_i \in \mathcal{V}} \sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot x_{i,t}^{(j)} + \sum_{v_i \in \mathcal{V}} \sum_{M_m \in \mathcal{M}} \beta_{i,m,t} \cdot y_{i,m,t} \\ & \quad - \sum_{v_i \in \mathcal{V}} \gamma_{i,t} \cdot x_{i,t}^{(0)} \cdot y_{i,m_i^t,t}, \end{aligned} \tag{29}$$

where $\lambda_{i,t}^{(0)} = Q_i(t)\epsilon_{i,t}^{loc} + V \cdot (\alpha \cdot (F_{i,m_i^t,t-1}^{loc} + 1) + (1-\alpha) \cdot cost_{i,t}^{(0)})$, $\lambda_{i,t}^{(j)} = Q_i(t)\epsilon_{i,t}^{up} + V \cdot (\alpha \cdot F_{m_i^t,t}^{cloud} + (1-\alpha) \cdot cost_{i,t}^{(j)})$ with $1 \leq j \leq |N|$, $\beta_{i,m,t} = Q_i(t) \cdot \epsilon_{i,m,t}^{down} + V \cdot (1-\alpha) \cdot cost_{i,m,t}^{down}$, and $\gamma_{i,t} = V \cdot \alpha \cdot (F_{i,m_i^t,t-1}^{loc} + 1 - F_{m_i^t,t}^{cloud})$. These coefficients are all constants at time slot t, as $Q_i(t) = \max(Q_i(t-1) + \mathcal{E}_{i,t-1} - \overline{\mathcal{E}}_i, 0)$ by Eq. (23), of which the value is determined by historical information from time slots 1 to $t-1$. Solving (29) is equivalent to solving the following INP that minimizes the upper bound of the drift-plus-penalty function at time slot t.

$$\begin{aligned} \textbf{P2:} \quad & \underset{x_{i,t}^{(j)}, y_{i,m,t}}{\text{Minimize}} \sum_{v_i \in \mathcal{V}} \sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot x_{i,t}^{(j)} \\ & + \sum_{v_i \in \mathcal{V}} \sum_{M_m \in \mathcal{M}} \beta_{i,m,t} \cdot y_{i,m,t} - \sum_{v_i \in \mathcal{V}} \gamma_{i,t} \cdot x_{i,t}^{(0)} \cdot y_{i,m_i^t,t} \end{aligned} \tag{30}$$

subject to: from (1) to (14),

$$\sum_{v_i \in \mathcal{V}_t} comp_{m_i^t} \cdot x_{i,t}^{(j)} \leq Cap_j, \forall j \in N \tag{31}$$

$$\sum_{j=0}^{|N|} x_{i,t}^{(j)} = 1, \forall v_i \in \mathcal{V}_t \tag{32}$$

$$y_{i,m,t} \in \{0,1\}, \forall v_i \in \mathcal{V}, \forall M_m \in \mathcal{M} \tag{33}$$

$$y_{i,m,t} = 0, \forall v_i \in \mathcal{V}, \forall M_m \in \mathcal{M} \backslash \mathcal{M}_i \tag{34}$$

$$x_{i,t}^{(j)} \in \{0,1\}, 0 \leq j \leq |N|, \forall v_i \in \mathcal{V} \tag{35}$$

$$x_{i,t}^{(j)} = 0, 0 \leq j \leq |N|, \forall v_i \in \mathcal{V} \backslash \mathcal{V}_t \tag{36}$$

The reason why the average energy budget constraints on devices can be negligible is as follows. Due to $Q_i(t+1) - Q_i(t) \geq \mathcal{E}_{i,t} - \overline{\mathcal{E}}_i$ from Eq.(23), by adding up the inequality from time slots 0 to $|\mathbb{T}| - 1$, we have

$$\frac{Q_i(|\mathbb{T}|) - Q_i(0)}{|\mathbb{T}|} + \overline{\mathcal{E}}_i \geq \frac{1}{|\mathbb{T}|} \sum_{t=0}^{|\mathbb{T}|-1} \mathcal{E}_{i,t}. \tag{37}$$

As $Q_i(0) = 0$, we have

$$\lim_{|\mathbb{T}|\to\infty} \frac{\mathbb{E}[Q_i(|\mathbb{T}|)]}{|\mathbb{T}|} + \overline{\mathcal{E}}_i \geq \lim_{|\mathbb{T}|\to\infty} \frac{1}{|\mathbb{T}|} \sum_{t=0}^{|\mathbb{T}|-1} \mathbb{E}[\mathcal{E}_{i,t}]. \quad (38)$$

Hence, if virtual queue $Q_i(t)$ is stable, i.e., $\lim_{|\mathbb{T}|\to\infty} \frac{\mathbb{E}[Q_i(|\mathbb{T}|)]}{|\mathbb{T}|} = 0$, the long-term average energy budget constraint (16) can be guaranteed. As a result, we consider the stability of virtual queues in the following, instead of considering the average energy budget constraints on devices. Since **P2** is NP-hard, it is challenging to find an optimal solution to it in polynomial time. Instead, we here devise a randomized algorithm for it.

Specifically, we first reformulate the INP **P2** to an equivalent ILP **P3**. Then, we propose a randomized algorithm for solving the linear relaxation LP of the ILP, which in turn returns a 2-approximate solution to the ILP with high probability, at the expense of moderate resource violations. Notice that the non-linearity of the INP **P2** can be removed by introducing a new binary variable $z_{i,t}$ to replace $x_{i,t}^{(0)} \cdot y_{i,m_i^t,t}$. That is, $z_{i,t} = 1$ indicates that device $v_i \in \mathcal{V}_t$ downloads the latest version of its requested model $M_{m_i^t}$, and the request is processed in device v_i locally. Otherwise, $z_{i,t} = 0$. Hence, **P2** can be reformulated as the following Integer Linear Programming (ILP).

$$\textbf{P3:} \quad \text{Minimize} \sum_{v_i\in\mathcal{V}} \sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot x_{i,t}^{(j)} + \sum_{v_i\in\mathcal{V}} \sum_{M_m\in\mathcal{M}} \beta_{i,m,t} \cdot y_{i,m,t} - \sum_{v_i\in\mathcal{V}} \gamma_{i,t} \cdot z_{i,t} \quad (39)$$

subject to: from (1) to (14), (31) to (36)

$$z_{i,t} \leq x_{i,t}^{(0)}, \forall v_i \in \mathcal{V} \quad (40)$$

$$z_{i,t} \leq y_{i,m_i^t,t}, \forall v_i \in \mathcal{V} \quad (41)$$

$$z_{i,t} \geq x_{i,t}^{(0)} + y_{i,m_i^t,t} - 1, \forall v_i \in \mathcal{V} \quad (42)$$

$$z_{i,t} \in \{0,1\}, \forall v_i \in \mathcal{V} \quad (43)$$

where Constraints (40) to (43) ensure that if $x_{i,t}^{(0)} = 0$ or $y_{i,m_i^t,t} = 0$, then $z_{i,t} = 0$; otherwise, if both $x_{i,t}^{(0)} = 1$ and $y_{i,m_i^t,t} = 1$, the value of $z_{i,t}$ will be 1. Hence, ILP **P3** is equivalent to INP **P2**. The ILP solution **P3** is applicable only when the problem size is small; otherwise, its runtime is prohibitively long. In the following, we devise a randomized algorithm for the problem when the problem size is large by adopting the randomized rounding technique [13].

We perform linear relaxation for the ILP solution **P3** to a Linear Programming (LP), **P4**, by relaxing the binary variables $x_{i,t}^{(j)}$, $y_{i,m,t}$, and $z_{i,t}$ to real variables of which the values are between 0 and 1. It is noticed that the LP solution can be solved within polynomial time. An integral solution for the original ILP is then derived by randomly rounding the fractional solution with the specified probability [13].

Let $\widetilde{W}_t^{opt}$ be the optimal solution for the linear relaxation LP. Let $\tilde{x}_{i,t}^{(j)}$, $\tilde{y}_{i,m,t}$, and $\tilde{z}_{i,t}$ be the values of $x_{i,t}^{(j)}$, $y_{i,m,t}$, and $z_{i,t}$ delivered by the optimal solution for the LP, respectively,

Algorithm 2 A randomized rounding algorithm to minimize the upper bound of the Lyapunov drift-plus-penalty function at time slot t

Input: The upper bound of the Lyapunov drift-plus-penalty function at time slot t.

Output: Devices make decisions for model downloading and request offloading to minimize the upper bound.

1: Obtain an optimal solution $\widetilde{W}_t^{opt}$ to the LP relaxation of the ILP formulation **P3**. Let $\tilde{x}_{i,t}^{(j)}$, $\tilde{y}_{i,m,t}$, and $\tilde{z}_{i,t}$ be the values of variables delivered by the LP solution;
2: **for** each $v_i \in \mathcal{V}$ **do**
3: $\hat{x}_{i,t}^{(j)} \leftarrow 1$ with probability $\tilde{x}_{i,t}^{(j)}$ for some j with $0 \leq j \leq |N|$;
4: **for** each model $M_m \in \mathcal{M}$ **do**
5: $\hat{y}_{i,m,t} \leftarrow 1$ with probability $\tilde{y}_{i,m,t}$;
6: **end for**;
7: **if** $\hat{x}_{i,t}^{(0)} = 1$ and $\hat{y}_{i,m_i^t,t} = 1$ **then**
8: $\hat{z}_{i,t} \leftarrow 1$ with probability $\frac{\tilde{z}_{i,t}}{\tilde{x}_{i,t}^{(0)} \cdot \tilde{y}_{i,m_i^t,t}}$
9: **end if**;
10: **end for**
11: **return** A feasible solution $\mathcal{S}_t$ to ILP **P3** based on $\hat{x}_{i,t}^{(j)}$, $\hat{y}_{i,m,t}$, and $\hat{z}_{i,t}$ with high probability.

where $\tilde{x}_{i,t}^{(j)} \in [0,1]$, $\tilde{y}_{i,m,t} \in [0,1]$, and $\tilde{z}_{i,t} \in [0,1]$. For the rounding, as the request issued by each device $v_i \in \mathcal{V}_t$ will be offloaded to a cloudlet or be processed locally, then $\hat{x}_{i,t}^{(j)}$ is set to 1 with probability $\tilde{x}_{i,t}^{(j)}$. As each device v_i decides whether to download model M_m or not at the beginning of time slot t, then $\hat{y}_{i,m,t}$ is set to 1 with probability $\tilde{y}_{i,m,t}$. And $\hat{z}_{i,t}$ is set to 1 with probability $\frac{\tilde{z}_{i,t}}{\tilde{x}_{i,t}^{(0)} \cdot \tilde{y}_{i,m_i^t,t}}$ if both $\hat{x}_{i,t}^{(0)}$ and $\hat{y}_{i,m_i^t,t}$ are set to 1. The randomized algorithm to minimize the upper bound of the Lyapunov drift-plus-penalty function at time slot t is detailed in `Algorithm 2`.

D. Algorithm analysis

We now analyze the approximation ratio of `Algorithm 2`. Similarly, let W_t^{opt} be the optimal solutions of the ILP. As the ILP is a minimization problem, then we have $\widetilde{W}_t^{opt} \leq W_t^{opt}$. Recall that the probabilities of setting $\hat{x}_{i,t}^{(j)}$, $\hat{y}_{i,m,t}$, and $\hat{z}_{i,t}$ to 1 are $\mathbf{Pr}[\hat{x}_{i,t}^{(j)} = 1] = \tilde{x}_{i,t}^{(j)}$, $\mathbf{Pr}[\hat{y}_{i,m,t} = 1] = \tilde{y}_{i,m,t}$, and $\mathbf{Pr}[\hat{z}_{i,t} = 1] = \mathbf{Pr}[\hat{z}_{i,t} = 1 | \hat{x}_{i,t}^{(0)} = 1,\ \hat{y}_{i,m_i^t,t} = 1] \cdot \mathbf{Pr}[\hat{x}_{i,t}^{(0)} = 1] \cdot \mathbf{Pr}[\hat{y}_{i,m_i^t,t} = 1] = \frac{\tilde{z}_{i,t}}{\tilde{x}_{i,t}^{(0)} \cdot \tilde{y}_{i,m_i^t,t}} \cdot \tilde{x}_{i,t}^{(0)} \cdot \tilde{y}_{i,m_i^t,t} = \tilde{z}_{i,t}$, respectively. The expectation of the solution delivered by the randomized algorithm is

$$\mathbb{E}[\sum_{v_i\in\mathcal{V}} \sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)} + \sum_{v_i\in\mathcal{V}} \sum_{M_m\in\mathcal{M}} \beta_{i,m,t} \cdot \hat{y}_{i,m,t} - \sum_{v_i\in\mathcal{V}} \gamma_{i,t} \cdot \hat{z}_{i,t}] = \widetilde{W}_t^{opt}, \quad (44)$$

As $\widetilde{W}_t^{opt} = \sum_{v_i\in\mathcal{V}} \sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \tilde{x}_{i,t}^{(j)} + \sum_{v_i\in\mathcal{V}} \sum_{M_m\in\mathcal{M}} \beta_{i,m,t} \cdot \tilde{y}_{i,m,t} - \sum_{v_i\in\mathcal{V}} \gamma_{i,t} \cdot \tilde{z}_{i,t}$.

Denote by $\tilde{\phi}_1 = \sum_{v_i \in \mathcal{V}} (\sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \tilde{x}_{i,t}^{(j)} - \gamma_{i,t} \cdot \tilde{z}_{i,t})$ and $\tilde{\phi}_2 = \sum_{v_i \in \mathcal{V}} \sum_{M_m \in \mathcal{M}} \beta_{i,m,t} \cdot \tilde{y}_{i,m,t}$. Then, $\widetilde{W}_t^{opt} = \tilde{\phi}_1 + \tilde{\phi}_2$. Similarly, ϕ_1 and ϕ_2 are defined as the corresponding components in the optimal solution of the ILP, i.e., $W_t^{opt} = \phi_1 + \phi_2$. Denote by Γ a parameter which is defined as follows.

$$\begin{aligned}\Gamma = \max\{ & \max\{\lambda_{i,t}^{(j)} \mid \forall v_i \in \mathcal{V}, 0 \le j \le |N|\} \\ & \max\{\beta_{i,m,t} \mid \forall v_i \in \mathcal{V}, \forall M_m \in \mathcal{M}\} \\ & \max\{comp_m \mid \forall M_m \in \mathcal{M}\}\}. \end{aligned} \tag{45}$$

Lemma 2: The value of $\sum_{v_i \in \mathcal{V}} (\sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)} - \gamma_{i,t} \cdot \hat{z}_{i,t})$ is no more than $2 \cdot \phi_1$ with high probability of $1 - \frac{1}{|\mathcal{V}|}$.

Proof The detailed proof is shown in the APPENDIX.

Lemma 3: The value of $\sum_{v_i \in \mathcal{V}} \sum_{M_m \in \mathcal{M}} \beta_{i,m,t} \cdot \hat{y}_{i,m,t}$ is no more than $2 \cdot \phi_2$ with high probability of $1 - \frac{1}{|\mathcal{V}|}$.

Proof The proof is similar to that of Lemma 2, omitted.

Theorem 2: There is a randomized algorithm `Algorithm 2` for **P2**, which delivers a 2-approximate solution with high probability of $\min\{1 - \frac{1}{|\mathcal{V}|}, 1 - \frac{1}{|N|}\}$, at the expense of no more than twice the computing resource violation on any cloudlet, provided that $W_t^{opt} \ge 3\Gamma \ln |\mathcal{V}|$, and $\min\{Cap_j \mid \forall j \in N\} \ge 6\Gamma \cdot \ln |N|$, where Γ is defined by Eq. (45), and W_t^{opt} is the optimal solution for **P2**.

Proof The detailed proof is shown in the APPENDIX.

We finally consider the competitive ratio of the online algorithm `Algorithm 1` as follows. Denote by $\widehat{W}_t$ the value of the solution delivered by online algorithm `Algorithm 2` at time slot t.

Lemma 4: To guarantee the constraint of the average energy budget on each device, a virtual queue as a historical measure of energy consumption exceeding the average budget of each device $v_i \in \mathcal{V}$ is defined as $Q_i(t+1) = \max(Q_i(t) + \mathcal{E}_{i,t} - \overline{\mathcal{E}}_i, 0)$, and $Q_i(0) = 0$ initially. Then, we have

$$\lim_{|\mathbb{T}| \to \infty} \frac{1}{|\mathbb{T}|} \sum_{t=0}^{|\mathbb{T}|-1} \sum_{v_i \in \mathcal{V}} \mathbb{E}[Q_i(t)] \le \frac{B + 2V \cdot OPT}{2\varepsilon} \tag{46}$$

where $B = \frac{1}{2} \sum_{v_i \in \mathcal{V}} (\mathcal{E}_i^{max})^2 + \overline{\mathcal{E}}_i^2$ is a constant, $\varepsilon > 0$ is a positive constant representing the gap between the time-average energy consumption and the energy budget per time slot, i.e., $\mathbb{E}[\mathcal{E}_{i,t} - \overline{\mathcal{E}}_i \mid Q_i(t)] \le -\varepsilon$ by the Lyapunov optimization condition, $V > 0$ is a control parameter, and OPT is the expected optimal solution of the freshness and cost minimization problem over time horizon $\mathbb{T}$.

Proof The detailed proof is shown in the APPENDIX.

Theorem 3: Given an MEC network $G = (N, E)$, a set $\mathcal{M}$ of service models, a set $\mathcal{V}$ of devices with each $v_i \in \mathcal{V}$ accommodating a set $\mathcal{M}_i$ of service models, and a time horizon $\mathbb{T}$, let $\mathcal{V}_t$ be the set of devices that issues inference requests at time slot t. Each device $v_i \in \mathcal{V}_t$ will either process its request locally or offload the request to a cloudlet for processing. For any positive control parameter V, the long-term average of the weighted sum of freshness and costs in the solution delivered by online algorithm `Algorithm 1` satisfies

$$\lim_{|\mathbb{T}| \to \infty} \frac{1}{|\mathbb{T}|} \sum_{t=1}^{|\mathbb{T}|} \mathbb{E}[\widehat{W}_t] \le 2 \cdot OPT + \frac{B}{V}, \tag{47}$$

with high probability of $\min\{1 - \frac{1}{|\mathcal{V}|}, 1 - \frac{1}{|N|}\}$, at the expense of no more than twice the computing resource violation on any cloudlet, where $B = \frac{1}{2} \sum_{v_i \in \mathcal{V}} (\mathcal{E}_i^{max})^2 + \overline{\mathcal{E}}_i^2$ is a constant, and OPT is the optimal solution of the problem.

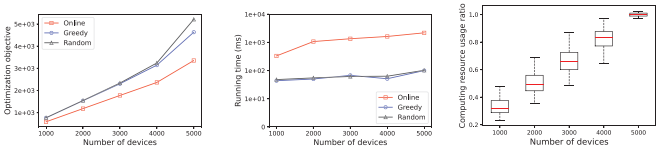
Proof The detailed proof is shown in the APPENDIX.

IV. Performance evaluation

A. Experimental environment settings

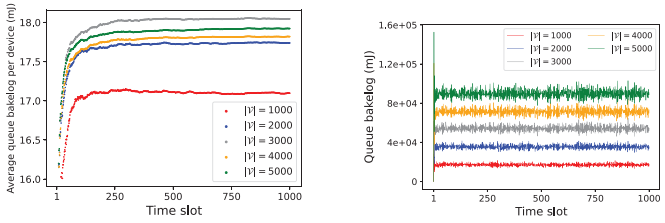
Consider an MEC network instance generated by GT-ITM [6], where the number $|N|$ of APs (and their co-located cloudlets) is set to 200 [11]. The bandwidth capacity of each AP is randomly drawn in [100, 200] Mbps [19]. The computing capacity of each cloudlet is randomly drawn from 1,000 to 2,000 MHz [18]. The cost per MHz computing resource is \$0.0025 [20]. The transmission cost per MB data along each link is randomly drawn in \$[0.0002, 0.0005] [20]. There are 200 service models, and the continual training interval of each model is randomly drawn from 2 to 5 time slots. The amount of computing resource demanded by an instance of each model for inference service on cloudlets is randomly drawn in [200,400] MHz [11]. The parameter size of each model is drawn within [100, 600] MB (e.g., BERT and GPT-2) [4]. There are 2,000 devices in the network, of which the locations are randomly chosen in N. The average energy budget of each device per time slot is randomly drawn in [400, 500] millijoule (mJ) [9]. The cost of each device per mJ energy consumption is randomly drawn in \$[0.0008, 0.0012]. The number $|\mathcal{M}_i|$ of service models deployed on each device v_i is randomly drawn from 5 to 10, and these models are randomly chosen from $\mathcal{M}$. For each device v_i, it issues an inference request at each time slot, and its requested model is randomly chosen from the model set $\mathcal{M}_i$. The size of input data of the request is randomly drawn from 5 to 10 MB [19]. The CPU frequency of each device is randomly drawn in [1.3, 2.5] GHz [12]. The coefficient ξ is set to 10. For request uploading, the uploading power of each device is 23 dBm [20]. The time horizon consists of 1,000 time slots. The positive coefficient α for balancing the weights between the freshness and the cost is set at 0.2. The non-negative control parameter V in the Lyapunov optimization is set to 1,000. Unless otherwise specified, we adopt the above-mentioned parameters by default.

We refer to the online algorithm `Algorithm 1` as `Online`. To evaluate its performance, we proposed the following two comparison benchmarks. One is algorithm `Greedy` that chooses the stalest models to be downloaded by devices at each time slot, subject to the long-term average energy budgets. For each inference request, it will be processed locally or offloaded to a cloudlet with sufficient computing resource



(a) Average weighted sum of inference freshness and cost (b) Running time (c) Computing resource usage ratio

Fig. 2: Performance of different algorithms, by varying the number $|\mathcal{V}|$ of devices.



(a) Average queue backlog per device (b) Queue backlog

Fig. 3: Queue backlogs delivered by `Online`, by varying the number $|\mathcal{V}|$ of devices.

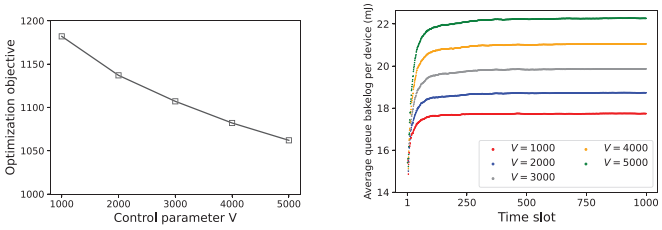
greedily based on the freshness and cost it brings. The other is a randomized algorithm `Random` that randomly chooses models to be downloaded by devices at each time slot, subject to the long-term average energy budgets. And each inference request will be processed locally or offloaded to a cloudlet with sufficient computing resource, the same way as `Greedy`.

The value in each figure is the mean of 20 different network instances with the same size. The running time of each algorithm is obtained in a desktop with 11th Gen Intel(R) Core(TM) i7-11700K @ 3.60 GHz, 64G RAM. We adopted the IBM ILOG CPLEX Optimizer for solving the LPs [8].

B. Performance evaluation of the proposed online algorithm

We first evaluated the performance of `Online` against `Greedy` and `Random`, by varying the number $|\mathcal{V}|$ of devices from 1,000 to 5,000. Fig. 2 plots the optimization objective and running time curves of different algorithms. It can be seen from Fig. 2(a) that when the number $|\mathcal{V}|$ of devices is set to 1,000, `Online` outperforms `Greedy` and `Random` by nearly 23.4% and 23.9%, respectively. When $|\mathcal{V}|$ is set to 5,000, `Online` outperforms `Greedy` and `Random` by 27.6% and 35.6%, respectively. This indicates that when the resource is significantly scarce, `Online` can achieve high resource efficiency. Fig. 2(b) depicts the average running time of `Online`, `Greedy`, and `Random`. `Online` takes the longest running time as it delivers an LP solution for randomized rounding at each time slot. Fig. 2(c) shows that only moderate computing resource violations will be caused by `Online`.

Fig. 3 depicts the queue backlogs delivered by `Online` with different values of $|\mathcal{V}|$. Fig. 3(a) shows the evolution of average queue backlog per device with the increase of time slot t. It can be seen that the average queue backlog becomes stable when t approaches infinity. Hence, we have $\lim_{|\mathbb{T}|\to\infty} \frac{\sum_{v_i\in\mathcal{V}} Q_i(|\mathbb{T}|)}{|\mathbb{T}|} = 0$, i.e., the long-term average energy budget constraint (16) is guaranteed. Fig. 3(b) shows that the queue backlog at each time slot t, i.e., $\sum_{v_i\in\mathcal{V}} Q_i(t)$, fluctuates within a certain interval as t grows. It can be seen from Fig. 3 that when the value of $|\mathcal{V}|$ increases, the queue backlog increases, too, but it takes more time for the average queue backlog to become stable, and the queue backlog fluctuates more dramatically.



(a) Average weighted sum of inference freshness and cost (b) Average queue backlog per device

Fig. 4: Impact of control parameter V on the performance of `Online` for the freshness and cost minimization problem.

C. Impact of control parameter V on the performance of the proposed online algorithm

In the following, we investigated the impact of control parameter V on the performance of the online algorithm `Algorithm` 1, by varying the value of V from 1,000 to 5,000. Fig. 4 plots the performance curve of `Online`, and the curves of average queue backlog per device, while fixing the number of devices at 2,000. It can be seen from Fig. 4(a) that although the optimization objective decreases with the increase of V, the marginal decrease diminishes. Fig. 4(b) shows that the average queue backlog per device is proportional to the value of V.

V. Conclusion

In this paper, we studied freshness-aware hierarchical inference service provisioning in an MEC network between edge servers and resource-constrained devices. We first formulated a novel optimization problem for hierarchical service provisioning - the freshness and cost minimization problem with the aim of maximizing overall inference fidelity while minimizing the total service cost. To this end, we developed an Integer Nonlinear Programming (INP) solution to the offline version of the problem, and then presented an equivalent ILP solution to the INP. We thirdly devised an online algorithm with a provable competitive ratio for the problem, by leveraging the Lyapunov optimization and randomized rounding techniques. We finally evaluated the performance of the proposed online algorithm through simulations, and simulation results demonstrate that the online algorithm outperforms the comparison baselines by at least 23%.

Acknowledgment

The work by Weifa Liang was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China with grant No: CityU 11202723, CityU 11202824, 7005845, 8730094, 9380137, and the CRF grant No: C1042-23GF, respectively.

REFERENCES

[1] X. Ai, W. Liang, Y. Zhang, and W. Xu. Fidelity-aware inference services in DT-assisted edge computing via service model retraining. *IEEE Transactions on Services Computing*, to be published, 2025, doi:10.1109/TSC.2025.3586126

[2] M. A. Ameddah, B. Das, and J. Almhana. Cloud-assisted real-time road Condition monitoring system for vehicles. *Proc. of GLOBECOM'18*, pp. 1 – 6, 2018.

[3] H. B. Beytur, A. G. Aydin, G. de Veciana, and H. Vikalo. Optimization of offloading policies for accuracy-delay tradeoffs in hierarchical inference. *Proc. of INFOCOM'24*, IEEE, pp. 1989 – 1998, 2024.

[4] H. Dai, J. Wu, Y. Wang, J. Yen, Y. Zhang, and C. Xu. Cost-efficient sharing algorithms for DNN model serving in mobile edge networks. *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2517 – 2531, 2023.

[5] A. Fresa and J. P. Champati. Offloading algorithms for maximizing inference accuracy on edge device in an edge intelligence system. *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 7, pp. 2025 – 2039, 2023.

[6] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/, 2019.

[7] P. Han, S. Wang, Y. Jiao, and J. Huang. Federated learning while providing model as a service: Joint training and inference optimization. *Proc. of INFOCOM'24*, IEEE, pp. 631 – 640, 2024.

[8] IBM ILOG CPLEX Optimizer. https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer, 2024.

[9] Y. Li, W. Liang, J. Li, X. Cheng, D. Yu, A. Y. Zomaya, and S. Guo. Energy-aware, device-to-device assisted federated learning in edge computing. *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 7, pp. 2138 – 2154, 2023.

[10] H. Liu, S. Liu, S. Long, Q. Deng, and Z. Li. Joint optimization of model deployment for freshness-aware task assignment in edge intelligence. *Proc. of INFOCOM'24*, IEEE, pp. 1751 – 1760, 2024.

[11] J. Li, W. Liang, W. Xu, Z. Xu, X. Jia, W. Zhou, and J. Zhao. Maximizing user service satisfaction for delay-sensitive IoT applications in edge computing. *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 5, pp. 1199 – 1212, 2022.

[12] Y. Luo, L. Pu, and C. -H. Liu. CPU frequency scaling optimization in sustainable edge computing. *IEEE Transactions on Sustainable Computing*, vol. 8, no. 2, pp. 194 – 207, 2023.

[13] Y. Li, D. Zeng, L. Gu, M. Ou, and Q. Chen. On efficient Zygote container planning toward fast function startup in serverless edge cloud. *Proc. of INFOCOM'23*, IEEE, pp. 1 – 9, 2023.

[14] K. Luo, K. Zhao, T. Ouyang, X. Zhang, Z. Zhou, H. Wang, and X. Chen. Efficient coordination of federated learning and inference offloading at the edge: A proactive optimization paradigm. *IEEE Transactions on Mobile Computing*, vol. 24, no. 1, pp. 407 – 421, 2025.

[15] V. N. Moothedath, J. P. Champati, and J. Gross. Online algorithms for hierarchical inference in deep learning applications at the edge. arXiv:2304.00891, 2023.

[16] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge Univ. Press, 2005.

[17] J. Wu, J. Guo, Z. Tang, C. Luo, T. Wang, and W. Jia. Sequence-aware online container scheduling with reinforcement learning in parked vehicle edge computing. *IEEE Transactions on Vehicular Technology*, doi: 10.1109/TVT.2025.3554595, to be published.

[18] Z. Xu, D. Li, W. Liang, W. Xu, Q. Xia, P. Zhou, O. F. Rana, and H. Li. Energy or accuracy? Near-optimal user selection and aggregator placement for federated learning in MEC. *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2470 – 2485, 2024.

[19] Z. Xu, L. Zhou, H. Dai, W. Liang, W. Zhou, P. Zhou, W. Xu, and G. Wu. Energy-aware collaborative service caching in a 5G-enabled MEC with uncertain payoffs. *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1058 – 1071, 2022.

[20] Y. Zhang, W. Liang, Z. Xu, W. Xu, and M. Chen. AoI-aware inference services in edge computing via digital twin network slicing. *IEEE Transactions on Services Computing*, vol. 17, no. 6, pp. 3154 – 3170, 2024.

APPENDIX

A. Proof for Lemma 1

By Eq. (23), we have

$$Q_i(t+1)^2 \le (Q_i(t) + \mathcal{E}_{i,t} - \overline{\mathcal{E}_i})^2$$
$$\le Q_i(t)^2 + 2Q_i(t)(\mathcal{E}_{i,t} - \overline{\mathcal{E}_i}) + (\mathcal{E}_{i,t} - \overline{\mathcal{E}_i})^2$$
$$\le Q_i(t)^2 + 2Q_i(t)(\mathcal{E}_{i,t} - \overline{\mathcal{E}_i}) + (\mathcal{E}_{i,t})^2 + \overline{\mathcal{E}_i}^2.$$

Then,

$$\Delta L(t) + V \cdot W_t = L(t+1) - L(t) + V \cdot W_t$$
$$= \frac{1}{2}\sum_{v_i \in \mathcal{V}} (Q_i(t+1)^2 - Q_i(t)^2) + V \cdot W_t$$
$$\le B + \sum_{v_i \in \mathcal{V}} Q_i(t)(\mathcal{E}_{i,t} - \overline{\mathcal{E}_i}) + V \cdot W_t.$$

B. Proof for Lemma 2

Let α_1 be a constant with $0 < \alpha_1 \le 1$. We have

$$\mathbf{Pr}[\sum_{v_i \in \mathcal{V}} (\sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)} - \gamma_{i,t} \cdot \hat{z}_{i,t}) \ge (1+\alpha_1) \cdot \phi_1]$$
$$\le \mathbf{Pr}[\sum_{v_i \in \mathcal{V}} (\sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)} - \gamma_{i,t} \cdot \hat{z}_{i,t}) \ge (1+\alpha_1) \cdot \tilde{\phi}_1],$$
since $\tilde{\phi}_1 \le \phi_1$
$$= \mathbf{Pr}[\sum_{v_i \in \mathcal{V}} X_{i,t} \ge (1+\alpha_1) \cdot \frac{\tilde{\phi}_1}{\Gamma}],$$
let $X_{i,t} = \dfrac{\sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)} - \gamma_{i,t} \cdot \hat{z}_{i,t}}{\Gamma}$
$$\le \exp(\frac{-\alpha_1^2 \cdot \tilde{\phi}_1}{(2+\alpha_1) \cdot \Gamma}), \tag{48}$$

where Ineq. (48) is obtained by the Chernoff Bound [16]: variable $X_{i,t}$ is independent from each other and $0 \le X_{i,t} \le 1$, as $X_{i,t} \le \frac{\sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)}}{\Gamma} \le \frac{\sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)}}{\max\{\lambda_{i,t}^{(j)} \mid \forall v_i \in \mathcal{V}, 0 \le j \le |N|\}} \le 1$. And $\mathbb{E}[\sum_{v_i \in \mathcal{V}} X_{i,t}] = \frac{\tilde{\phi}_1}{\Gamma}$.

Assume $\exp(\frac{-\alpha_1^2 \cdot \tilde{\phi}_1}{3 \cdot \Gamma}) \le \frac{1}{|\mathcal{V}|}$. We have $\exp(\frac{-\alpha_1^2 \cdot \tilde{\phi}_1}{(2+\alpha_1) \cdot \Gamma}) \le \exp(\frac{-\alpha_1^2 \cdot \tilde{\phi}_1}{3 \cdot \Gamma}) \le \frac{1}{|\mathcal{V}|}$ as $0 < \alpha_1 \le 1$. Since $\tilde{\phi}_1 \le \phi_1 \le W_t^{opt}$, then

$$\alpha_1 \ge \sqrt{\frac{3\Gamma \cdot \ln|\mathcal{V}|}{\tilde{\phi}_1}} \ge \sqrt{\frac{3\Gamma \cdot \ln|\mathcal{V}|}{W_t^{opt}}}.$$

Since $\alpha_1 \le 1$, there must be $W_t^{opt} \ge 3\Gamma \cdot \ln|\mathcal{V}|$. Hence, the value of $\sum_{v_i \in \mathcal{V}} (\sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)} - \gamma_{i,t} \cdot \hat{z}_{i,t})$ is less than $2 \cdot \phi_1$ with high probability of $1 - \frac{1}{|\mathcal{V}|}$, as $1 + \alpha_1 \le 2$.

C. Proof for Theorem 2

We first analyze the approximation ratio of `Algorithm 2` as follows.

Let $\alpha_{\max} = \max\{\alpha_1, \alpha_2\}$. By Lemma 2 and 3, we have

$$\mathbf{Pr}[\sum_{v_i \in \mathcal{V}} \sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)} + \sum_{v_i \in \mathcal{V}} \sum_{M_m \in \mathcal{M}} \beta_{i,m,t} \cdot \hat{y}_{i,m,t}$$
$$- \sum_{v_i \in \mathcal{V}} \gamma_{i,t} \cdot \hat{z}_{i,t} \ge (1+\alpha_{\max}) \cdot W_t^{opt}]$$
$$\le \mathbf{Pr}[\sum_{v_i \in \mathcal{V}} \sum_{j=0}^{|N|} \lambda_{i,t}^{(j)} \cdot \hat{x}_{i,t}^{(j)} + \sum_{v_i \in \mathcal{V}} \sum_{M_m \in \mathcal{M}} \beta_{i,m,t} \cdot \hat{y}_{i,m,t}$$

$$-\sum_{v_i\in\mathcal{V}}\gamma_{i,t}\cdot\hat{z}_{i,t}\geq(1+\alpha_{\max})\cdot\widetilde{W}_t^{opt}],\ \text{as}\ \widetilde{W}_t^{opt}\leq W_t^{opt}$$
$$\leq\frac{1}{|\mathcal{V}|}.$$

Hence, the solution delivered by `Algorithm 2` is no greater than twice the optimal one with high probability of $1-\frac{1}{|\mathcal{V}|}$, as $0<\alpha_{\max}\leq 1$.

Similarly, it can be shown that the computing resource consumption on any cloudlet j is no more than twice its capacity with high probability $1-\frac{1}{|N|}$, by leveraging the union bound inequality and Chernoff bound. Hence, the proof of computing resource violations of the solution delivered by `Algorithm 2` is omitted.

D. Proof for Lemma 4

We define $\widehat{\mathcal{E}}_{i,t}$, $\mathcal{E}_{i,t}^{opt}$, and $\mathcal{E}_{i,t}^{*}$ the energy consumption of each device $v_i\in\mathcal{V}$ at time slot t, which is delivered by the approximation algorithm, the optimal solution of the ILP **P3**, and the optimal solution of **P1**, respectively. And we define OPT_t as the value of W_t delivered by the optimal solution of **P1** at t. By Lemma 1, we have

$$\Delta L(t)\leq B+\sum_{v_i\in\mathcal{V}}Q_i(t)\cdot(\widehat{\mathcal{E}}_{i,t}-\overline{\mathcal{E}_i})+V\cdot\widehat{W}_t.$$

By Eq.(24) and (25), we have

$$\sum_{t=0}^{|\mathbb{T}|-1}\Delta L(t)=L(|\mathbb{T}|)-L(0)=L(|\mathbb{T}|),\quad\text{as}\ L(0)=0.$$

Then,

$$\mathbb{E}[L(|\mathbb{T}|)]\leq B\cdot|\mathbb{T}|+\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]\cdot\mathbb{E}[\widehat{\mathcal{E}}_{i,t}-\overline{\mathcal{E}_i}|Q_i(t)]$$
$$+\sum_{t=0}^{|\mathbb{T}|-1}V\cdot\mathbb{E}[\widehat{W}_t|Q_i(t)]$$
$$\leq B\cdot|\mathbb{T}|+2\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]\cdot\mathbb{E}[\mathcal{E}_{i,t}^{opt}-\overline{\mathcal{E}_i}|Q_i(t)]$$
$$+2\sum_{t=0}^{|\mathbb{T}|-1}V\cdot\mathbb{E}[W_t^{opt}|Q_i(t)],\text{by Theorem 2}$$
$$\leq B\cdot|\mathbb{T}|+2\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]\cdot\mathbb{E}[\mathcal{E}_{i,t}^{*}-\overline{\mathcal{E}_i}|Q_i(t)]$$
$$+2\sum_{t=0}^{|\mathbb{T}|-1}V\cdot\mathbb{E}[OPT_t|Q_i(t)],\text{as}\ W_t^{opt}\ \text{is conditional optimal.}$$

The expected energy consumption of any device v_i at time slot t satisfies $\mathbb{E}[\mathcal{E}_{i,t}-\overline{\mathcal{E}_i}]\leq-\varepsilon$ with $\varepsilon>0$. Since $\mathbb{E}[L(|\mathbb{T}|)]\geq 0$ and $\mathbb{E}[OPT_t|Q_i(t)]=OPT_t$, we have

$$B\cdot|\mathbb{T}|-2\varepsilon\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]+2\sum_{t=0}^{|\mathbb{T}|-1}V\cdot OPT_t\geq 0.$$
$$\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]\leq\frac{1}{2\varepsilon}(B\cdot|\mathbb{T}|+2\sum_{t=0}^{|\mathbb{T}|-1}V\cdot OPT_t)\quad(49)$$

By dividing $|\mathbb{T}|$ and taking limit with $|\mathbb{T}|\to\infty$ on both sides of Ineq. (49), we have

$$\lim_{|\mathbb{T}|\to\infty}\frac{1}{|\mathbb{T}|}\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]$$
$$\leq\frac{1}{2\varepsilon}(B+\lim_{|\mathbb{T}|\to\infty}\frac{2}{|\mathbb{T}|}\sum_{t=0}^{|\mathbb{T}|-1}V\cdot OPT_t)$$
$$\leq\frac{B+2V\cdot OPT}{2\varepsilon}$$

Lemma 4 then follows.

E. Proof for Theorem 3

By adopting the similar technique to the proof of Lemma 4, we show the claim is correct as follows.

By Eq. (25), Ineq. (27), and Theorem 2, the following inequality holds with probability of $\min\{1-\frac{1}{|\mathcal{V}|},1-\frac{1}{|N|}\}$

$$\mathbb{E}[L(|\mathbb{T}|)]+V\sum_{t=0}^{|\mathbb{T}|-1}\mathbb{E}[\widehat{W}_t|Q_i(t)]$$
$$\leq B\cdot|\mathbb{T}|+V\sum_{t=0}^{|\mathbb{T}|-1}\mathbb{E}[\widehat{W}_t|Q_i(t)]$$
$$+\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]\cdot\mathbb{E}[\widehat{\mathcal{E}}_{i,t}-\overline{\mathcal{E}_i}|Q_i(t)]$$
$$\leq B\cdot|\mathbb{T}|+2V\sum_{t=0}^{|\mathbb{T}|-1}\mathbb{E}[W_t^{opt}|Q_i(t)]$$
$$+2\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]\cdot\mathbb{E}[\mathcal{E}_{i,t}^{opt}-\overline{\mathcal{E}_i}|Q_i(t)]$$
$$\leq B\cdot|\mathbb{T}|+2V\sum_{t=0}^{|\mathbb{T}|-1}\mathbb{E}[OPT_t|Q_i(t)]$$
$$+2\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)]\cdot\mathbb{E}[\mathcal{E}_{i,t}^{*}-\overline{\mathcal{E}_i}|Q_i(t)]$$
$$\leq B\cdot|\mathbb{T}|+2V\sum_{t=0}^{|\mathbb{T}|-1}OPT_t-2\varepsilon\sum_{t=0}^{|\mathbb{T}|-1}\sum_{v_i\in\mathcal{V}}\mathbb{E}[Q_i(t)].$$

As $\mathbb{E}[L(|\mathbb{T}|)]\geq 0$, we have

$$\lim_{|\mathbb{T}|\to\infty}\frac{1}{|\mathbb{T}|}\sum_{t=0}^{|\mathbb{T}|-1}\mathbb{E}[\widehat{W}_t]\leq\frac{B}{V}+2\lim_{|\mathbb{T}|\to\infty}\frac{1}{|\mathbb{T}|}\sum_{t=0}^{|\mathbb{T}|-1}OPT_t.$$

Hence,

$$\lim_{|\mathbb{T}|\to\infty}\frac{1}{|\mathbb{T}|}\sum_{t=0}^{|\mathbb{T}|-1}\mathbb{E}[\widehat{W}_t]\leq\frac{B}{V}+2\cdot OPT$$

The theorem then follows.