An Online Market Mechanism for Edge Emergency Demand Response via Cloudlet Control

Shutong Chen[1] Lei Jiao[2] Lin Wang[3,4] Fangming Liu*[1]
[1]Key Laboratory of Services Computing Technology and System, Ministry of Education,
School of Computer Science and Technology, Huazhong University of Science and Technology, China
[2]University of Oregon, USA [3]VU Amsterdam, The Netherlands [4]TU Darmstadt, Germany

Abstract—**The computing frontier is moving from centralized mega datacenters towards distributed cloudlets at the network edge. We argue that cloudlets are well-suited for participation in Emergency Demand Response (EDR) programs due to their enormous energy consumption and flexible workload distribution, while existing EDR mechanisms for clouds and colocation datacenters are not suitable for cloudlets. We propose a novel online market mechanism, EdgeEDR, to incentivize cloudlets to participate in EDR, featuring multiple cloudlet-specific designs. At a high level, we observe that cloudlet operators can dynamically switch on/off entire cloudlets to compensate for the energy reduction required by the power grid. We formulate a long-term social cost minimization problem and decompose it into a series of one-round procurement auctions. In each auction instance, we propose to let the cloudlet tenants bid with cost functions of their service quality degradation tolerance, and let the cloudlet operator choose the service quality, allocate the workload, and shut down the cloudlets. Via rigorous analysis, we exhibit that our bidding policy is individually rational and truthful; our workload distribution algorithm has near-optimal performance in each auction; and our overall online algorithm achieves a provable competitive ratio. We further confirm the performance of our mechanism through extensive trace-driven simulations.**

I. Introduction

Cloudlets are the key infrastructures to realizing the promise of edge computing [1], [2]. Often in the forms of small data centers, machine rooms, and server clusters, cloudlets can provide low latency, service redundancy, and data privacy to end users from the Internet edge such as metro stations, enterprise premises, cellular towers, and WiFi neighborhoods. Due to the massive existence, a network of cloudlets consume significant energy from the power grid; moreover, the wide distribution of cloudlets contributes to more flexibility in their workload management. These characteristics make cloudlets well-suited for participation in Emergency Demand Response (EDR) programs to help the power grid maintain stability, reliability, and security [3]–[5]. In a typical EDR program, at certain times such as the peak hours when the grid is under pressure, it sends signals to the participants in terms of power caps or energy reduction goals, and in response the participants mandatorily reduce their energy consumption and also receive financial rewards as contracted from the grid.

Similar to an Infrastructure-as-a-Service (IaaS) cloud, not all resources of an IaaS cloudlet are controlled by a single party—the cloudlet facilities/hardware (e.g., servers) and the cloudlet software (e.g., virtual machines) are often operated by the cloudlet operator and the tenants (i.e., service providers who provide services to end users), respectively. This creates the so-called "split incentives" hurdle [3], [6]. For the EDR to work, both the cloudlet operator and the tenants would need to participate. However, tenants often have no motivation to join, because their concerns are not about reducing energy but about obtaining software resources guaranteed by the cloudlet via paying usage fees to the cloudlet operator. This hurdle also exists in "colocation" data centers, where tenants own and run their servers, together with virtual machines and services.

Many efforts have been devoted to designing mechanisms to address split incentives for demand response; however, existing solutions have limitations, and are unsuitable for cloudlets. First, they lack flexibility in procurements and are restrictive in adapting to the changing market conditions. Auction-based mechanisms procure tenants' bids of fixed commitments [3] while rewards-based mechanisms set reward rates and accept tenants' offers obliviously [7], [8]. Second, they mostly assume tenants to reduce their workload to reduce power consumption [6], [9] rather than manage tenants' workload themselves, while cloudlet tenants often provide end-user-facing services and are concerned about reducing workload. The only workload-aware mechanism known to us explores temporal flexibility for batch jobs [4], [5], which does not match the type of workload of cloudlets with spacial flexibility for workload distribution. Third, they primarily rely on costly, environment-unfriendly, diesel-powered generators to compensate for the energy reduction required by the grid. As EDR becomes more frequent [4], the current reward from the grid may not suffice to cover the generation cost; operators are actually cutting the power infrastructure investment by down-sizing the capacity of generators, which could compromise the EDR capability [9]. Thus, it is intriguing to seek other methods to compensate for energy reduction.

For cloudlets, we make two key observations. Our first observation is that a mechanism for cloudlets can explore flexible service quality degradation without rejecting work-

*The corresponding author is Fangming Liu (fmliu@hust.edu.cn).

This work was supported in part by the National Key Research & Development (R&D) Plan under grant 2017YFB1001703, in part by NSFC under Grant 61722206 and 61761136014 (and 392046569 of DFG) and 61520106005, in part by the Fundamental Research Funds for the Central Universities under Grant 2017KFKJXX009 and 3004210116, and in part by the National Program for Support of Top-notch Young Professionals in National Program for Special Support of Eminent Professionals.

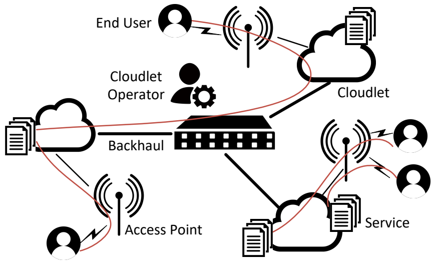

Fig. 1. An example of the cloudlet network.

load. As in Fig. 1, cloudlets are often connected to one another via local area networks [10], with network delay usually one order of magnitude smaller than the remote clouds. Therefore, end user requests can be moved and processed at locations different from where they originate with moderate additional delay. A procurement-auction-based mechanism, for example, can let each tenant bid with a cost function of the delay degradation tolerance and let the operator choose the desired delay quality for each tenant and distribute the workload in the cloudlet network accordingly. Inspired by divisible-goods auctions with finite-capacity suppliers [11], this design achieves adaptability in procurements; compared to other methods such as “supply functions”, it is also function-based, but allows further adaptability by requiring no unified market clearing price [9], [12].

Our next observation is that the cloudlet operator can switch off entire cloudlets to compensate for energy reduction [13]. Non-IT appliances (e.g., cooling and lighting) can consume $33\% \sim 52\%$ of the total energy of a cloudlet of up to 500 servers [14]; thus, moving workloads around to empty entire cloudlets and shutting them down can eliminate the considerable non-IT energy. However, switching on and off cloudlets frequently can incur considerable “switching” penalty such as start-up delay, system oscillation, and hardware wear-and-tear [13], [15], [16], which may hurt the power saving benefit. Thus, the operator needs to carefully strike the balance between energy reduction and the switching cost for conducting online auctions with no information about the future available.

In this paper, to the best of our knowledge, we propose the first online auction mechanism — EdgeEDR, specifically for performing EDR at the edge. Our proposal features a set of unique designs for cloudlets to meet the split incentives, the adaptability, and the greenness requirements. Based on our observations, we make multiple contributions:

We build models to capture the cloudlet operator’s and the tenants’ costs, design online procurement auctions, and formulate the long-term social cost minimization problem. Particularly, the cloudlet operator has the switching cost incurred by the dynamic control of cloudlets, in addition to cloudlets’ maintenance cost and the possible cost of local power generation. Assuming truthfulness of bidding, our online auction mechanism at each time slot solicits a bid from each tenant in the form of a cost function based on the tenant’s tolerance range for any additional delay incurred and a per-unit delay penalty within that range. We minimize the long-term social cost as a mixed-integer nonlinear program, subject to cloudlet capacities, EDR demand satisfaction, and service delay tolerance.

We design an online algorithmic framework, EdgeEDR, to solve our social cost minimization problem and also a procurement auction mechanism to ensure individual rationality and truthful bids. Our online algorithm framework solves the one-shot NP-hard problem at each time slot using a primal-dual-based, polynomial-time approximation algorithm, obtains a near-optimal cloudlets switching solution, and then postpones cloudlets switching as required by the near optimum for avoiding excessive switching cost. We rigorously prove that the social cost over time incurred by EdgeEDR is no greater than a parameterized constant (i.e., the competitive ratio) times the offline optimal social cost in the worst case. Our procurement auction mechanism provably guarantees individual rationality and truthfulness by following the “bid-monotonic” and the “critical” sufficient conditions.

We finally conduct extensive evaluations using real-world data traces. We simulate a real EDR event in 2014 and use London’s underground network to simulate the edge system consisting of APs and cloudlets. The results show that the auction-based EdgeEDR has a high effectiveness in improving long-term social welfare of both the cloudlet operator and tenants in EDR event. Moreover, EdgeEDR achieves more than 50% saving in local generation power compared to no tenant-incentive EDR mechanism. A small-scale simulation also indicates a great empirical competitive performance of EdgeEDR.

II. Model and Problem Formulation

System Settings. We consider a system that consists of a set of distributed heterogenous cloudlets $\mathcal{N} = \{0, 1, ..., N\}$ which are connected to each other through a wireline backhaul, a set of tenants or service providers $\mathcal{M} = \{0, 1, ..., M\}$ who operate and provide services to end users, and a set of Access Points (APs) $\mathcal{S} = \{0, 1, ..., S\}$ via which end users can access the services deployed at any cloudlet in the system. We denote the capacity of cloudlet k as R_k, $\forall k \in \mathcal{N}$. We represent the time horizon with multiple continuous time slots as $\mathcal{T} = \{0, 1, ..., T\}$. We use a binary variable x_k^t to indicate whether the cloudlet k is active ($x_k^t = 1$) or not ($x_k^t = 0$) at time slot t, $\forall t \in \mathcal{T}$. We denote the workload originated from end users via AP i, $\forall i \in \mathcal{S}$ towards service provider j, $\forall j \in \mathcal{M}$ at time slot t by λ_{ij}^t, and denote the maximum delay λ_{ij}^t can tolerate by D_{ij}^t. The cloudlet operator of this IaaS cloudlet system, as discussed in Sec. I, determines the distribution of the workload of each service provider. We use a binary variable y_{jik}^t to indicate whether the workload λ_{ij}^t is allocated to cloudlet k ($y_{jik}^t = 1$) or not ($y_{jik}^t = 0$). Without loss of generality, we assume each service provider operates only one service, as multiple services can be treated as multiple “virtual” service providers correspondingly; for the ease of management, we also assume the end users’ workload from any AP i for any service j at one time slot is processed at one and only one cloudlet.

TABLE I
LIST OF NOTATIONS

Notations	Description
M	# of service providers
N	# of cloudlets
S	# of APs
T	# of time slots
D_{ij}^t	maximum delay tolerance of workload λ_{ij}^t
P_{EDR}^t	EDR requirement at time slot t
p	fuel cost of the local generator for one-unit power
x_k^t	cloudlet k is active at time slot t or not
y_{jik}^t	workload λ_{ij}^t is allocated to cloudlet k or not
z_{ij}^t	delay degradation of workload λ_{ij}^t
r_{ij}^t	payment to workload λ_{ij}^t
λ_{ij}^t	workload originated via AP i to service provider j at t
a_{ijk}^t	value evaluation of allocating workload λ_{ij}^t to cloudlet k
θ_{ij}^t	maximum delay degradation workload λ_{ij}^t can tolerate
c_{ij}^t	per-unit delay penalty when the delay exceeds D_{ij}^t
u_g^t	# of local generation
α_k	start-up cost of activating the cloudlet k
ς	maintaining cost of one active cloudlet
e_k^t	power consumption of cloudlet k at time slot t
d_{ijk}^t	the difference between delay tolerance D_{ij}^t and propagation delay from AP i to cloudlet k

When an EDR event bursts, based on the agreement signed between the cloudlet operator and the utility (e.g., PJM [17]), the cloudlet operator should cut down a certain level of its own power demand from the utility. When receiving the EDR signal, the cloudlet operator can shed energy in two ways: consolidating workload to shut down cloudlets while guaranteeing service quality, and using local generation.

Auction Design. As illustrated in Fig. 2, at the beginning of the time slot t within the EDR frame, the operator, who acts as the auctioneer, solicits a bid from each service provider. Then, in addition to the workload λ_{ij}^t and the delay tolerance D_{ij}^t, service provider j voluntarily submits a service quality reduction bid as $\{(\theta_{ij}^t, c_{ij}^t) | i \in \mathcal{S}\}$, where θ_{ij}^t is the maximum delay degradation that service provider j can tolerate for end user workload originated from AP i, and c_{ij}^t is the per-unit delay penalty when the delay exceeds D_{ij}^t. After receiving the bids, the operator determines the cloudlet activation status $\{x_k^t | k \in \mathcal{N}\}$ and the workload distribution $\{y_{jik}^t | i \in \mathcal{S}, j \in \mathcal{M}, k \in \mathcal{N}\}$, notifies the winning bids and payments $\{(z_{ij}^t, r_{ij}^t) | i \in \mathcal{S}, j \in \mathcal{M}\}$, and sets the amount of local generation u_g^t. In this work, we assume each cloudlet is equipped with local generators which can fully support the cloudlet's power demand. For simplicity, we regard these generators as a large virtual generator and we only consider the total amount of generation. The objective of EdgeEDR is to meet the EDR requirement with the long-term minimum social operational cost which consists of the operational costs of both the operator and the service provider.

Operational Cost of the Cloudlet Operator. The operator's total operational cost consists of the following components: 1) the local generation cost of pu_g^t, where p is the fuel cost of the local generator for one-unit power; 2) the switching cost of $\sum_k \alpha_k [x_k^t - x_k^{t-1}]^+$, where $[\cdot]^+ = max\{0, \cdot\}$, and α_k is the start-up cost of activating the cloudlet k; 3) the maintenance cost for active cloudlets of $\varsigma \sum_k x_k^t$, where ς is the sunk cost of maintaining one active cloudlet at one time slot; 4) the payment of $\sum_i \sum_j r_{ij}^t$ to service providers.

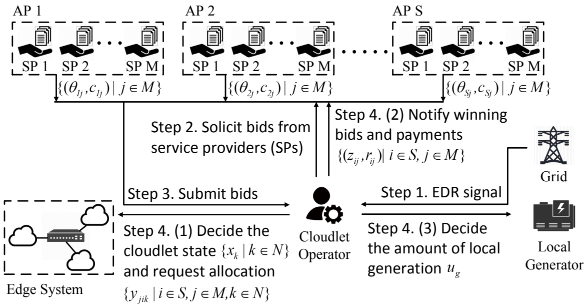
AP 1
AP 2
AP S
SP 1
SP 2
SP M
$\{(\theta_{1j}, c_{1j}) | j \in M\}$
$\{(\theta_{2j}, c_{2j}) | j \in M\}$
$\{(\theta_{Sj}, c_{Sj}) | j \in M\}$
Step 2. Solicit bids from service providers (SPs)
Step 4. (2) Notify winning bids and payments $\{(z_{ij}, r_{ij}) | i \in S, j \in M\}$
Grid
Step 3. Submit bids
Step 1. EDR signal
Step 4. (1) Decide the cloudlet state $\{x_k | k \in N\}$ and request allocation $\{y_{jik} | i \in S, j \in M, k \in N\}$
Cloudlet Operator
Step 4. (3) Decide the amount of local generation u_g
Local Generator
Edge System

Fig. 2. An illustration of our auction mechanism in one round.

Operational Cost of Service Providers. The operational cost of service provider j is $\sum_i (f_{ij}(z_{ij}^t) - r_{ij}^t - \sum_k a_{ijk}^t y_{jik}^t)$. For the end user workload submitted from AP i, r_{ij}^t is the payment received from the operator; a_{ijk}^t indicates service provider j's evaluation of the value of allocating such workload to cloudlet k, which could capture the varying levels of service reliability for different workload distribution, for example; $f_{ij}(z_{ij}^t)$ is the penalty function of the delay violation z_{ij}^t:

$$f_{ij}(z_{ij}^t) = \begin{cases} c_{ij}^t z_{ij}^t, & z_{ij}^t \leqslant \theta_{ij}^t \\ +\infty, & z_{ij}^t > \theta_{ij}^t \end{cases}.$$

Problem Formulation. Having the operational costs of the cloudlet operator and the service providers, we formulate the social cost minimization (or the social welfare maximization, equivalently [18]) problem to determine the workload distribution, the activation status of each cloudlet, the delay degradation of each service, and the amount of local generation, where the payments of the operator and the service providers cancel one another:

$$\begin{aligned} \max \quad & \sum_t \Big(\sum_i \sum_j \sum_k a_{ijk} y_{jik}^t - \sum_i \sum_j f_{ij}(z_{ij}^t) - \\ & pu_g^t - \varsigma \sum_k x_k^t - \sum_k \alpha_k [x_k^t - x_k^{t-1}]^+ \Big) \qquad (1) \\ \text{s.t.} \quad & \sum_i \sum_j \lambda_{ij}^t y_{jik}^t \leqslant R_k x_k^t, \forall k \in \mathcal{N}, \forall t \in \mathcal{T} \qquad (1a) \\ & \sum_k e_k^t \leqslant P_{EDR}^t + u_g^t, \forall t \in \mathcal{T} \qquad (1b) \\ & \sum_k d_{ijk}^t y_{jik}^t \leqslant z_{ij}^t, \forall i \in \mathcal{M}, \forall j \in \mathcal{M}, \forall t \in \mathcal{T} \qquad (1c) \\ & \sum_k y_{jik}^t \leqslant 1, \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall t \in \mathcal{T} \qquad (1d) \\ & y_{jik}^t \in \{0, 1\}, \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall k \in \mathcal{N}, \forall t \in \mathcal{T} \qquad (1e) \\ & 0 \leqslant z_{ij}^t \leqslant \theta_{ij}^t, \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall t \in \mathcal{T} \qquad (1f) \\ & u_g^t \geqslant 0, \forall t \in \mathcal{T} \qquad (1g) \end{aligned}$$

For clarity, the important notations are listed in Table I. Here, the power consumption of cloudlet k is represented by $e_k^t = (L_k P_{idle}^k x_k^t + (P_{peak}^k - P_{idle}) \sum_i \sum_j \lambda_{ij}^t y_{jik}^t) \cdot \text{PUE}_k$, where PUE_k refers to the power usage effectiveness (PUE) of cloudlet k, L_k is the number of servers in cloudlet k, and P^k_{idle} and P^k_{peak} denote the server's power in the idle and the peak utilization, respectively. For simplicity, we reformulate e^t_k as $e^t_k = \beta_k \sum_i \sum_j \lambda^t_{ij} y^t_{jik} + \gamma_k x^t_k$, where $\beta_k = (P^k_{peak} - P_{idle})\mathrm{PUE}_k$ and $\gamma_k = L_k P^k_{idle}\mathrm{PUE}_k$. In the constraint (1c), $d^t_{ijk} = \max\{0, l_{ik} - D^t_{ij}\}$, where l_{ik} is the propagation delay from AP i to cloudlet k. d^t_{ijk} denotes the difference between the delay tolerance D^t_{ij} and the propagation delay from AP i to cloudlet k, and $d^t_{ijk} = 0$ indicates the propagation delay between i and k is within the delay tolerance D^t_{ij}. Note that in this paper, we only consider that different workload allocation schemes affect the propagation latency between the end user and the cloudlet. For the future work, the effect of workload allocation on the computation time, which is caused by the different computation power of heterogeneous cloudlets, will be taken into account.

In problem (1), the constraint (1a) ensures that when the cloudlet k is active, k's resource utilization is under its capacity; if k is switched off, any workload cannot be allocated to it. The constraint (1b) guarantees that the total power demand does not exceed the sum of the EDR power cap and the local generation. The constraints (1c) and (1f) guarantee that the delay degradation does not exceed the bound θ^t_{ij}. The constraint (1d) and constraint (1e) guarantee that workload is allocated to at most one cloudlet. Later in Sec. III, we will show that the workload is indeed ensured to be allocated to one and only one cloudlet by the algorithm design.

Algorithmic Challenges. Solving the social welfare maximization problem in an online manner is highly challenging. Although the cloudlet operator can switch off cloudlets for energy saving, high switching cost may be incurred by frequent cloudlet activations. Without knowledge of the future inputs, it is nontrivial for the cloudlet operator to determine the cloudlet activation status at each time slot, because a decision at a time slot will influence the switching cost between that time slot and its next time slot; as the next time slot has not yet arrived, it is not easy to make a good decision for the current time slot. Note that the problem (1) subsumes a case of $z^t_{ij} = 0$ and $u^t_g = 0$, $\forall i \in \mathcal{S}$, $\forall j \in \mathcal{M}$, $\forall t \in \mathcal{T}$, where it reduces to a NP-hard multi-dimensional knapsack problem [19]. So even in the offline scenario where all the inputs are known in advance, the problem is still NP-hard in general. To guarantee individually rational and truthful bidding, we would need to leverage the Vickrey-Clark-Groves (VCG) auctions; however, such NP-hardness makes the direct utilization of VCG impossible, as it requires the social welfare maximization in one round to be solved in polynomial time [3].

To overcome such challenges and design a computationally efficient mechanism, we divide the social welfare into two parts: 1) the switching cost of $C^t_{SC} = \sum_k \alpha_k [x^t_k - x^{t-1}_k]^+$, the only term that is coupled over time, depending on the past cloudlet activation status $\boldsymbol{x}^{t-1}$ and the current status $\boldsymbol{x}^t$; 2) the non-switching welfare of $W^t_{-SC} = \sum_i \sum_j \sum_k a^t_{ijk} y^t_{jik} - \sum_i \sum_j f_{ij}(z^t_{ij}) - pu^t_g - \varsigma \sum_k x^t_k$ which can be obtained at each time slot t independently if the activation status $\boldsymbol{x}^t$ is given. Leveraging the separation of social welfare, we first study the workload allocation and winner determination problem at each round with given cloudlet activation status $\boldsymbol{x}^t$. A primal-dual-based approximation algorithm for the one-shot problem is proposed in Sec. III. Based on the one-shot solution, in Sec. IV, we further present an online algorithm that decomposes the long-term social welfare maximization problem into a series of one-round problems. Both algorithms are in polynomial time.

III. Algorithm for One-Round Auction

A. Primal-Dual-Based Algorithm Design

In this section, we focus on the one-shot problem at a single time slot based on the assumption that the cloudlet activation status $\boldsymbol{x}^t$ is given. As obtaining the optimal non-switching welfare W^t_{-SC} is NP-hard, instead of pursuing an optimal solution, we propose a primal-dual-based algorithm to obtain an efficient approximate solution within polynomial time.

For simplicity of the presentation, in the following, we omit the time index t of all the parameters and the variables. We reformulate the problem as follows:

$$\max \quad \sum_i \sum_j \sum_k a_{ijk} y_{jik} - \sum_i \sum_j f_{ij}(z_{ij}) - g(u) \tag{2}$$

$$\text{s.t.} \quad \sum_i \sum_j \lambda_{ij} y_{jik} \leqslant R_k x_k, \forall k \in \mathcal{N} \tag{2a}$$

$$\sum_i \sum_j \sum_k \beta_k \lambda_{ij} y_{jik} \leqslant u \tag{2b}$$

$$\sum_k d_{ijk} y_{jik} \leqslant z_{ij}, \forall i \in \mathcal{M}, \forall j \in \mathcal{M} \tag{2c}$$

$$\sum_k y_{jik} \leqslant 1, \forall i \in \mathcal{S}, \forall j \in \mathcal{M} \tag{2d}$$

$$y_{jik} \in \{0,1\}, \forall i \in \mathcal{S}, \forall j \in \mathcal{M}, \forall k \in \mathcal{N} \tag{2e}$$

$$u, z_{ij} \geqslant 0, \forall i \in \mathcal{S}, \forall j \in \mathcal{M} \tag{2f}$$

Note that since the cloudlet activation status $\boldsymbol{x}$ is given, the maintenance cost $\varsigma \sum_k x_k$ is a constant and thus omitted. We also reformulate the local generation cost:

$$g(u) = \begin{cases} 0, & u \leqslant P'_{EDR} \\ p(u - P'_{EDR}), & u \geqslant P'_{EDR} \end{cases},$$

and replace (1b) by (2b), where $P'_{EDR} = P_{EDR} - \sum_k \gamma_k x_k$ and u is the active power caused by workload allocation. When $u \leqslant P'_{EDR}$, i.e., the active power demand is no larger than the EDR cap minus idle power consumption, the local generation cost is zero; otherwise, when $u \geqslant P'_{EDR}$, the extra power has to be met by the local generation.

By relaxing the binary variables y_{jik} to the continuous non-negative variables and introducing the dual variables μ_k, φ, ξ_{ij}, and ρ_{ij} for the constraints (2a)-(2d), respectively, we obtain the dual problem [20] of the relaxed problem (2):

$$\min \quad \sum_k R_k x_k \mu_k + \sum_i \sum_j \rho_{ij} + \sum_i \sum_j f^*_{ij}(\xi_{ij}) + g^*(\varphi) \tag{3}$$

$$\text{s.t.} \quad \rho_{ij} \geqslant a_{ijk} - (\lambda_{ij}\mu_k + d_{ijk}\xi_{ij} + \beta_k \lambda_{ij}\varphi), \quad \forall i \in \mathcal{S}, j \in \mathcal{M}, k \in \mathcal{N} \tag{3a}$$

$$\mu_k, \varphi, \xi_{ij}, \rho_{ij} \geqslant 0 \tag{3b}$$

where $f^*_{ij}(\xi_{ij})$ and $g^*(\varphi)$ are the Fenchel conjugates [20], [21] of the functions $f_{ij}(z_{ij})$ and $g(u)$, respectively:

$$f^*_{ij}(\xi_{ij}) = \sup_{z_{ij}\geqslant 0}\{z_{ij}\xi_{ij} - f_{ij}(z_{ij})\} = \begin{cases}\theta_{ij}(\xi_{ij} - c_{ij}), & \xi^t_{ij} \geqslant c_{ij} \\ 0, & \xi^t_{ij} \leqslant c_{ij}\end{cases},$$

$$g^*(\varphi) = \sup_{u\geqslant 0}\{u\varphi - g(u)\} = \begin{cases}\varphi P'_{EDR}, & u \leqslant P'_{EDR} \\ pP'_{EDR}, & u \geqslant P'_{EDR} \text{ and } \varphi \leqslant p \\ +\infty, & u \geqslant P'_{EDR} \text{ and } \varphi > p\end{cases}.$$

Following the idea of primal-dual optimization, y_{jik} remains zero unless its corresponding dual constraint (3a) becomes tight. We let ρ_{ij} be the greater quantity between 0 and the right hand side of constraint (3a):

$$\rho_{ij} = \max\{0, \max_k\{a_{ijk} - (\lambda_{ij}\mu_k + d_{ijk}\xi_{ij} + \beta_k\lambda_{ij}\varphi)\}\}. \quad (4)$$

When $\rho_{ij} > 0$, the operator allocates the workload λ_{ij} to the cloudlet k which maximizes the right hand side of (3a), i.e., $k = \arg\max\{a_{ijk} - (\lambda_{ij}\mu_k + d_{ijk}\xi_{ij} + \beta_k\lambda_{ij}\varphi)\}$; otherwise, the workload λ_{ij} will not be allocated, which, however, will not actually happen as described below.

The principle of the proposed approximation algorithm is as follows: If we interpret the dual variable μ_k as the per-unit resource price of cloudlet k, ξ_{ij} as the per-unit delay penalty, and φ as the per-unit local generation cost, then the term $\lambda_{ij}\mu_k + d_{ijk}\xi_{ij} + \beta_k\lambda_{ij}\varphi$ can be regarded as the total cost caused by allocating workload λ_{ij} to cloudlet k, and the right hand side of (3a) can be regarded as the social utility of this allocation. As defined in Sec. II, c_{ij} is the per-unit penalty when the delay exceeds the delay tolerance D_{ij} and p is the fuel cost of local generation, then we let $\xi_{ij} = c_{ij}$ and $\varphi = p$. Given that as cloudlets are heterogenous, the resource price μ may vary from one cloudlet to another. μ_k can be set following historical processing statistics, and it should be bounded by $\mu_k < \min_{i,j}\{(a_{ijk} - d_{ijk}c_{ij})/\lambda_{ij} - \beta_k p\}$ to guarantee that every unit of end user workload can be processed. Based on this idea, the workload allocation and winner determination algorithm ensures that the workload λ_{ij} is processed with maximum social utility and hence guarantees the social welfare maximization.

Algorithm 1 illustrates the one-round auction framework. In Line 1, O is the set of winners, Q is the set of unallocated workload, ω_k presents the resource utilization of cloudlet k, $\mathcal{N}_{ij}$ indicates the candidate cloudlets where workload λ_{ij} can be allocated, and $\rho_{ijk} = a_{ijk} - (\lambda_{ij}\mu_k + d_{ijk}\xi_{ij} + \beta_k\lambda_{ij}\varphi)$. Line 3 and Lines 5-6 determine the workload allocation y_{jik} and winner's delay penalty $d_{ijk}c_{ij}$ based on the above rationale. After that, Lines 3-4 and Lines 7-9 calculate the payment to the winner, whose effectiveness is proved in the later Theorem 3. And then, Lines 10-13 update the cloudlet k^*'s resource utilization ω_{k^*}, the active power consumption u, the workload allocation index (i^*, j^*, k^*), and the set of unallocated workload Q, respectively.

B. Performance Analysis

Theorem 1. *Algorithm 1 generates a feasible solution to both the problem (2) and the problem (3) in polynomial time.*

Algorithm 1 The workload allocation and winner determination framework

```
1:  Initialize: $z_{ij} = 0$, $y_{jik} = 0$, $r_{ij} = 0$, and $u = 0$; $O = \emptyset$,
    $Q = \{(i,j)|i \in \mathcal{S}, j \in \mathcal{M}\}$, $\mathcal{N}_{ij} = \{k|d_{ijk} \leqslant \theta_{ij}\}$,
    $\omega_k = 0$, $\xi_{ij} = c_{ij}$, $\varphi = p$, $\forall i \in \mathcal{S}$, $\forall j \in \mathcal{M}$, $\forall k \in \mathcal{N}$;
2:  while $Q \neq \emptyset$ do
3:      $(i^*, j^*, k^*) = \arg\max\{\rho_{ijk}|\omega_k + \lambda_{ij} \leqslant R_k x_k, (i,j) \in Q, k \in \mathcal{N}_{ij}\}$;
4:      $(i^-, j^-, k^-) = \arg\max\{\rho_{ijk}|\omega_k+\lambda_{ij} \leqslant R_k x_k, (i,j) \in Q, k \in \mathcal{N}_{ij} \setminus \{k^* \in \mathcal{N}_{i^*j^*}\}\}$;
5:      $y_{j^*i^*k^*} = 1$;
6:      $z_{i^*j^*} = d_{i^*j^*k^*}$;
7:      if $z_{i^*j^*} > 0$ do
8:          $r_{i^*j^*} = c_{i^*j^*}d_{i^*j^*k^*} + \rho_{i^*j^*k^*} - \rho_{i^-j^-z^-}$;
9:      end if
10:     $\omega_{k^*} = \omega_{k^*} + \lambda_{i^*j^*}$;
11:     $u = u + \beta_{k^*}\lambda_{i^*j^*}$
12:     $O = O \cup (i^*, j^*, k^*)$;
13:     $Q = Q \setminus (i^*, j^*)$;
    end while
14: $W_{-SC} = \sum_i \sum_j \sum_k a_{ijk}y_{jik} - \sum_i \sum_j f_{ij}(z_{ij}) - g(u) - \varsigma \sum_k x_k$;
```

Theorem 2. *Algorithm 1 is a $\frac{\sigma}{\sigma-1}$-approximation algorithm to the problem (2), i.e., the social welfare obtained by Algorithm 1 is at least $\frac{\sigma-1}{\sigma}$ times the optimal social welfare in the problem (2), where $\frac{1}{\sigma} = \frac{\sum_k R_k x_k \mu_k + pP'_{EDR}}{\sum_k R_k x_k \mu_k + pP'_{EDR} + \sum_i \sum_j \rho_{min}}$ and $\rho_{min} = \min_{i,j,k}\{a_{ijk} - (\lambda_{ij}\mu_k + d_{ijk}\xi_{ij} + \beta_k\lambda_{ij}\varphi)|\lambda_{ij} \leqslant R_k x_k, d_{ijk} \leqslant \theta_{ij}, i \in \mathcal{S}, j \in \mathcal{M}, k \in \mathcal{N}\}$.*

Theorem 3. *The proposed procurement auction mechanism achieves individual rationality and truthfulness.*

Due to the page limit, we provide the proofs of Theorems 1, 2, and 3 in the Appendix of our technical report [22].

IV. Online Algorithm for Long-Term Problem

A. Online Algorithm Design

In this section, we present an online algorithm to determine the cloudlet activation state to achieve the long-term social welfare maximization, based on the one-shot solution from Algorithm 1 at each time slot.

Due to the lack of a priori knowledge, the operator should be careful about the control of cloudlets at each time slot. One solution is to pursue only the long-term optimum non-switching welfare by obtaining the maximum non-switching value among all possible cloudlet statuses at each time slot. However, this solution is hard to accomplish, and may result in aggressive switching of cloudlets and thus hurt the long-term social welfare. First, there are totally 2^N possible cloudlet statuses combinations in the edge system consisting of N heterogenous cloudlets, which makes obtaining the optimum solution at one time slot computationally prohibitive. Moreover, even if one has the optimal solution, not considering the switching cost can lead to the result that, for example,

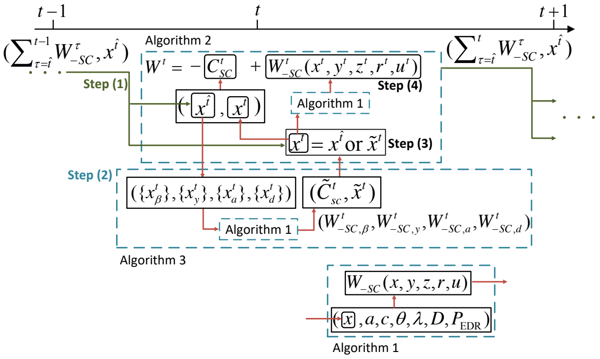
$t-1$
t
$t+1$
$(\sum_{\tau=\hat{t}}^{t-1} W_{-SC}^{\tau}, x^{\hat{t}})$
Algorithm 2
$(\sum_{\tau=\hat{t}}^{t} W_{-SC}^{\tau}, x^{\hat{t}})$
$W^t = -\boxed{C_{SC}^t} + \boxed{W_{-SC}^t(x^t, y^t, z^t, r^t, u^t)}$
Step (1)
Step (4)
$(\boxed{x^{\hat{t}}}, \boxed{x^t})$
Algorithm 1
$\boxed{x^t} = x^{\hat{t}}$ or $\tilde{x}^t$
Step (3)
Step (2)
$(\{x_\beta^t\}, \{x_y^t\}, \{x_a^t\}, \{x_d^t\})$
$(\tilde{C}_{sc}^t, \tilde{x}^t)$
Algorithm 1
$(W_{-SC,\beta}^t, W_{-SC,y}^t, W_{-SC,a}^t, W_{-SC,d}^t)$
Algorithm 3
$W_{-SC}(x, y, z, r, u)$
$(\boxed{x}, a, c, \theta, \lambda, D, P_{\text{EDR}})$
Algorithm 1

Fig. 3. An illustration of our online framework.

the local optimal solution suggests to shut down and then activate a cloudlet, while better long-term solutions may let this cloudlet remain activated to save the switching cost over time.

We propose an online algorithm for obtaining cost-efficient solutions while avoiding aggressive cloudlet switching, as shown in Algorithm 2 with a component shown in Algorithm 3. The principle is as follows. In Algorithm 3, we obtain a near-optimal cloudlet activation status. And then in Algorithm 2, we postpone the cloudlet switching required by the near-optimal solution, until the surplus (i.e., the cumulative non-switching welfare) exceeds the switching cost obviously or until there is no feasible solution with the unchanged activation state. Given that Algorithm 1, as a component of Algorithm 2 and Algorithm 3, is used to obtain the workload allocation and winner determination with corresponding non-switching welfare when the cloudlet status is given.

Fig. 3 shows the illustration of the proposed online framework. We first describe how to obtain a near-optimal solution in Algorithm 3. Assume the last switching time slot before the current t is $\hat{t}$. Based on the formulation (2), given the previous cloudlet activation state $\boldsymbol{x}^{t-1} = \boldsymbol{x}^{\hat{t}}$ (step (1) in Fig. 3), there are four possible variations in cloudlet status $\boldsymbol{x}^t$ by which we may achieve a higher non-switching welfare W_{-SC}^t: 1) Closing cloudlets with lower power efficiency to reduce active power of allocating workload and total static power of cloudlets; 2) Closing cloudlets with less allocated workload to cut down static power while ensure limited utility decrease; 3) Opening cloudlets with higher "reliability" to increase the utility of allocated workload; 4) Opening cloudlets with less propagation delay to other connected cloudlets to reduce the possible delay penalty. As indicated by step (2) in Fig. 3 and Lines 1 to 5 in Algorithm 3, we compare the non-switching welfare obtained by above variations in cloudlet status and choose the status with highest non-switching welfare, denoted by $\tilde{\boldsymbol{x}}^t$, as the near-optimal solution. And the switching cost caused by cloudlet switching is $\tilde{C}_{SC}^t(\tilde{\boldsymbol{x}}^t, \boldsymbol{x}^{\hat{t}})$.

In Algorithm 2, at time slot t, we compare the cumulative non-switching welfare from $\hat{t}$ to $t-1$, $\sum_{\tau=\hat{t}}^{t-1} W_{-SC}^{\tau}$, with the switching cost $\tilde{C}_{SC}^t(\tilde{\boldsymbol{x}}^t, \boldsymbol{x}^{\hat{t}})$ obtained by Algorithm 3: if $\sum_{\tau=\hat{t}}^{t-1} W_{-SC}^{\tau}$ exceeds κ times $\tilde{C}_{SC}^t(\tilde{\boldsymbol{x}}^t, \boldsymbol{x}^{\hat{t}})$ or there is no feasible solution with the status $\boldsymbol{x}^{\hat{t}}$, we will vary the cloudlet activation status from $\boldsymbol{x}^{\hat{t}}$ to $\tilde{\boldsymbol{x}}^t$; otherwise, the activation state is unchanged at time slot t. After deciding the cloudlet status $\boldsymbol{x}^t$, the corresponding non-switching welfare W_{-SC}^t as well as workload allocation and winner determination can be obtained by Algorithm 1. The above process is indicated by step (3) and (4) in Fig. 3 and Lines 6 to 15 in Algorithm 2.

Algorithm 2 The Online Algorithm Framework

```
1:  Define: $t = 1$, $\hat{t} = 0$, $\Delta W = 0$;
2:  while $t \leqslant T$ do
3:    Service providers submit requests $\{(\lambda_{ij}^t, D_{ij}^t) | i \in \mathcal{S},\ j \in \mathcal{M}\}$ bids $\{(\theta_{ij}^t, c_{ij}^t) | i \in \mathcal{S},\ j \in \mathcal{M}\}$;
4:    Obtain $W_{-SC}^t(\boldsymbol{x}^{\hat{t}}, \boldsymbol{y}^t, \boldsymbol{z}^t, \boldsymbol{r}^t, u^t)$ by Algorithm 1;
5:    Obtain $\tilde{C}_{SC}^t(\tilde{\boldsymbol{x}}^t, \boldsymbol{x}^{\hat{t}})$ and $\tilde{W}_{-SC}^t(\tilde{\boldsymbol{x}}^t, \tilde{\boldsymbol{y}}^t, \tilde{\boldsymbol{z}}^t, \tilde{\boldsymbol{r}}^t, \tilde{u}^t)$ by Algorithm 3;
6:    if $\Delta W \geqslant \kappa \tilde{C}_{SC}^t(\tilde{\boldsymbol{x}}^t, \boldsymbol{x}^{\hat{t}})$ or $W_{-SC}^t(\boldsymbol{x}^{\hat{t}}, \boldsymbol{y}^t, \boldsymbol{z}^t, \boldsymbol{r}^t, u^t) = -\infty$ then
7:      $\boldsymbol{x}^t = \tilde{\boldsymbol{x}}^t$;
8:      $W_{-SC}^t(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{z}^t, \boldsymbol{r}^t, u^t) = \tilde{W}_{-SC}^t(\tilde{\boldsymbol{x}}^t, \tilde{\boldsymbol{y}}^t, \tilde{\boldsymbol{z}}^t, \tilde{\boldsymbol{r}}^t, \tilde{u}^t)$;
9:      $\Delta W = W_{-SC}^t(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{z}^t, \boldsymbol{r}^t, u^t)$;
10:     $\hat{t} = t$;
11:   else
12:     $\boldsymbol{x}^t = \boldsymbol{x}^{\hat{t}}$;
13:     $W_{-SC}^t(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{z}^t, \boldsymbol{r}^t, u^t) = W_{-SC}^t(\boldsymbol{x}^{\hat{t}}, \boldsymbol{y}^t, \boldsymbol{y}^t, \boldsymbol{r}^t, u^t)$;
14:     $\Delta W = \Delta W + W_{-SC}^t(\boldsymbol{x}^t, \boldsymbol{y}^t, \boldsymbol{z}^t, \boldsymbol{r}^t, u^t)$;
15:   end if
16:   The operator notifies the winning bids $\boldsymbol{z}^t$ and payments $\boldsymbol{r}^t$, allocates workload according to $\boldsymbol{y}^t$, and sets the amount of local generation according to $u^t$;
17:   $t = t + 1$;
18: end while
```

Algorithm 3 Obtain $\tilde{W}_{-SC}^t(\tilde{\boldsymbol{x}}^t, \tilde{\boldsymbol{y}}^t, \tilde{\boldsymbol{z}}^t, \tilde{u}^t)$ and $\tilde{C}_{SC}^t(\tilde{\boldsymbol{x}}^t, \boldsymbol{x}^t)$

```
1: Close the active cloudlet in $\boldsymbol{x}^{\hat{t}}$ one by one according to the descending order of $\beta_k$, calculate the non-switching welfare by Algorithm 1, and obtain the maximum $W_{-SC,\beta}^t(\boldsymbol{x}_\beta^t, \boldsymbol{y}_\beta^t, \boldsymbol{z}_\beta^t, \boldsymbol{r}_\beta^t, u_\beta^t)$;
2: Close the active cloudlet in $\boldsymbol{x}^{\hat{t}}$ one by one according to the ascending order of $\sum_i \sum_j y_{jik}^{t-1}$, calculate the non-switching welfare by Algorithm 1, and obtain the maximum $W_{-SC,y}^t(\boldsymbol{x}_y^t, \boldsymbol{y}_y^t, \boldsymbol{z}_y^t, \boldsymbol{r}_y^t, u_y^t)$;
3: Open the closed cloudlet in $\boldsymbol{x}^{\hat{t}}$ one by one according to the descending order of $\sum_i \sum_j a_{ijk}^t$, calculate the non-switching welfare by Algorithm 1, and obtain the maximum $W_{-SC,a}^t(\boldsymbol{x}_a^t, \boldsymbol{y}_a^t, \boldsymbol{z}_a^t, \boldsymbol{r}_a^t, u_a^t)$;
4: Open the closed cloudlet in $\boldsymbol{x}^{\hat{t}}$ one by one according to the ascending order of $\sum_i \sum_j d_{ijk}$, calculate the non-switching welfare by Algorithm 1, and obtain the maximum $W_{-SC,d}^t(\boldsymbol{x}_d^t, \boldsymbol{y}_d^t, \boldsymbol{z}_d^t, \boldsymbol{r}_d^t, u_d^t)$;
5: Select the maximum from the above four non-switching welfare, which is indicated by $\tilde{W}_{-SC}^t(\tilde{\boldsymbol{x}}^t, \tilde{\boldsymbol{y}}^t, \tilde{\boldsymbol{z}}^t, \tilde{\boldsymbol{r}}^t, \tilde{u}^t)$;
6: $\tilde{C}_{SC}^t(\tilde{\boldsymbol{x}}^t, \boldsymbol{x}^t) = \sum_k \alpha_k [\tilde{x}_k^t - x_k^{\hat{t}}]$;
```

B. Performance Analysis

We first provide a sketched proof to the truthfulness of EdgeEDR. As given by Theorem 3, the proposed one-round procurement auction mechanism is truthful with a given cloudlet activation status. The status is determined based on bids in Algorithm 2 and is only known to the cloudlet operator, which makes EdgeEDR with the one-round mechanism truthful [18].

The time complexities of Algorithm 2 and Algorithm 3 are analyzed as follows. For Algorithm 3, each of the Lines 1 to 4 runs at most N times. Given that Algorithm 1's time complexity is $O(S^2M^2N)$, the complexity of Algorithm 3 is $O(S^2M^2N^2)$. For Algorithm 2, the `while` loop in Line 2 runs T times, and the complexities of Lines 4 and 5 are $O(S^2M^2N)$ and $O(S^2M^2N^2)$, respectively. In summary, Algorithm 2's time complexity is $O(TS^2M^2N^2)$.

Theorem 4. *The online algorithm gives a* $\frac{\kappa}{(\kappa-1)\epsilon}\cdot(\frac{\sigma}{\sigma-1})^2$*-competitive solution to the social welfare maximization problem, i.e., the social welfare obtained by Algorithm 2 is at least* $\frac{(\kappa-1)\epsilon}{\kappa}\cdot(\frac{\sigma-1}{\sigma})^2$ *times the offline optimal social welfare, where* $\epsilon=\min_{t\in\mathcal{T}}\frac{\min_{\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t} W^t_{-SC}(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t)}{\max_{\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t} W^t_{-SC}(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t)}$*, and* $\frac{\sigma}{\sigma-1}$ *is the approximation ratio of Algorithm 1.*

Proof. In Algorithm 2, Line 5 guarantees that the switching cost at time slot t is at most $\frac{1}{\kappa}$ times the non-switching welfare incurred within time frame $[\hat{t},t-1]$, where $\hat{t}$ is the last time slot of cloudlet activation state switching before t. In the worst case, over the whole time frame $\mathcal{T}$, the cloudlet activation status $\boldsymbol{x}$ switches in each time frame, we have $\sum_{\tau=1}^{T}C^{\tau}_{SC}\leqslant\frac{1}{\kappa}\sum_{\tau=1}^{T}W^{\tau}_{-SC}$. Hence we have$\sum_{\tau=1}^{T}W^{\tau}=\sum_{\tau=1}^{T}W^{\tau}_{-SC}-\sum_{\tau=1}^{T}C^{\tau}_{SC}\geqslant(1-\frac{1}{\kappa})\sum_{\tau=1}^{T}W^{\tau}_{-SC}$. Let $\bar{W}^t$ denote the offline optimal social welfare at time slot t, and ϵ be the minimum ratio of the smallest non-switching welfare to the largest non-switching welfare at each time slot, i.e., $\epsilon=\min_{t\in\mathcal{T}}\frac{\min_{\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t} W^t_{-SC}(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t)}{\max_{\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t} W^t_{-SC}(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t)}$. Note that W^t_{-SC} given by Algorithm 1 is at least $\frac{\sigma-1}{\sigma}$ times the optimal non-switching welfare W^{*t}_{-SC}. Hence $\epsilon^*=\min_{t\in\mathcal{T}}\frac{\min_{\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t} W^{*t}_{-SC}(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t)}{\max_{\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t} W^{*t}_{-SC}(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t)}\leqslant\frac{\sigma}{\sigma-1}\epsilon$. And we have

$$\sum_{\tau=1}^{T}W^{\tau}\geqslant(1-\frac{1}{\kappa})\sum_{\tau=1}^{T}W^{\tau}_{-SC}\geqslant(\frac{\sigma-1}{\sigma})^2\epsilon(1-\frac{1}{\kappa})\sum_{\tau=1}^{T}\bar{W}^{\tau}_{-SC}$$

$$\geqslant(\frac{\sigma-1}{\sigma})^2\epsilon(1-\frac{1}{\kappa})\sum_{\tau=1}^{T}\bar{W}^{\tau}. \qquad \square$$

As obtaining the exact maximum and minimum non-switching welfare is computationally prohibitive, here we propose a rough estimate of the competitive ratio. According to the formulation of non-switching welfare, $W^t_{-SC}(\boldsymbol{x}^t,\boldsymbol{y}^t,\boldsymbol{z}^t,u^t)$ is bounded by $[\sum_i\sum_j\min_k\{a^t_{ijk}\}-p(P^t_{max}-P^t_{EDR})-\sum_i\sum_j c^t_{ij}\theta^t_{ij},\sum_i\sum_j\max_k\{a^t_{ijk}\}]$, where P^t_{max} is the power consumption of the case that all cloudlets are active and the workload are allocated to those most power-consuming cloudlets. And hence the competitive ratio is bounded by $\big[1,\frac{\kappa\sigma^2}{(\kappa-1)(\sigma-1)^2}\min_{t\in\mathcal{T}}\big\{\big(\sum_i\sum_j\min_k\{a^t_{ijk}\}-p(P^t_{max}-P^t_{EDR})-\sum_i\sum_j c^t_{ij}\theta^t_{ij}\big)/\sum_i\sum_j\max_k\{a^t_{ijk}\}\big\}\big]$. In the following section, the simulation results in Fig. 6 show an efficient empirical competitive ratio of our solution.

V. Experimental Evaluation

A. Experiment Setup

We simulate an edge system where the cloudlet operator owns 40 distributed cloudlets serving 80 service providers and each cloudlet is attached to an AP. We assume the cloudlets are located at London's 40 underground stations with heavy passenger traffic, and the geographical distance between any two stations is used to approximate the propagation delay between the attached cloudlets [13]. We assume servers are all homogeneous; each server can process 25 requests at one time slot. We divide the sum of entire edge system's peak workload by the number of cloudlets, and randomly scale it from $0.8\times$ to $1.2\times$ to generate cloudlets' capacity. The cloudlet's PUE ranges from 1.3 to 2 randomly, and the idle and peak power of a server is 100 W and 300 W, respectively. The diesel price for local generation is set to 0.8 \$/kWh [23], [24].

We utilize the dynamic passenger numbers at a station to represent the total amount of workload submitted from that station (i.e., AP), and the amount of request for each service is assigned randomly. The reliability of workload allocation is set to be one order of magnitude larger than the per-unit delay penalty. And the resource price of each cloudlet is generated referring to Sec. III-A. The delay tolerance of request is set according to the mean propagation delay between cloudlets. And the delay degradation tolerance submitted by service provider follows a uniform distribution between 60% delay tolerance to 100% delay tolerance. We vary the weight of the switching cost, κ in online framework, and the per-unit delay penalty to obtain a spectrum of results, hence we do not give the concrete metric here.

According to a real EDR event from the service region of PJM on Jan. 4, 2014 [25], we set the length of one time slot to 15 minutes and the total length of EDR event is 28 time slots. The EDR signal requires the edge system to reduce 25% of the edge system's peak IT power consumption, which is reported as a reasonable setting in EDR event without significant impact to participant's operation [5], [26].

To validate EdgeEDR's effectiveness, we compare its performance with following benchmark algorithms: 1) The online no auction, for which the operator does not motivate service providers to reduce QoS requirement; 2) The online greedy and auction, for which the operator directly varies the cloudlet status when the near-optimal social welfare is larger than that in previous status. We also compare EdgeEDR with the optimum algorithm where the one-shot problem is exactly obtained by CVX [27] with Gurobi [28]: 1) The online MIP optimum. 2) The offline optimum, for which we search all of possible cloudlet activation statuses at each time slot to obtain the optimum solution. As the optimum has to be obtained with worse time complexity, we only conduct the optimum framework simulation in small-scale scenario. For simplicity, all the values (except for results in Fig. 9) are normalized with respect to it achieved by EdgeEDR.

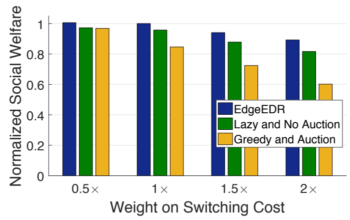

Fig. 4. Comparison of the normalized social welfare with different EDR frameworks as the weight on switching cost increases.

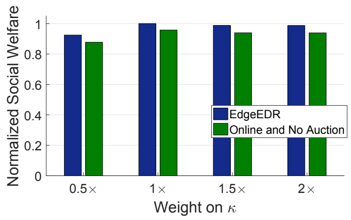

Fig. 5. Comparison of the normalized social welfare with online no auction framework as the weight on κ increases.

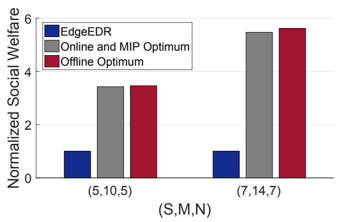

Fig. 6. Comparison of the normalized social welfare with online MIP optimum and offline optimum in small-scale simulation.

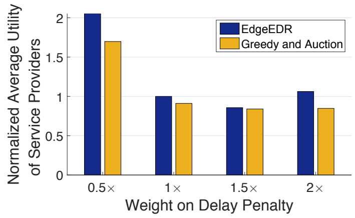

Fig. 7. Comparison of the service provider's normalized average utility as the weight on per-unit delay penalty increases.

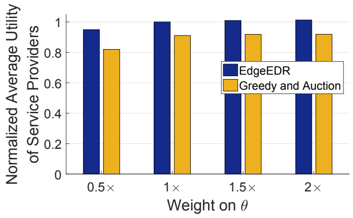

Fig. 8. Comparison of the service provider's normalized average utility as the weight on delay degradation tolerance increases.

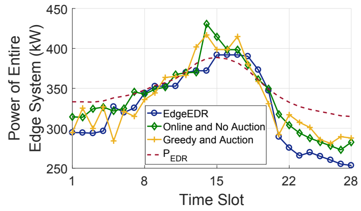

Fig. 9. Comparison of the entire edge system power within EDR event with different EDR frameworks.

B. Evaluation Results

Fig. 4 compares the normalized social welfare obtained by EdgeEDR and that obtained by two online benchmarks, as the weight on the switching cost increases. First, compared to the online no auction benchmark where the operator joins to EDR only by workload allocation and local generation, we find that the auction-based EdgeEDR has a better efficiency in improving the social welfare of both the cloudlet operator and tenants in EDR events. Compared to the online greedy framework, as EdgeEDR effectively avoids aggressive cloudlet statuses switching and corresponding high switching cost, it achieves a great effectiveness in improving long-term social welfare even if the future information is unknown. When the switching cost increases, the excessive decrease in long-term social welfare can also be avoided by EdgeEDR.

Fig. 5 compares the normalized social welfare achieved by EdgeEDR with that achieved by the online no auction benchmark, as the weight on κ increases. κ is used to compare the value of cumulative non-switching welfare and switching cost, and determine whether the cloudlet activation state should be switched or not at each time slot. The results indicate that in EdgeEDR, κ should be set carefully. A higher κ hurts the social welfare since it may delay the necessary switch of cloudlet activation state and cause lower non-switching welfare; while frequent activation state switching caused by a lower κ may bring in excessive switching cost.

Fig. 6 illustrates the normalized social welfare obtained by EdgeEDR, the online MIP benchmark, and the offline optimum, respectively, in a small-scale scenario. We find that the empirical competitive ratio of EdgeEDR is within $3.5\sim5.6$. When the one-shot problem at each time slot can be solved exactly, EdgeEDR shows a great empirical performance, which indicates that the efficiency of primal-dual-based algorithm for workload allocation and winner determination is the main reason for the performance gap between EdgeEDR and the offline optimum in the small-scale scenario.

Fig. 7 and Fig. 8 present the normalized average utility of service providers, i.e., the winning service provider's payment minuses the delay penalty as the per-unit delay penalty cost c_{ij} increases and as the delay degradation tolerance θ_{ij} increases, respectively. We find in Fig. 7 that in general, increasing per-unit delay penalty leads to a decline in tenant's utility. This is because high per-unit delay penalty decreases the social utility of workload allocation ρ_{ij}. Fig. 8 indicates that, when relaxing the delay degradation tolerance, there are more candidate cloudlets for allocating workload with QoS guarantee and the social utility achieved by workload allocation can be larger, and hence the relaxation of delay degradation tolerance improves the tenant's utility.

Fig. 9 shows the power of the entire edge system within EDR event. We find that in the computation low time, EdgeEDR and other benchmarks all perform well in power shedding; while in the computation peak period, EdgeEDR achieves the best power efficiency. EdgeEDR saves about 54.1% to 57.2% local generated power, indicating that cloudlets' participation in the EDR programs with the auction-based EdgeEDR ensures the "greenness" compared to other EDR mechanisms.

VI. Related Work

There exist many studies on improving the power efficiency and achieving power cost saving in datacenters, such as workload management [29]–[31] and datacenter heat harvesting [32]. In this section, we summarize very closely-related existing work only. Ren et. al. [6] is among the first to study the colocation data center (referred to as "colo" henceforth) demand response scenario, where they propose a simple reverse auction to meet the energy reduction requirement. Zhang et. al. [3] also design a reverse auction, introducing local generators to help meet the energy reduction target and VCG-based machanisms to guarantee truthful bidding. Chen et. al. [9] design a pricing mechanism based on supply function biddings to extract load reductions from tenants in demand response periods. Sun et. al. [4], [5] investigate two cases of demand response in geo-distributed colos with deferrable batch jobs, and propose online multi-round auctions while determining when to execute each job. Tran et. al. [8] and Islam et. al. [7] may be among the few that do not use auctions but reward-based mechanisms to issue rewards in exchange for tenants' energy reductions.

Our research in this paper differs from all the above existing work in multiple aspects. First, we do "online" reverse auctions for multiple demand response frames. All existing research, except [4], [5], [7], uses a one-round, static auction which does not explore the temporal connection between different demand response frames. Second, we manage (i.e., distribute) tenants' workload explicitly and adjust service quality based on the operator's needs. Previous mechanisms, except [4], [5], do not manage tenants' workload and are limited by tenants' fixed offers. Third, we are the first to explore the lever of switching on/off entire cloudlets, while considering the switching cost over time, to compensate for the energy reduction requirement. This is particularly a feature that may be possible for cloudlets only. Existing research mainly target large-scale clouds and data centers [33] that cannot be usually switched off entirely.

VII. Conclusion

We study the emergency demand response for distributed cloudlets. While designing a series of procurement auctions to address the operator-tenant split incentives, in our mechanism, we focus on enabling the operator to directly distribute tenants' workload across cloudlets in order to gain more flexibility in the procurement of tenants' bids and to adapt to the changing market conditions. We are also the first to propose to let the operator switch on/off entire cloudlets to compensate for the energy reduction requirement while dynamically striking the balance between the energy saving benefit and the incurred switching cost. We propose an online algorithm, which adopts a polynomial-time approximation algorithm in each auction, with provable performance guarantees and validated practical superiority compared to existing methods.

References

[1] M. Satyanarayanan, "The emergence of edge computing," *IEEE Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[2] W. Shi and S. Dustdar, "The promise of edge computing," *IEEE Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[3] L. Zhang, S. Ren, C. Wu, and Z. Li, "A truthful incentive mechanism for emergency demand response in colocation data centers," in *IEEE INFOCOM*, 2015.

[4] Q. Sun, S. Ren, C. Wu, and Z. Li, "An online incentive mechanism for emergency demand response in geo-distributed colocation data centers," in *ACM e-Energy*, 2016.

[5] Q. Sun, C. Wu, Z. Li, and S. Ren, "Colocation demand response: Joint online mechanisms for individual utility and social welfare maximization," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3978–3992, 2016.

[6] S. Ren and M. A. Islam, "Colocation demand response: Why do I turn off my servers?" in *USENIX ICAC*, 2014.

[7] M. A. Islam, H. Mahmud, S. Ren, and X. Wang, "Paying to save: Reducing cost of colocation data center via rewards," in *IEEE HPCA*, 2015.

[8] N. H. Tran, T. Z. Oo, S. Ren, Z. Han, E.-N. Huh, and C. S. Hong, "Reward-to-reduce: An incentive mechanism for economic demand response of colocation datacenters," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3941–3953, 2016.

[9] N. Chen, X. Ren, S. Ren, and A. Wierman, "Greening multi-tenant data center demand response," *Performance Evaluation*, vol. 91, pp. 229–254, 2015.

[10] B. P. Rimal, D. P. Van, and M. Maier, "Cloudlet enhanced fiber-wireless access networks for mobile-edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3601–3618, 2017.

[11] G. Iyengar and A. Kumar, "Optimal procurement mechanisms for divisible goods with capacitated suppliers," *Review of Economic Design*, vol. 12, no. 2, pp. 129–154, 2008.

[12] N. Li, L. Chen, and M. A. Dahleh, "Demand response using linear supply function bidding," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1827–1838, 2015.

[13] L. Jiao, L. Pu, L. Wang, X. Lin, and J. Li, "Multiple granularity online control of cloudlet networks for edge computing," in *IEEE SECON*, 2018.

[14] M. Ganeshalingam, A. Shehabi, and L.-B. Desroches, "Shining a Light on Small Data Centers in the U.S." *Lawrence Berkeley National Laboratory Report*, 2017.

[15] L. Wang, L. Jiao, T. He, J. Li, and M. Mühlhäuser, "Service entity placement for social virtual reality applications in edge computing," in *IEEE INFOCOM*, 2018.

[16] L. Jiao, A. M. Tulino, J. Llorca, Y. Jin, and A. Sala, "Smoothed online resource allocation in multi-tier distributed cloud networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2556–2570, 2017.

[17] PJM. Retail electricity consumer opportunities for demand response in PJMs wholesale markets. [Online]. Available: https://www.pjm.com

[18] R. Zhou, Z. Li, and C. Wu, "A truthful online mechanism for location-aware tasks in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1737–1749, 2017.

[19] I. Alaya, C. Solnon, and K. Ghédira, "Ant algorithm for the multi-dimensional knapsack problem," in *International Conference on Bioinspired Optimization Methods and their Applications*, 2004.

[20] R. Cole, N. Devanur, V. Gkatzelis, K. Jain, T. Mai, V. V. Vazirani, and S. Yazdanbod, "Convex program duality, fisher markets, and nash social welfare," in *ACM EC*, 2017.

[21] F. Nielsen, "Legendre transformation and information geometry," 2010.

[22] S. Chen, L. Jiao, L. Wang, and F. Liu, "An online market mechanism for edge emergency demand response via cloudlet control," *Technical report*. [Online]. Available: http://grid.hust.edu.cn/fmliu/infocom19-EdgeEDR-TechnicalReport.pdf

[23] Calculator of Diesel engine power. [Online]. Available: https://power-calculation.com/generator-diesel-energy-calculator-genset.php#consumptionvalues

[24] Price of diesel fuel in London. [Online]. Available: https://www.statista.com/statistics/299547/price-of-diesel-in-london/

[25] Demand response activity on January 7-8, 2014. [Online]. Available: http://www.pjm.com

[26] G. Ghatikar, "Demand response opportunities and enabling technologies for data centers: Findings from field studies," 2012.

[27] CVX. [Online]. Available: http://cvxr.com/cvx/

[28] Gurobi. [Online]. Available: http://www.gurobi.com

[29] Z. Zhou, F. Liu, B. Li, B. Li, H. Jin, R. Zou, and Z. Liu, "Fuel cell generation in geo-distributed cloud services: A quantitative study," in *IEEE ICDCS*, 2014, pp. 52–61.

[30] F. Liu, Z. Zhou, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in saas clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2648–2658, 2014.

[31] S.-K. Jo, L. Wang, J. Kangasharju, and M. Mühlhäuser, "Green named data networking using renewable energy," in *ACM e-Energy*, 2018, pp. 414–416.

[32] S. Chen, Z. Zhou, F. Liu, Z. Li, and S. Ren, "Cloudheat: An efficient online market mechanism for datacenter heat harvesting," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 3, no. 3, p. 11, 2018.

[33] Z. Zhou, F. Liu, S. Chen, and Z. Li, "A truthful and efficient incentive mechanism for demand response in green datacenters," *IEEE Transactions on Parallel and Distributed Systems*, 2018.