Scheduling DDoS Cloud Scrubbing in ISP Networks via Randomized Online Auctions

Wencong You[1], Lei Jiao[1*], Jun Li[1], Ruiting Zhou[2]
[1]University of Oregon, USA [2]Wuhan University, China

***Abstract*—While both Internet Service Providers (ISPs) and third-party Security Service Providers (SSPs) offer Distributed Denial-of-Service (DDoS) mitigation services through cloud-based scrubbing centers, it is often beneficial for ISPs to outsource part of the traffic scrubbing to SSPs to achieve less economic cost and better network performance. To explore this potential, we design an online auction mechanism, featured by the challenge of the switching cost of using different winning bids over time. Formulating the social cost minimization as a nonconvex integer program, we firstly relax it and design an online algorithm that breaks it into a series of modified single-shot problems and solves each of them in polynomial time, without requiring knowledge of future inputs; then, we design a randomized rounding algorithm to convert the fractional decisions into integers without violating any constraints; and finally, we design the payment for each bid based on its winning probability. We rigorously prove that our mechanism achieves a parameterized-constant competitive ratio for the long-term social cost, with truthfulness and individual rationality in expectation. We also exhibit its superior practical performance via evaluations driven by real-world data traces.**

I. Introduction

Internet Service Providers (ISPs) (e.g., AT&T [1]) nowadays provide cloud-based Distributed Denial-of-Service (DDoS) mitigation services. They build and operate Scrubbing Centers (SCs) [2] and divert the suspicious traffic to such centers, where the DDoS traffic is filtered out and the clean traffic is then re-injected into the network. Meanwhile, some third-party providers (e.g., Cloudflare [3]), which we refer to as Security Service Providers (SSPs), also offer similar services through their own distributed scrubbing centers. This is shown in Fig. 1, where the solid arrows indicate the suspicious traffic and the dashed arrows indicate the clean traffic.

To scrub large-scale traffic in ISP networks, it is often beneficial for an ISP to outsource some scrubbing to external SSPs. ISPs can leverage the often wider geographical distribution of the SSPs' scrubbing centers to achieve better network footprint and overall performance [4]. For instance, scrubbing a flow closer to its source incurs a lower DDoS footprint. Moreover, through market competition, ISPs can achieve lower total economic cost for traffic scrubbing via outsourcing [5]. External scrubbing centers can be complementary to an ISP's own ones. An ISP can resort to external scrubbing centers only when the total cost is lower, while still using its own facilities to ensure that all the target flows are scrubbed.

This motivates the need of setting up an appropriate market mechanism to enable ISPs to procure scrubbing services from

*The corresponding author is Lei Jiao (e-mail: jiao@cs.uoregon.edu).

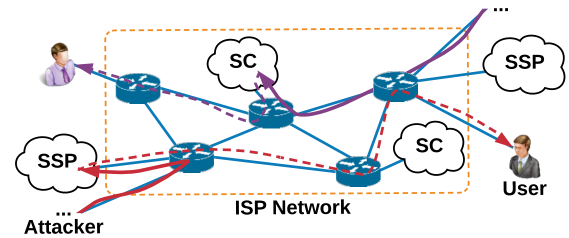

Fig. 1: An ISP network with internal SCs and external SSPs

SSPs. Letting SSPs "price" their services sounds straightforward, but can be actually tricky and hard to achieve the overall market efficiency, due to probable over-/under-pricing when the demand-supply varies (e.g., DDoS traffic dynamics, resource cost changes) [6]. Therefore, in this paper, we focus on mechanisms based on "auctions" [7]–[10]. Auctions enable market efficiency and agility with direct pricing based on the real-time supply-demand; it also reduces the chance of mispricing, better matches services to buyers that value them most, and increases the seller profit and the market social welfare.

Due to inherent market dynamics, auctions need to be conducted in multiple rounds in an online manner, where the auctioneer (i.e., the ISP) has to incur the "operation cost" to use the winning bids in each round after procuring them from the bidders (i.e., the SSPs), and incur the "switching cost" when switching from using the winning bids in one round of the auction to using possibly different winning bids in the next round. This is because, after purchasing the bids, the ISP needs to divert the traffic to the corresponding scrubbing centers to actually scrub them. Such traffic diversion is often achieved by establishing dedicated Border Gateway Protocol (BGP) and/or Open Shortest Path First (OSPF) routes in the networks [11], which occupy the router space and incur economic expense [12]; as the target traffic and the winning bids vary over time, the routes also need to be dynamically installed and removed, causing propagation traffic and convergence delay [13] that impact the network performance. While the traffic diversion may also be achieved by domain name system redirection sometimes, we consider a more general case in this paper that does not necessarily rely on the domain names.

This cost structure poses two *fundamental* and *unique* challenges to designing online auction mechanisms. First, minimizing the market social cost is hampered by the unpredictable inputs, including DDoS traffic to scrub, bids to collect from the bidders, and resources used to implement the scrubbing. It is challenging to balance the ISP's operation cost and switching cost on the fly without any future knowledge of such inputs, because determining and using the winning bids in one auction will impact the switching cost between the current auction and the next auction that is yet to come. Second, the existence of the switching cost also escalates the difficulty for designing proper payment schemes for the winning bids that ensure the desired guarantees of "truthfulness" (i.e., each bidder needs to bid its true cost in order to maximize its own utility) and "individual rationality" (i.e., each bid always brings profit to its bidder if it wins the auction). The well-known Vickrey-Clarke-Groves (VCG) mechanism that guarantees truthful bidding requires to optimally solve each auction, but doing so would indicate neglecting the switching cost across auctions, which could lead to excessive social cost in the long run. To the best of the authors' knowledge, there is no known approach which can readily overcome these two challenges simultaneously.

This paper is the first to study the cloud scrubbing market mechanism. Existing research about cloud scrubbing [4], [11], [14]–[16] has never studied the interactions between ISPs and SSPs, nor from an auction perspective. Substantial efforts have been made on online auctions for the cloud(s); however, the vast majority of them have never incorporated the auctioneer's switching cost [7]–[10], [17], not to mention the corresponding payment schemes. The only auction works known to the authors involving the switching cost [18], [19] adopt primal-dual-based algorithms while embedding payment calculations but are technically insufficient for the problem that we study in this paper. See Section VI for more discussions.

We model and formulate the online social cost optimization problem of minimizing the ISP's operation cost and switching cost, plus the SSPs' bidding cost, while ensuring every flow is scrubbed over time. We make zero assumption about the heterogeneity and the dynamism of all the inputs. Our problem turns out to be an NP-hard Nonconvex Integer Program (NIP). This is another reason for which we rule out the VCG method for payment calculation; we have to also exclude its fractional version [7] due to the existence of the switching cost.

We design a group of algorithms that work together to solve our problem in an online manner to determine the winning bids, divert the traffic flows, and calculate the payment. Firstly, given the significant challenge of the NIP, we relax our problem to its fractional version and transform it by replacing the nonconvex switching cost with carefully-designed logarithmic terms [20]. This way, we decouple the modified problem into a series of single-round convex problems which are polynomial-time-solvable in each corresponding time slot by only taking the inputs to that time slot and the solution from the previous time slot. Then, we design a randomized rounding algorithm to convert our fractional decisions into integers in each auction, which rounds two fractions to compensate each other in every iteration to violate no constraint of our problem after rounding [21], and uses the fractional solution before rounding as the winning probability of each bid. Finally, for each single-round auction, we use the winning probability of each bid to calculate a dedicated marginal cost and add it to the original bidding price to compose the payment, so that each auction aligns with the monotone allocation and finite payment rules [22] and can be provably truthful and individually rational in expectation. We also prove a constant competitive ratio for the long-term social cost as a function of the key parameters of our problem.

We conduct extensive numerical evaluations with real-world data traces. We utilize dynamic Amazon EC2 virtual machine prices [23], BGP routing cost [12], and Chicago electricity prices [24] to simulate the operation cost and the switching cost, and scrub dynamic traffic flows with a varying number of SSPs for a time horizon of 200 hours. Our approach achieves up to 35% and 32% less total cost compared to the industrial practice of using the Gurobi [25] solver and the state-of-the-art Lazy Capacity Provisioning [26] algorithm, respectively, and only incurs about 11% more total cost compared to the offline optimum. Our payment design induces truthful bidding and attains individual rationality successfully, and preserves frugality for the ISP. Our algorithm executes efficiently, and responds promptly to traffic flow variations in practice.

II. Model and Problem Formulation

A. System Modeling

ISP, SSPs, and Traffic. We consider an ISP that owns and operates a network, with a set of distributed scrubbing centers that are represented by $\mathcal{L}$. We also consider a set of SSPs, represented by $\mathcal{I}$, which offer scrubbing services via their own distributed scrubbing centers connected to this ISP's network. Each SSP may manage one or multiple scrubbing centers, which is transparent to the ISP. We study the dynamic problem over a horizon of a series of time slots $\mathcal{T} = \{1, 2, ..., |\mathcal{T}|\}$. There are a set of suspicious traffic flows to be scrubbed, represented by $\mathcal{K}$, which travel through this ISP's network. Such traffic can appear and disappear arbitrarily over time: we use a binary indicator λ_{kt} to show whether the flow $k \in \mathcal{K}$ appears in the ISP's network at the time slot $t \in \mathcal{T}$ (i.e., $\lambda_{kt} = 1$) or not (i.e., $\lambda_{kt} = 0$).

Auction Model. At every time slot t, after observing the current traffic flows, the ISP provides such information to the SSPs and solicits bids. Then, each SSP $i \in \mathcal{I}$ submits a bid to the ISP in the form of $\{c_{it}, \{f_{ikt} | \forall k \in \mathcal{K}\}\}$. The list $\{f_{ikt} | \forall k \in \mathcal{K}\}$ indicates the set of flows, where it has $f_{ikt} = 1$ if the SSP i is willing to scrub the flow k and has $f_{ikt} = 0$ otherwise, and c_{it} indicates the bidding price, i.e., the price that the SSP i wants to charge. Afterwards, the ISP decides which bids win by solving the social cost minimization problem in an online manner, and for each winning bid, calculates the payment ρ_{it} and pays it to the corresponding SSP i. We do not restrict the number of bids that can be procured, but we only allow each SSP to issue one bid; the case where each SSP issues multiple bids for potentially different set of flows with different prices can be inherently captured by our model via regarding each different bid as from a different "virtual" SSP. The auction model is shown in Fig. 2.

Decision Variables. The ISP needs to make the following binary decisions, as we study in this paper: $x_{it} \in \{1, 0\}$, which implies whether or not the ISP purchases the SSP i's bid at time t; $y_{ikt} \in \{1, 0\}$, which implies whether or not the ISP

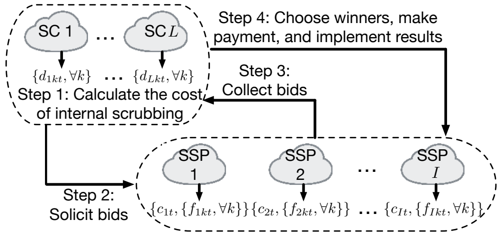

Fig. 2: Our auction mechanism in a single time slot

redirects the flow k to the corresponding scrubbing center as required by the SSP i's bid at time t; and $z_{lkt} \in \{1, 0\}$, which implies whether or not the ISP redirects the flow k to its own scrubbing center l to do the scrubbing at time t.

Cost of SSPs. The cost of the SSP i has two components at t. The first is the cost for scrubbing the flows as specified in its bid, i.e., $x_{it}c_{it}$. This may include the cost of the virtual machines, the scrubbing software, the electricity consumption and so on incurred in the SSP's scrubbing center(s). Note this cost may be different from the bidding price; however, as we will show in this paper, our auction mechanism will guarantee to be truthful, and thus they will be the same and we will use the same notation c_{it}. The second component is the payment received from the ISP, i.e., $-\rho_{it}$, where we take the negation to count it as cost. Note $\rho_{it} \geq x_{it}c_{it}$, as we will show the individual rationality of our auction mechanism.

Operation Cost of ISP. The ISP redirects each flow to one SSP scrubbing center, or to one of its own scrubbing centers, at each t. We denote the redirection cost in the ISP network for redirecting the flow k to the SSP i's scrubbing center as $a_{ikt}y_{ikt}$; accordingly, we denote the sum of the redirection cost in the ISP network for redirecting the flow k to its own scrubbing center l and the scrubbing cost for scrubbing that flow as $d_{lkt}z_{lkt}$. The redirection cost can correspond to the expense incurred by maintaining the Border Gateway Protocol (BGP) routes in the network. The scrubbing cost, as described above, can refer to multiple types of cost in the scrubbing center. Furthermore, the operation cost also includes the payment that the ISP pays, corresponding to each winning bid i, i.e., ρ_{it} (where $\rho_{it} \stackrel{\text{def}}{=} 0$ for those bids that lose).

Switching Cost of ISP. In this paper, we take into account the switching cost incurred by changing the decisions of redirecting and scrubbing the flows from one time slot to the next. This also characterizes the effect of switching from using one set of winning bids in one round of auction to using another set in the next round. Specifically, for the ISP's network, it can capture the performance impact incurred by dynamically installing/removing the BGP routes—for example, installing BGP routes takes time before the network enters a consistent status and causes the propagation traffic to spread the routes across routers; for the ISP's scrubbing centers, it also can capture the performance impact incurred by resource "reconfiguration", such as virtual machine booting/leading, software initialization, and server wear-and-tear. We use b_{ik} to generally denote the unit BGP installation cost associated to changing the redirection of the flow k to the SSP i's scrubbing center; we use e_{lk} to denote the unit BGP installation cost and the unit resource reconfiguration cost associated to changing the redirection and the scrubbing of the flow k to/at the ISP's own scrubbing center l. The corresponding switching cost of the ISP is thus written as $b_{ik}(y_{ikt} - y_{ikt-1})^+ + e_{lk}(z_{lkt} - z_{lkt-1})^+$, where we have $(\cdot)^+ \stackrel{\text{def}}{=} \max\{\cdot, 0\}$.

B. Social Cost Minimization Problem

Social Cost. In order to obtain the entire system's total cost over time, i.e., the social cost over time, we sum up the cost of the SSPs, i.e., $\sum_t \sum_i c_{it}x_{it} - \sum_t \sum_i \rho_{it}$, and the cost of the ISP, i.e., $\sum_t \sum_i \sum_k a_{ikt}y_{ikt} + \sum_t \sum_l \sum_k d_{lkt}z_{lkt} + \sum_t \sum_i \sum_k b_{ik}(y_{ikt} - y_{ikt-1})^+ + \sum_t \sum_l \sum_k e_{lk}(z_{lkt} - z_{lkt-1})^+ + \sum_t \sum_i \rho_{it}$. Note that the payment components of the SSPs and the ISP will cancel each other. We highlight it here, and we will design the payment calculation later.

Problem Formulation. We formulate the social cost minimization problem as below:

$$
\begin{aligned}
\min \quad & P = \textstyle\sum_t \sum_i c_{it}x_{it} \\
& + \textstyle\sum_t \sum_i \sum_k a_{ikt}y_{ikt} + \sum_t \sum_i \sum_k b_{ik}(y_{ikt} - y_{ikt-1})^+ \\
& + \textstyle\sum_t \sum_l \sum_k d_{lkt}z_{lkt} + \sum_t \sum_l \sum_k e_{lk}(z_{lkt} - z_{lkt-1})^+ \\
\text{s.t.} \quad & f_{ikt}x_{it} \geq y_{ikt}, \qquad \forall i, \forall k, \forall t, & \text{(1a)} \\
& \textstyle\sum_i y_{ikt} + \sum_l z_{lkt} \geq \lambda_{kt}, \qquad \forall k, \forall t, & \text{(1b)} \\
& \textstyle\sum_i y_{ikt} + \sum_l z_{lkt} \leq 1, \qquad \forall k, \forall t, & \text{(1c)} \\
& x_{it} \in \{0,1\}, y_{ikt} \in \{0,1\}, z_{lkt} \in \{0,1\}, \\
& \qquad \forall i, \forall k, \forall l, \forall t. & \text{(1d)}
\end{aligned}
$$

Constraint (1a) ensures that a flow is redirected to an SSP's scrubbing center only if the ISP procures the bid from that SSP and the bid claims to scrub that flow in its list. Constraints (1b) and (1c) ensure that every flow is handled, and must be handled by either an SSP's or the ISP's own scrubbing service, but not both, which captures the fact that one BGP route directs the flow to only one destination. Constraint (1d) ensures that all the decisions to be made are binary. There can exist different weights associated to each term in the objective; we omit such weights for the ease of presentation.

III. Online Social Cost Optimization

In this section, we design online algorithms to determine the winning bids and the traffic diversion to optimize the social cost with a provable competitive ratio. Solving our problem is challenging due to its *online* nature and *intractability*. To address these challenges, we propose (`i`) an online algorithm to obtain the fractional solutions for the relaxed problem and (`ii`) an online randomized rounding algorithm to convert such fractions to integers. We prove the overall competitive ratio of our approach as $r = r_1 r_2$, where r_1 is the competitive ratio associated to our online fractional algorithm and r_2 is the multiplicative integrality gap associated to our online rounding algorithm. In each of the two subsections, we elaborate the key challenge, present our algorithm, and perform the *competitiveness analysis* and the *integrality gap analysis*, respectively.

We also introduce some additional notations used throughout the rest of this paper: **P** is our original problem; **P'** is the relaxed problem of **P**; $\mathbf{P_t}$ is the one-shot problem at t of **P'**;

and $\tilde{\mathbf{P}}_\mathbf{t}$ is the "regularized" problem corresponding to $\mathbf{P_t}$. Also, we use P, P', P_t, and $\tilde{P}_t$ to refer to the objective functions of these problems, respectively. $\mathbf{x_t}$, $\mathbf{y_t}$, and $\mathbf{z_t}$ represent x_{it}, y_{ikt}, and z_{lkt}, $\forall i, \forall l, \forall k, \forall t$. $\tilde{\mathbf{x}}_\mathbf{t}$, $\tilde{\mathbf{y}}_\mathbf{t}$, and $\tilde{\mathbf{z}}_\mathbf{t}$ are the online fractional solutions; $\bar{\mathbf{x}}_\mathbf{t}$, $\bar{\mathbf{y}}_\mathbf{t}$, and $\bar{\mathbf{z}}_\mathbf{t}$ are the online integral solutions.

A. Regularization-based Online Fractional Algorithm

Algorithmic Challenge. After relaxing Constraint (1d) to allow fractional decisions, we still face the challenge of solving the problem online. The reason is that when determining $\{\mathbf{x_{t-1}},\mathbf{y_{t-1}},\mathbf{z_{t-1}}\}$ at $t-1$, $\{\mathbf{x_t},\mathbf{y_t},\mathbf{z_t}\}$ have not been determined, as they will only be determined at t; that is, without knowing $\{\mathbf{x_t},\mathbf{y_t},\mathbf{z_t}\}$, it is hard to decide $\{\mathbf{x_{t-1}},\mathbf{y_{t-1}},\mathbf{z_{t-1}}\}$ to minimize $(\mathbf{y}_t - \mathbf{y}_{t-1})^+$ and $(\mathbf{z}_t - \mathbf{z}_{t-1})^+$ at $t-1$.

Algorithm Design. Our idea of overcoming this challenge is replacing the switching cost in the objective function by carefully-designed "regularization" terms—logarithmic terms in our case—so that we actually change the problem, solve the changed problem at each single time slot, and use its solution as the solution to the problem before such changes [20]. Our intuition is that, at every time slot, without knowing what the inputs and our decision will be at the next time slot, we try to regularize our current decision in a controlled manner: if the workload increases (i.e., a flow appears) currently, there is no other choice because we must increase our decision to scrub the workload; if the workload decreases (i.e., a flow disappears) currently, then we reduce our decision conservatively rather than drastically, in order to prevent the excessive switching cost that could be incurred by the potential workload increase in the future.

Algorithm 1: Online Fractional Algorithm, $\forall t$

Solve the problem $\tilde{\mathbf{P}}_\mathbf{t}$ below and get the solution $\{\tilde{\mathbf{x}}_\mathbf{t},\tilde{\mathbf{y}}_\mathbf{t},\tilde{\mathbf{z}}_\mathbf{t}\}$:

$$\begin{array}{ll}\min & \tilde{P}_t = \sum_i c_{it}x_{it} + \sum_i \sum_k a_{ikt}y_{ikt} + \sum_l \sum_k d_{lkt}z_{lkt} \\ & + \sum_i \sum_k \frac{b_{ik}}{\sigma}\left((y_{ikt}+\varepsilon)\ln\frac{y_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon} - y_{ikt}\right) \\ & + \sum_l \sum_k \frac{e_{lk}}{\sigma}\left((z_{lkt}+\varepsilon)\ln\frac{z_{lkt}+\varepsilon}{\tilde{z}_{lkt-1}+\varepsilon} - z_{lkt}\right) \\ \text{s.t.} & \text{(1a)} \sim \text{(1c), without "}\forall t\text{"}, \\ & x_{it} \le 1, y_{ikt} \ge 0, z_{lkt} \ge 0, \forall i, \forall k, \forall l,\end{array}$$

where $\varepsilon > 0$ and $\sigma = \ln\left(1+\frac{1}{\varepsilon}\right)$ are parameters.

Analysis of Competitiveness. We prove the competitive ratio associated to our Algorithm 1. That is, we exhibit the constant r_1 which satisfies

$$P'(\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\}) \le r_1 D(\{\pi(\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}), \forall t\}) \le r_1 P^{OPT},$$

where P^{OPT} refers to the offline optimal value of the original problem $\mathbf{P}$, D refers to the objective function of the Lagrange dual problem $\mathbf{D}$ of the relaxed problem **P'**, and π refers to a mapping that can map our online fractional solutions $\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\}$ to a feasible solution to the Lagrange dual problem. We note that $D(\{\pi(\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}), \forall t\}) \le P^{OPT}$ holds naturally, due to weak duality and relaxation. Consequently, our job here can actually proceed with the following three steps: (`i`) deriving the Lagrange dual problem for the problem **P'**, (`ii`) constructing the mapping π, and (`iii`) finding out r_1 and proving $P'(\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\}) \le r_1 D(\{\pi(\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}), \forall t\})$.

Step 1: Deriving the Lagrange Dual Problem. We firstly present the relaxed problem **P'**:

$$\begin{array}{lll}\min & P' = \sum_t P_t = \sum_t \sum_i c_{it}x_{it} & \\ & + \sum_t \sum_i \sum_k a_{ikt}y_{ikt} + \sum_t \sum_l \sum_k d_{lkt}z_{lkt} & \\ & + \sum_t \sum_i \sum_k b_{ik}w_{ikt} + \sum_t \sum_l \sum_k e_{lk}v_{lkt} & \\ \text{s.t.} & \text{(1a)} \sim \text{(1c)}, & \\ & y_{ikt} - y_{ikt-1} \le w_{ikt}, & \forall i, \forall k, \forall t, \\ & z_{lkt} - z_{lkt-1} \le v_{lkt}, & \forall l, \forall k, \forall t, \\ & x_{it} \le 1, y_{ikt} \ge 0, z_{lkt} \ge 0, \forall i, \forall k, \forall l, \forall t, & \end{array}$$

where we introduce the auxiliary variables w_{ikt} and v_{lkt} and change the problem to an equivalent linear program.

Then, following the definition of the Lagrange dual problem, we derive the dual problem for **P'**, denoted as **D**, where α_{kt}, β_{ikt}, μ_{kt}, γ_{it}, ϕ_{ikt}, and τ_{lkt} are the dual variables [27]:

$$\begin{array}{lll}\max & D = \sum_t \sum_k \alpha_{kt}\lambda_{kt} + \sum_t \sum_k (\sum_k \lambda_{kt} - 1)\mu_{kt} & \\ & + \sum_t \sum_i (\sum_k \lambda_{kt} - 1)\gamma_{it} & \\ \text{s.t.} & c_{it} - \sum_k f_{ikt}\beta_{ikt} + \gamma_{it} - \sum_i \sum_k f_{ikt}\gamma_{it} \ge 0, \forall i, \forall t, & \\ & a_{ikt} - \alpha_{kt} + \beta_{ikt} + \mu_{kt} - \sum_k \mu_{kt} + \phi_{ikt} - \phi_{ikt+1} \ge 0, & \\ & & \forall i, \forall k, \forall t, \\ & d_{lkt} - \alpha_{kt} + \mu_{kt} - \sum_k \mu_{kt} - \sum_i \gamma_{it} + \tau_{lkt} - \tau_{lkt+1} \ge 0, & \\ & & \forall l, \forall k, \forall t, \\ & b_{ik} - \phi_{ikt} \ge 0, & \forall i, \forall k, \forall t, \\ & e_{lk} - \tau_{lkt} \ge 0, & \forall l, \forall k, \forall t, \\ & \text{all dual variables} \ge 0. & \end{array}$$

Step 2: Constructing the Mapping. We present the mapping π that maps our online fractional solutions, i.e., the optimal solution to $\tilde{\mathbf{P}}_\mathbf{t}$, $\forall t$, together with the dual solution to $\tilde{\mathbf{P}}_\mathbf{t}$, $\forall t$ to a feasible solution of **D**. It can be easily verified that the mapped solutions satisfy all of **D**'s constraints.

$$\alpha_{kt} = \alpha_k, \forall k; \beta_{ikt} = \beta_{ik}, \forall i, \forall k; \mu_{kt} = \mu_k, \forall k; \gamma_{it} = \gamma_i, \forall i;$$
$$\phi_{ikt} = \frac{b_{ik}}{\sigma}\ln\frac{1+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon}, \forall i, \forall k;\ \tau_{lkt} = \frac{e_{lk}}{\sigma}\ln\frac{1+\varepsilon}{\tilde{z}_{lkt-1}+\varepsilon}, \forall l, \forall k.$$

Step 3: Finding out r_1 and Upper-Bounding P'. We place our online fractional solutions $\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\}$ into P', and then leverage a chain of inequalities to connect it to D with $\{\pi(\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}), \forall t\}$ while finding out the constant r_1. We use the Karush-Kuhn-Tucker (KKT) conditions of the problem $\tilde{\mathbf{P}}_\mathbf{t}$ to derive this chain of inequalities. We bound the operation cost and the switching cost of P', respectively, in Lemmas 1 and 2, based on which we further have Theorem 1. We move all the proof details into our appendices.

Lemma 1. *The operation cost in P' satisfies $\sum_t \sum_i c_{it}\tilde{x}_{it} + \sum_t \sum_i \sum_k a_{ikt}\tilde{y}_{ikt} + \sum_t \sum_l \sum_k d_{lkt}\tilde{z}_{lkt} \le D$.*

Proof. See Appendix A. □

Lemma 2. *For the switching cost in P', we have the following: $\sum_t \sum_i \sum_k b_{ik}(\tilde{y}_{ikt} - \tilde{y}_{ikt-1})^+ + \sum_t \sum_l \sum_k e_{lk}(\tilde{z}_{lkt} - \tilde{z}_{lkt-1})^+ \le 2(1+\varepsilon)\ln(1+\frac{1}{\varepsilon})|\mathcal{K}|D$.*

Proof. See Appendix B. □

Theorem 1. *$P'(\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\}) \leq r_1 D(\{\pi(\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}), \forall t\})$, where $r_1 = 1 + 2(1+\varepsilon)\ln(1+\frac{1}{\varepsilon})|\mathcal{K}|$.*

Proof. The proof is by joining Lemmas 1 and 2. □

B. Progressive Randomized Rounding Algorithm

Algorithmic Challenge. The challenge for rounding the fractional solution $\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}\}$ to integers at each t is how to ensure the problem's constraints, i.e., (1a) $\sim$ (1c), are still satisfied after rounding. There is no guarantee to keep such satisfaction if the each single variable is rounded independently.

Algorithm Design. Our idea is two-fold. First, we round our fractional solutions progressively. We round the fractional $\tilde{\mathbf{x}}_\mathbf{t}$ to the integral $\bar{\mathbf{x}}_\mathbf{t}$, place such $\bar{\mathbf{x}}_\mathbf{t}$ into the problem (while keeping the problem feasible), re-solve the problem to obtain the new fractional $\{\mathbf{y}^*_\mathbf{t}, \mathbf{z}^*_\mathbf{t}\}$, and then round them to the integral $\{\bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}\}$. This ensures Constraint (1a) is satisfied. Second, we round both $\tilde{\mathbf{x}}_\mathbf{t}$ and $\mathbf{y}^*_\mathbf{t}, \mathbf{z}^*_\mathbf{t}$ in a pair-by-pair manner. In every iteration, we always round a pair of fractions altogether, so that one or both of them can become integral while compensating each other and keeping their weighted sum constant before and after rounding [21], [28]. This ensures (1b) and (1c) are satisfied. We design Algorithm 2 based on this idea.

The rounding procedure of Algorithm 2 is mainly Line 7 through 20. Consider rounding $\tilde{\mathbf{x}}_\mathbf{t}$, for example. Every iteration in the loop of Line 9 through 19 ensures the following: (i) either $\tilde{x}_{i_1 t}$, or $\tilde{x}_{i_2 t}$, or both are rounded into integer(s); (ii) we have $U_{i_1}\theta'_{i_1 t} + U_{i_2}\theta'_{i_2 t} = U_{i_1}\theta_{i_1 t} + U_{i_2}\theta_{i_2 t}$, no matter we choose Line 13 or 14; (iii) the expectation of the integral $\bar{x}_{it}$ equals to the fractional $\tilde{x}_{it}$, i.e., $E(\bar{x}_{it}) = \tilde{x}_{it}, \forall i \in \mathcal{I} \setminus \mathcal{I}'_t$—for example, if $\tilde{x}_{i_2 t}$ becomes integral, then $E(\bar{x}_{i_2 t}) = \frac{\omega_2}{\omega_1+\omega_2}(\tilde{x}_{i_2 t} - \frac{U_{i_1}}{U_{i_2}}\omega_1) + \frac{\omega_1}{\omega_1+\omega_2}(\tilde{x}_{i_2 t} + \frac{U_{i_1}}{U_{i_2}}\omega_2) = \tilde{x}_{i_2 t}$. This equation will be utilized for the integrality gap analysis and for the payment design later, which also motivates us to design a randomized rather than a deterministic rounding algorithm.

Analysis of Integrality Gap. We analyze the integrality gap incurred by our Algorithm 2. That said, we demonstrate the constant r_2 which satisfies

$$E(P'(\{\bar{\mathbf{x}}_\mathbf{t}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}, \forall t\})) \leq r_2 P'(\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\}).$$

We follow two steps: (i) upper-bounding $\sum_t \sum_i \sum_k f_{ikt}\bar{x}_{it}$ by a constant times $P'(\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\})$; (ii) based on (i), upper-bounding each of the terms in $E(P'(\{\bar{\mathbf{x}}_\mathbf{t}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}, \forall t\}))$, respectively, and deriving the constant r_2. We choose to start with $\sum_t \sum_i \sum_k f_{ikt}\bar{x}_{it}$, because it is that weighted sum that is tried to be maintained unchanged in Algorithm 2. As a matter of fact, corresponding to the two steps, we show the following Lemma 3, based on which we further show Theorem 2.

Lemma 3. *For every value the random variable $\bar{x}_{it}, \forall i, \forall t$ takes, we have $\sum_t \sum_i \sum_k f_{ikt}\bar{x}_{it} \leq r'_2 P'(\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\})$, where $r'_2 = \max_{t,i,k} \frac{2f_{ikt}}{c_{it}} + \max_{t,k,l} \frac{1}{d_{lkt}}$.*

Proof. See Appendix C. □

Algorithm 2: Progressive Rounding Algorithm, $\forall t$

▷ First, round $\tilde{\mathbf{x}}_\mathbf{t}$.
1 Denote $\bar{x}_{it}$ as $\bar{u}_{it}$, $\tilde{x}_{it}$ as $\hat{u}_{it}$, and $\sum_k f_{ikt}$ as U_i, $\forall i$;
2 Execute Line 7 through 20, and continue with Line 3;
▷ Then, based on $\bar{\mathbf{x}}_\mathbf{t}$, obtain and round $\mathbf{y}^*_\mathbf{t}$ and $\mathbf{z}^*_\mathbf{t}$.
3 Fix $\bar{\mathbf{x}}_\mathbf{t}$, solve $\mathbf{P_t}$ and get its solution $\{\bar{\mathbf{x}}_\mathbf{t}, \mathbf{y}^*_\mathbf{t}, \mathbf{z}^*_\mathbf{t}\}$;
4 Denote $\mathcal{I} \cup \mathcal{L}$ as the new $\mathcal{I}$, and execute Lines 5 and 6 for all k;
5 Denote $\bar{y}_{ikt}$, $\bar{z}_{lkt}$ as $\bar{u}_{it}$, y^*_{ikt}, z^*_{lkt} as $\hat{u}_{it}$, and 1 as U_i;
6 Execute Line 7 through 20;
▷ Round the given fractions in a randomized manner.
7 $\theta_{it} \overset{\text{def}}{=} \hat{u}_{it}, \forall i$;
8 $\mathcal{I}'_t \overset{\text{def}}{=} \mathcal{I} \setminus \{i | \theta_{it} \in \{0,1\}\}$;
9 **while** $|\mathcal{I}'_t| > 1$ **do**
10   Select $i_1, i_2 \in \mathcal{I}'_t$, where $i_1 \neq i_2$;
11     $\omega_1 \overset{\text{def}}{=} \min\{1 - \theta_{i_1 t}, \frac{U_{i_2}}{U_{i_1}}\theta_{i_2 t}\}$;
12     $\omega_2 \overset{\text{def}}{=} \min\{\theta_{i_1 t}, \frac{U_{i_2}}{U_{i_1}}(1 - \theta_{i_2 t})\}$;
13   With the probability $\frac{\omega_2}{\omega_1+\omega_2}$,
       Set $\theta'_{i_1 t} = \theta_{i_1 t} + \omega_1, \theta'_{i_2 t} = \theta_{i_2 t} - \frac{U_{i_1}}{U_{i_2}}\omega_1$;
14   With the probability $\frac{\omega_1}{\omega_1+\omega_2}$,
       Set $\theta'_{i_1 t} = \theta_{i_1 t} - \omega_2, \theta'_{i_2 t} = \theta_{i_2 t} + \frac{U_{i_1}}{U_{i_2}}\omega_2$;
15   **if** $\theta'_{i_1 t} \in \{0,1\}$ **then** Set $\bar{u}_{i_1 t} = \theta'_{i_1 t}, \mathcal{I}'_t = \mathcal{I}'_t \setminus \{i_1\}$;
16   **else** Set $\theta_{i_1 t} = \theta'_{i_1 t}$;
17   **if** $\theta'_{i_2 t} \in \{0,1\}$ **then** Set $\bar{u}_{i_2 t} = \theta'_{i_2 t}, \mathcal{I}'_t = \mathcal{I}'_t \setminus \{i_2\}$;
18   **else** Set $\theta_{i_2 t} = \theta'_{i_2 t}$;
19 **end**
20 **if** $|\mathcal{I}'_t| = 1$ **then** Set $\bar{u}_{it} = 1$ for the only $i \in \mathcal{I}'_t$;

Theorem 2. *With Lemma 3, we have $E(P'(\{\bar{\mathbf{x}}_\mathbf{t}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}, \forall t\})) \leq r_2 P'(\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}, \forall t\})$, where $r_2 = \delta_x + \delta_{yz} + \delta_{wv}$,*

$$\delta_x = r'_2 \max_{t,i} \frac{c_{it}}{\sum_k f_{ikt}},$$
$$\delta_{yz} = 2|\mathcal{K}|(\max_{t,i,k} a_{ikt} + \max_{t,l,k} d_{lkt})(\max_{t,i,k} \frac{1}{a_{ikt}} + \max_{t,l,k} \frac{1}{d_{lkt}}),$$
$$\delta_{wv} = 2|\mathcal{K}|(\max_{i,k} b_{ik} + \max_{l,k} e_{lk})(\max_{i,k} \frac{1}{b_{ik}} + \max_{l,k} \frac{1}{e_{lk}}).$$

Proof. See Appendix D. □

IV. Randomized Online Auction Mechanism

In this section, we design the payment for each bid, present the entire online auction mechanism, and prove the desired properties of "truthfulness" and "individual rationality".

Algorithm 3: Online Randomized Auction Mechanism, $\forall t$

▷ First, determine winning bids and traffic diversion.
Invoke Algorithm 1 to obtain $\{\tilde{\mathbf{x}}_\mathbf{t}, \tilde{\mathbf{y}}_\mathbf{t}, \tilde{\mathbf{z}}_\mathbf{t}\}$;
With $\tilde{\mathbf{x}}_\mathbf{t}$ as input, invoke Algorithm 2 to obtain $\{\bar{\mathbf{x}}_\mathbf{t}, \bar{\mathbf{y}}_\mathbf{t}, \bar{\mathbf{z}}_\mathbf{t}\}$;
▷ Then, determine the payment for each winning bid.
With $\tilde{\mathbf{x}}_\mathbf{t}$ and $\bar{\mathbf{x}}_\mathbf{t}$ as inputs,
for $i \in \mathcal{I}$
    if $\bar{x}_{it} = 0$ **then** Set $\rho_{it} = 0$;
    else Set $\rho_{it} = c_{it}\tilde{x}_{it}(c_{it}, \mathbf{c}_{-it}) + \int_{c_{it}}^{\zeta_{it}} \tilde{x}_{it}(c, \mathbf{c}_{-it})dc$,
    where $\zeta_{it} = \sum_l \sum_k f_{ikt}(d_{lkt} + e_{lk})$.
end

Payment and Auction Mechanism Design. Our online auction mechanism in Algorithm 3 is a randomized mechanism, because Algorithm 2 introduces randomization into rounding. In Algorithm 3, ρ_{it} is the payment the auctioneer pays in order to buy the bid i in the auction at t, $\mathbf{c}_{-it}$ refers to the prices of all the bids except for the bid i, i.e., $\{c_{jt}|\forall j \in \mathcal{I}, j \neq i\}$, and $\tilde{x}_{it}(c_{it}, \mathbf{c}_{-it})$, written as a function of c_{it} and $\mathbf{c}_{-it}$, refers to the fractional solution returned by Algorithm 2 when the bid i bids the price of c_{it} and other bids bid the prices of $\mathbf{c}_{-it}$. We note $E(\bar{x}_{it}) = \tilde{x}_{it}, \forall i \in \mathcal{I} \setminus \mathcal{I}'_t$, as shown previously. We also note $\zeta = \sum_l \sum_k f_{ikt}(d_{lkt} + e_{lk})$, which serves as an upper bound for the integral, and captures the extreme case where the SSPs' scrubbing center i would lose in the auction for sure if i's bidding price is higher than the cost of scrubbing the same traffic by the ISP's scrubbing centers.

Analysis of Truthfulness and Individual Rationality. In our randomized auction mechanism, for each bid (or bidder, as a bidder issues only one bid) i that has the true cost c_{it}, the expected utility is

$$u_i(b_{it}, \mathbf{b}_{-it}) \stackrel{\text{def}}{=} \rho_i(b_{it}, \mathbf{b}_{-it}) - c_{it} E(\bar{x}_{it}(b_{it}, \mathbf{b}_{-it})),$$

where b_{it} denotes the bidding price of the bid i, $\mathbf{b}_{-it}$ denotes the bidding prices of other bids except i, and ρ_{it} is the payment received from the auctioneer. Because our auction mechanism is truthful as shown next, we assume everyone bids its true cost by default, and thus we have been using c_{it}, rather than b_{it}, to denote the bidding price; but we need to differentiate them when defining utility. We have the following definitions:

Definition 1. ***Truthfulness.*** *A randomized auction is truthful in expectation if every bidder i maximizes its expected utility by bidding its truth cost, i.e., $u_i(c_{it}, \mathbf{b}_{-it}) \geq u_i(b_{it}, \mathbf{b}_{-it})$, $\forall\, b_{it} \neq c_{it}$, $\forall\, \mathbf{b}_{-it}$.*

Definition 2. ***Individual Rationality.*** *A randomized auction is individually rational in expectation if every bidder i always has a nonnegative expected utility, i.e., $u_i(b_{it}, \mathbf{b}_{-it}) \geq 0$, $\forall\, b_{it}$, $\forall\, \mathbf{b}_{-it}$.*

A randomized auction needs to satisfy the sufficient and necessary conditions [22] in order to be both truthful and individually rational in expectation. These conditions are centered around the "monotone allocation rule", i.e., the higher price a bid bids, the less "workload" (i.e., $E(\bar{x}_{it})$ in our scenario) it receives from the auctioneer, and the "finite payment rule", i.e., the payment is always a finite value. This motivates us to derive a randomized rather than a deterministic auction mechanism, as we can exploit the fractional solution $\tilde{\mathbf{x}}_{\mathbf{t}}$ that satisfies such a monotone allocation rule; in contrast, it may be hard for a deterministic rounding algorithm to achieve such monotonicity. In Theorem 3, we demonstrate the sufficient and necessary conditions, with the fact that our randomized auction in each time slot satisfies these conditions indeed:

Theorem 3. *The online randomized auction mechanism of Algorithm 3 achieves truthful bidding and individual rationality in expectation, by satisfying the following conditions: (`i`) $E(\bar{x}_{it})$ is monotonically nonincreasing in terms of $c, \forall i$; (`ii`) $\int_0^\infty E(\bar{x}_{it}) dc < \infty, \forall i$; (`iii`) the payment is in the form of $\rho_i = c_{it} E(\bar{x}_{it}(c_{it}, \mathbf{c}_{-it})) + \int_{c_{it}}^\infty E(\bar{x}_{it}(c, \mathbf{c}_{-it})) dc, \forall i$.*

Proof. See Appendix E. □

V. Numerical Evaluation

A. Evaluation Setup

ISP, SSPs, and Traffic: We simulate the scenario where an ISP purchases scrubbing services from multiple SSPs. We assume that the ISP owns 10 internal scrubbing centers in its network, and vary the number of SSPs as $|\mathcal{I}| = 5$, 10, 15, 20, 25, 30, respectively. We treat one hour as one time slot, and consider a 200-hour horizon. The maximum number of traffic flows is set as $|\mathcal{K}| = 1000$, 2000 and 3000, respectively. We randomly generate λ_{kt} within the ranges to reflect the variation of the number of flows over time. We set ε as 0.001.

Prices: We adopt Amazon EC2's c3.2xlarge [23] virtual machine (VM) price for Linux/UNIX as the SSPs' bidding price $c_{it} \simeq \$0.0645$/hour, and set the BGP routing cost as $a_{ikt} \simeq \$0.913242$/hour [12], varying with time. We choose the dynamic hourly electricity price in the Chicago area to be the unit operational cost of the internal scrubbing centers d_{lkt}, which follows a Gaussian distribution with a mean of \$40.6/MWh and a standard deviation of \$26.9/MWh [24].

Switching Cost: We synthesize the switching cost b_{ik} and e_{lk} as the VM price times a weight which we vary as 1, 10 and 100 to reveal how the switching cost impacts the results.

Algorithms: We use Python and A Mathematical Programming Language (AMPL) [29] to implement our algorithms, and invoke the interior-point-based IPOPT [30] solver to solve the underlying fractional problems. We run Algorithm 1 to obtain the fractional solutions. Then, we upload such results to Algorithm 2 to obtain the integer solutions and determine the winning bids. Finally, we prompt Algorithm 3 to calculate the payments to the winning bids.

For the social cost comparison, we implement the following algorithms: (1) the approach that uses the Gurobi [25] solver to solve each one-shot integer program separately (which essentially ignores the switching cost), (2) the state-of-the-art Lazy Capacity Provisioning [26] algorithm for solving online optimization problems with the switching cost, and (3) the offline optimum, which knows all the inputs in advance and uses Gurobi to solve the integer program over the entire time horizon. We refer to them as "`Gurobi`", "`Lcp`", and "`Offline`", respectively. We run our evaluations on a laptop with an Intel Core i7 2.7-GHz CPU and 16-GB memory.

B. Evaluation Results

Social Cost of Our Algorithm: The normalized social cost over the entire time horizon is given in Fig. 3. The top figure focuses on the influence of the different number of SSPs. The social cost mostly decreases as the number of SSPs grows. That is, when an ISP has more purchase options to choose from, the whole market becomes more competitive, and thus it is more likely for the ISP to purchase the services at a lower price, i.e., the social cost decreases. The bottom figure shows that the heavier the weight on the switching cost is, the less likely the ISP switches services across scrubbing centers from time to time. As a result, the social cost also reduces.

Social Cost Comparison: We compare the social cost of different algorithms over 200 hours with different numbers of

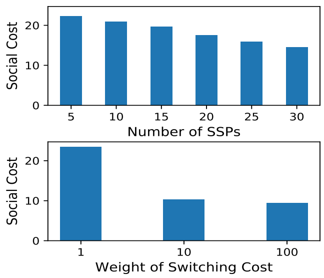

Fig. 3: Social Cost

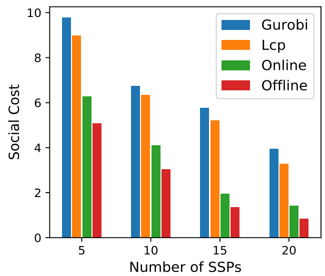

Fig. 4: Social Cost Comparison

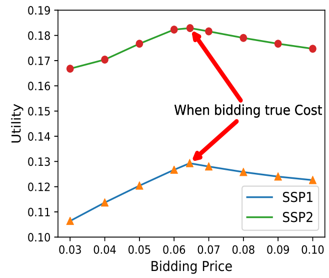

Fig. 5: Utility

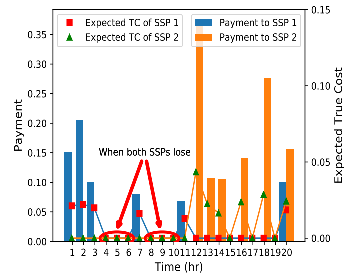

Fig. 6: Payment

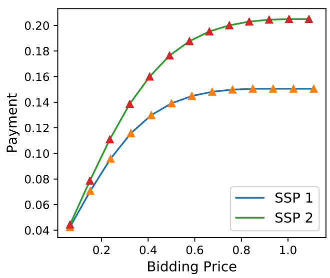

Fig. 7: Frugality

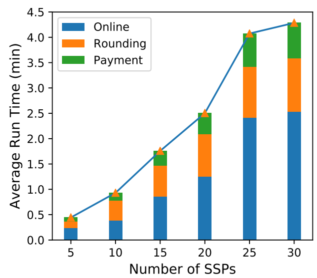

Fig. 8: Running Time

SSPs in Fig. 4. Our approach has up to 35.47% and 31.97% less total cost than `Gurobi` and `Lcp`, respectively, and only incurs 11.06% more total cost than `Offline`. Considering the influence of the number of SSPs, this result matches the top figure in Fig. 3, indicating the more choices the ISP has, the more economically efficient the whole system becomes.

Truthfulness: We randomly choose two SSPs and calculate their utilities when varying their bidding prices in Fig 5. It is clear that only when SSPs are bidding their true costs, they receive the maximum utilities. If SSPs are bidding prices other than the true cost, the utilities are always lower. Our payment design successfully induces SSPs to bid truthfully.

Individual Rationality: We randomly choose 20 consecutive hours and two SSPs and depict their received payments in Fig. 6. It shows that the payments can vary as time goes, and that an SSP may not win in every auction (and actually all SSPs may lose in an auction, as the ISP's internal scrubbing centers may take charge). We note that the payment is always no less than the expected total cost (TC), i.e., our payment design ensures the individual rationality.

Payment Frugality: In Fig 7, we check whether the payments in our auctions are "frugal", compared to the true cost of the bids. We observe the following for two randomly-chosen SSPs: as the bidding price goes up, the payment does not keep growing along with the bidding price but gradually converges to a constant value at some point. This shows that our payment function avoids large over-payments compared to the true cost. Thus, we can claim that our proposed payment design for the online auction mechanism preserves frugality.

Algorithm Running Time: We study our algorithm's running time in Fig. 8. The scattered line shows the overall approach's running time. The stacked bars show the running time of each algorithmic component. As the number of traffic flows is up to 2000 and the number of SSPs is up to 30, our approach only takes less than 5 minutes to finish each round of the auction. With more powerful servers or data centers, our algorithm's running time can be further reduced in practice.

VI. Related Work

Cloud Scrubbing for DDoS Mitigation. Zilberman et. al. [4] study the scrubbing center deployment strategies and their impact on network footprint, load, and latency. Dietzel et. al. [11] implement fine-grained blackholing using hardware filters and signaling mechanisms for traffic scrubbing that can jointly work with scrubbing centers. Jiao et. al. [14] schedule traffic flows into different scrubbing centers to minimize the total network footprint of unknown malicious traffic. Liu et. al. [15] design a multi-layer defense approach easily deployable for ISPs while preserving customers' privacy. Jin et. al. [16] discover customers' potential IP address leakage when they use domain name system based scrubbing center services, and propose corresponding countermeasures.

Online Auctions for Cloud Management. Substantial research efforts adopt online auctions for cloud resource provisioning. Shi et. al. [7] focus on tenants' long-term budget and the cloud's dynamic resource availability in the auction design. Zhu et. al. [8] sublet tenants' underutilized virtual machines to others via auctions. Zhang et. al. [9] incorporate tenants' desired occupation duration and cloud servers' operation cost in the auctions. Zhou et. al. [10] bid cloud resources for job execution while considering job deadline violation and server operation cost. Another branch of research concentrates on demand response, and manages cloud resource usage by incentivizing tenants to reduce grid energy consumption through reducing workload and/or shutting down servers [17]–[19].

Our research differs from all the aforementioned existing works. Unlike [4], [11], [14]–[16] that often focus on the service operator's perspective only, we investigate the interactions between ISPs and SSPs, and the intersection between the cloud scrubbing operation and the online auction markets. While the primal-dual-based online algorithms are predominantly exploited in the auction papers mentioned above to optimize the social cost or welfare, it remains unclear how to adapt such algorithms to address the auctioneer's switching cost, as featured in our problem. For payment design, [17] uses the VCG mechanism. To overcome the inapplicability of VCG, [7] resorts to its fractional version, plus randomized auctions; [8]–[10] switch to the posted price mechanisms; and [18], [19] relate to Myerson's monotone allocation rules. None of them are readily applicable to our case. The most promising ones may be [18], [19] which also have the switching cost; however, the former relies on accurate predictions of future inputs, which are unavailable in our settings, and the latter produces a less satisfactory competitive ratio depending on the solution (unknown before solving the problem) instead of input parameters. Our approach, with no prediction, attains a constant competitive ratio relying merely on the input parameters, and preserves truthfulness and individual rationality.

VII. Conclusion

In this paper, we propose an online auction mechanism to enable ISPs to procure traffic scrubbing services from external SSPs to scrub the dynamic, unpredictable traffic. We devise an online fractional algorithm and a randomized rounding algorithm to determine the winning bids and the traffic diversion in each single-round auction, with a provable competitive ratio for the long-term social cost. We also design the payment calculation based on each bid's winning probability to ensure truthfulness and individual rationality. Finally, we conduct evaluations using real-world data to validate the theoretical properties and the practical efficacy of our mechanism.

Acknowledgement

Ruiting Zhou appreciates the support from the National Natural Science Foundation of China (61502504), the Technological Innovation Major Projects of Hubei Province (2017AAA125), and the Science and Technology Program of Wuhan City (2018010401011288).

Appendix

A. Proof of Lemma 1

The proofs to Lemmas 1 and 2 exploit the KKT conditions for the problem $\tilde{\mathbf{P}}_\mathbf{t}$. To derive these KKT conditions, we firstly transform $\tilde{\mathbf{P}}_\mathbf{t}$ to the following equivalent form [27]:

$$\begin{aligned}
\min\quad & \tilde{P}_t = \sum_i c_{it}x_{it} + \sum_i \sum_k a_{ikt}y_{ikt} + \sum_l \sum_k d_{lkt}z_{lkt} \\
& + \sum_i \sum_k \tfrac{b_{ik}}{\sigma}((y_{ikt}+\varepsilon)\ln \tfrac{y_{ikt}+\varepsilon}{y_{ikt-1}+\varepsilon} - y_{ikt}) \\
& + \sum_l \sum_k \tfrac{e_{lk}}{\sigma}((z_{lkt}+\varepsilon)\ln \tfrac{z_{lkt}+\varepsilon}{z_{lkt-1}+\varepsilon} - z_{lkt}) \\
\text{s.t.}\quad & \text{(1a)} \sim \text{(1b), without “}\forall t\text{”}, \\
& \sum_k \sum_i y_{ikt} + \sum_k \sum_l z_{lkt} - \sum_i y_{ikt} - \sum_l z_{lkt} \\
& \qquad\qquad \geq \sum_k \lambda_{kt} - 1, \quad \forall k, \\
& \sum_k \sum_i f_{ikt}x_{it} + \sum_k \sum_l z_{lkt} - x_{it} \\
& \qquad\qquad \geq \sum_k \lambda_{kt} - 1, \quad \forall i, \forall j.
\end{aligned}$$

For this new form of $\tilde{\mathbf{P}}_\mathbf{t}$, we write the KKT conditions that characterize the optimal solution of $\tilde{\mathbf{P}}_\mathbf{t}$:

$$\begin{aligned}
& c_{it} - \sum_k f_{ikt}\beta_{ik} + \gamma_i - \sum_i \sum_k f_{ikt}\gamma_i = 0, \forall i, && \text{(2a)} \\
& a_{ikt} + \tfrac{b_{ik}}{\sigma}\ln \tfrac{\tilde{y}_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon} - \alpha_k + \beta_{ik} + \mu_k - \sum_k \mu_k = 0, \\
& \qquad\qquad \forall i, \forall k, && \text{(2b)} \\
& d_{lkt} + \tfrac{e_{lk}}{\sigma}\ln \tfrac{\tilde{z}_{lkt}+\varepsilon}{\tilde{z}_{lkt-1}+\varepsilon} - \alpha_k + \mu_k - \sum_k \mu_k - \sum_i \gamma_i = 0, \\
& \qquad\qquad \forall l, \forall k, && \text{(2c)} \\
& \alpha_k(\lambda_{kt} - \sum_i \tilde{y}_{ikt} - \sum_l \tilde{z}_{lkt}) = 0, \forall k, && \text{(2d)} \\
& \beta_{ik}(\tilde{y}_{ikt} - f_{ikt}\tilde{x}_{it}) = 0, \quad \forall i, \forall k, && \text{(2e)} \\
& \mu_k(\sum_k \sum_i \tilde{y}_{ikt} + \sum_k \sum_l \tilde{z}_{lkt} - \sum_i \tilde{y}_{ikt} - \sum_l \tilde{z}_{lkt} \\
& \quad -(\sum_k \lambda_{kt} - 1)) = 0, \quad \forall k, && \text{(2f)} \\
& \gamma_i(\sum_k \sum_i f_{ikt}\tilde{x}_{it} + \sum_k \sum_l \tilde{z}_{lkt} - \tilde{x}_{it} \\
& \quad -(\sum_k \lambda_{kt} - 1)) = 0, \quad \forall i, && \text{(2g)} \\
& \text{primal and dual solutions} \geq 0. && \text{(2h)}
\end{aligned}$$

Now, we bound the operation cost in P':

$$\begin{aligned}
& \sum_t \sum_i c_{it}\tilde{x}_{it} + \sum_t \sum_i \sum_k a_{ikt}\tilde{y}_{ikt} + \sum_t \sum_l \sum_k d_{lkt}\tilde{z}_{lkt} \\
=& \sum_t \sum_i (\sum_k f_{ikt}\beta_{ikt} - \gamma_{it} + \sum_i \sum_k f_{ikt}\gamma_{it})\tilde{x}_{it} \\
& + \sum_t \sum_i \sum_k (\alpha_{kt} - \beta_{ikt} - \mu_{kt} + \sum_k \mu_{kt} - \tfrac{b_{ik}}{\sigma}\ln \tfrac{\tilde{y}_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon})\tilde{y}_{ikt} \\
& + \sum_t \sum_l \sum_k (\alpha_{kt} - \mu_{kt} + \sum_k \mu_{kt} + \sum_i \gamma_{it} \\
& \quad - \tfrac{e_{lk}}{\sigma}\ln \tfrac{\tilde{z}_{lkt}+\varepsilon}{\tilde{z}_{ikt-1}+\varepsilon})\tilde{z}_{lkt} && \text{(3a)} \\
\leq& \sum_t \sum_i (\sum_i \sum_k f_{ikt}\gamma_{it} - \gamma_{it})\tilde{x}_{it} \\
& + \sum_t \sum_i \sum_k (\alpha_{kt} - \mu_{kt} + \sum_k \mu_{kt})\tilde{y}_{ikt} \\
& + \sum_t \sum_l \sum_k (\alpha_{kt} - \mu_{kt} + \sum_k \mu_{kt} + \sum_i \gamma_{it})\tilde{z}_{lkt} && \text{(3b)} \\
=& \sum_t \sum_i \gamma_{it}(\sum_i \sum_k f_{ikt}\tilde{x}_{it} - \tilde{x}_{it} + \sum_l \sum_k \tilde{z}_{lkt}) \\
& + \sum_t \sum_k \alpha_{kt}(\sum_i \tilde{y}_{ikt} + \sum_l \tilde{z}_{lkt}) \\
& + \sum_t \sum_k \mu_{kt}(\sum_k \sum_i \tilde{y}_{ikt} - \sum_i \tilde{y}_{ikt} + \sum_k \sum_l \tilde{z}_{lkt} - \sum_l \tilde{z}_{lkt}) \\
& && \text{(3c)} \\
=& \sum_t \sum_k \alpha_{kt}\lambda_{kt} + \sum_t \sum_k \mu_{kt}(\sum_k \lambda_{kt} - 1) \\
& + \sum_t \sum_i \gamma_{it}(\sum_k \lambda_{kt} - 1) && \text{(3d)} \\
=& D.
\end{aligned}$$

(3a) holds because of (2a) $\sim$ (2c). (3b) holds, due to the two inequalities $\sum_t \tilde{y}_{ikt}\ln \frac{\tilde{y}_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon} \geq 0$ and $\sum_t \tilde{z}_{lkt}\ln \frac{\tilde{z}_{lkt}+\varepsilon}{\tilde{z}_{lkt-1}+\varepsilon} \geq 0$. Here, we only prove the former inequality as an example; the latter can be proved analogously. We equip ourselves with the following two inequalities first: $\forall p, \forall q > 0$,

$$(\textstyle\sum_n p_n)\ln \frac{\sum_n p_n}{\sum_n q_n} \leq \sum_n p_n \ln \frac{p_n}{q_n}, p - q \leq p\ln \frac{p}{q}.$$

Then, based on the above, we have the following, $\forall i, \forall k$:

$$\begin{aligned}
& \sum_t \tilde{y}_{ikt}\ln \tfrac{\tilde{y}_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon} \\
=& \sum_t (\tilde{y}_{ikt}+\varepsilon)\ln \tfrac{\tilde{y}_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon} - \sum_t \varepsilon \ln \tfrac{\tilde{y}_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon} \\
\geq& (\sum_t (\tilde{y}_{ikt}+\varepsilon))\ln \tfrac{\sum_t (\tilde{y}_{ikt}+\varepsilon)}{\sum_t (\tilde{y}_{ikt-1}+\varepsilon)} + (\tilde{y}_{ik0}+\varepsilon)\ln \tfrac{\tilde{y}_{ik0}+\varepsilon}{\tilde{y}_{ikT}+\varepsilon} \\
\geq& \sum_t (\tilde{y}_{ikt}+\varepsilon) - \sum_t (\tilde{y}_{ikt-1}+\varepsilon) + \tilde{y}_{ik0} - \tilde{y}_{ikT} \\
=& 0.
\end{aligned}$$

We continue with (3c). (3c) holds due to (2d), (2f) and (2g). (3d) holds due to the definition of D.

B. Proof of Lemma 2

We bound the switching cost in P'. Here, we define $\eta = (1+\varepsilon)\sigma, \mathcal{I}' = \{i|\tilde{y}_{ikt} > \tilde{y}_{ikt-1}\}$, and $\mathcal{L}' = \{l|\tilde{z}_{lkt} > \tilde{z}_{lkt-1}\}$. Then, we have

$$\begin{aligned}
& \sum_t \sum_i \sum_k b_{ik}(\tilde{y}_{ikt} - \tilde{y}_{ikt-1})^+ \\
& + \sum_t \sum_l \sum_k e_{lk}(\tilde{z}_{lkt} - \tilde{z}_{lkt-1})^+ \\
=& \sum_t \sum_k \sum_{i \in \mathcal{I}'} b_{ik}(\tilde{y}_{ikt} - \tilde{y}_{ikt-1}) \\
& + \sum_t \sum_k \sum_{l \in \mathcal{L}'} e_{lk}(\tilde{z}_{lkt} - \tilde{z}_{lkt-1}) && \text{(4a)} \\
\leq& \sum_t \sum_k \sum_{i \in \mathcal{I}'} b_{ik}(\tilde{y}_{ikt}+\varepsilon)\ln \tfrac{\tilde{y}_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon} \\
& + \sum_t \sum_k \sum_{l \in \mathcal{L}'} e_{lk}(\tilde{z}_{lkt}+\varepsilon)\ln \tfrac{\tilde{z}_{lkt}+\varepsilon}{\tilde{z}_{lkt-1}+\varepsilon} && \text{(4b)} \\
\leq& \eta \sum_t \sum_k (\sum_{i \in \mathcal{I}'} \tfrac{b_{ik}}{\sigma}\ln \tfrac{\tilde{y}_{ikt}+\varepsilon}{\tilde{y}_{ikt-1}+\varepsilon} + \sum_{l \in \mathcal{L}'} \tfrac{e_{lk}}{\sigma}\ln \tfrac{\tilde{z}_{lkt}+\varepsilon}{\tilde{z}_{lkt-1}+\varepsilon}) \\
& && \text{(4c)} \\
=& \eta \sum_t \sum_k (\sum_{i \in \mathcal{I}'}(\alpha_{kt} - \beta_{ikt} - \mu_{kt} + \sum_k \mu_{kt} - a_{ikt})
\end{aligned}$$

$$
\begin{aligned}
&+\textstyle\sum_{l\in\mathcal{L}'}(\alpha_{kt}-\mu_{kt}+\sum_k\mu_{kt}+\sum_i\gamma_{it}-d_{lkt})) && \text{(4d)}\\
\le&\eta\textstyle\sum_t\sum_k(\sum_{i\in\mathcal{I}'}(\alpha_{kt}+\sum_k\mu_{kt})\\
&+\textstyle\sum_{l\in\mathcal{L}'}(\alpha_{kt}+\sum_k\mu_{kt}+\sum_i\gamma_{it})) && \text{(4e)}\\
\le&2\eta\textstyle\sum_t\sum_k(\alpha_{kt}+\sum_k\mu_{kt}+\sum_i r_{it}) && \text{(4f)}\\
\le&2\eta|\mathcal{K}|(\textstyle\sum_t\sum_k\alpha_{kt}\lambda_{kt}+\sum_t\sum_k\mu_{kt}(\sum_k\lambda_{kt}-1)\\
&+\textstyle\sum_t\sum_i\gamma_{it}(\sum_k\lambda_{kt}-1)) && \text{(4g)}\\
\le&2\eta|\mathcal{K}|D. && \text{(4h)}
\end{aligned}
$$

(4a) is given by the definition of $\mathcal{I}'$ and $\mathcal{L}'$. (4b) holds due to the facts shown in the poof of Lemma 1. (4c) holds because of $\tilde{y}_{ikt},\tilde{z}_{lkt}\le 1$. (4d) follows from the KKT conditions (2b) and (2c). And (4e) holds by removing the negative terms. (4g) and (4h) hold since $\sum_k\lambda_{kt}\ge 1$ and $\sum_k\lambda_{kt}-1\ge 1$. Additionally, these two inequalities hold because $\lambda_{kt},\forall k,\forall t$ is binary. The sum of λ_{kt} over all k must be no less than 1; also, note that if $\sum_k\lambda_{kt}-1\le 0$ is the case, D changes correspondingly, and (4g) and (4h) still hold for the new D.

C. Proof of Lemma 3

$$
\begin{aligned}
&\textstyle\sum_t\sum_i\sum_k f_{ikt}\bar{x}_{it}\\
\le&\textstyle\sum_t(\sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t}\sum_k f_{ikt}\tilde{x}_{it}+\sum_{i\in\mathcal{I}'_t}\sum_k f_{ikt}) && \text{(5a)}\\
\le&\textstyle\sum_t(\sum_{i\in\mathcal{I}}\sum_k f_{ikt}\tilde{x}_{it}+\max_{i\in\mathcal{I}}\sum_k f_{ikt}) && \text{(5b)}\\
\le&\textstyle\sum_t(\sum_i\sum_k f_{ikt}\tilde{x}_{it}+\sum_k\lambda_{kt}) && \text{(5c)}\\
\le&\textstyle\sum_t(\sum_i\sum_k f_{ikt}\tilde{x}_{it}+\sum_k(\sum_i\tilde{y}_{ikt}+\sum_l\tilde{z}_{lkt})) && \text{(5d)}\\
\le&\textstyle\sum_t(\sum_i\sum_k f_{ikt}\tilde{x}_{it}+\sum_k(\sum_i f_{ikt}\tilde{x}_{it}+\sum_l\tilde{z}_{lkt})) && \text{(5e)}\\
=&\textstyle\sum_t\sum_i\sum_k\frac{2f_{ikt}}{c_{it}}c_{it}\tilde{x}_{it}+\sum_t\sum_k\sum_l\frac{1}{d_{lkt}}d_{lkt}\tilde{z}_{lkt} && \text{(5f)}\\
\le&\Big(\max_{t,i,k}\frac{2f_{ikt}}{c_{it}}+\max_{t,k,l}\frac{1}{d_{lkt}}\Big)P'(\{\tilde{\mathbf{x}}_\mathbf{t},\tilde{\mathbf{y}}_\mathbf{t},\tilde{\mathbf{z}}_\mathbf{t},\forall t\}). && \text{(5g)}
\end{aligned}
$$

(5a) follows from Algorithm 2. After executing Algorithm 2 to round $\bar{\mathbf{x}}_\mathbf{t}$, if $\mathcal{I}'_t=\emptyset$, we can still reach (5g); if $\mathcal{I}'_t\ne\emptyset$, we reach (5b), as $|\mathcal{I}'_t|=1$. Next, we reach (5c) because the number of flows covered by any bid cannot exceed the total number of flows in the system. We continue with (5d) due to Constraint (1b), and further with (5e) due to Constraint (1a). (5f) is by some simple algebra, and (5g) is by the definition of the objective function P'.

D. Proof of Theorem 2

There are five terms summed up in $E(P'(\{\bar{\mathbf{x}}_\mathbf{t},\bar{\mathbf{y}}_\mathbf{t},\bar{\mathbf{z}}_\mathbf{t},\forall t\}))$. In the following, we treat them separately. Firstly, let us start with $\sum_t\sum_i c_{it}\bar{x}_{it}$:

$$
\begin{aligned}
&\textstyle\sum_t\sum_i c_{it}\bar{x}_{it}\\
=&\textstyle\sum_t\sum_i\sum_k f_{ikt}\bar{x}_{it}\frac{c_{it}}{\sum_k f_{ikt}} && \text{(6a)}\\
\le&\textstyle\max_{t,i}\frac{c_{it}}{\sum_k f_{ikt}}\sum_t\sum_i\sum_k f_{ikt}\bar{x}_{it} && \text{(6b)}\\
\le&\delta_x P'(\{\tilde{\mathbf{x}}_\mathbf{t},\tilde{\mathbf{y}}_\mathbf{t},\tilde{\mathbf{z}}_\mathbf{t},\forall t\}). && \text{(6c)}
\end{aligned}
$$

(6a) and (6b) follow from some simple transformation; (6c) is by Lemma 3. The above applies to any possible value $\bar{x}_{it}$ takes, the expectation of the left-hand side is thus no greater than the right-hand side: $E(\sum_t\sum_i c_{it}\bar{x}_{it})\le\delta_x P'$.

Then, as $\mathbf{y}^*_\mathbf{t}$ and $\mathbf{z}^*_\mathbf{t}$ are rounded together, let us consider $\sum_t\sum_i\sum_k a_{ikt}\bar{y}_{ikt}+\sum_t\sum_l\sum_k d_{lkt}\bar{z}_{lkt}$:

$$
\begin{aligned}
&E\left(\textstyle\sum_t\sum_i\sum_k a_{ikt}\bar{y}_{ikt}+\sum_t\sum_l\sum_k d_{lkt}\bar{z}_{lkt}\right)\\
\le&\delta'_{yz}\textstyle\sum_t\sum_k E(\sum_i\bar{y}_{ikt}+\sum_l\bar{z}_{lkt}) && \text{(7a)}\\
=&\delta'_{yz}\textstyle\sum_t\sum_k\left(\sum_{i\in\mathcal{I}\setminus\mathcal{I}'_t}y^*_{ikt}+\sum_{l\in\mathcal{I}\setminus\mathcal{I}'_t}z^*_{lkt}+\sum_{i\in\mathcal{I}'_t}1\right) && \text{(7b)}\\
\le&2\delta'_{yz}|\mathcal{K}|\textstyle\sum_t\sum_k\lambda_k && \text{(7c)}\\
\le&2\delta'_{yz}|\mathcal{K}|\textstyle\sum_t\sum_k(\sum_i\tilde{y}_{ikt}+\sum_l\tilde{z}_{lkt}) && \text{(7d)}\\
=&2\delta'_{yz}|\mathcal{K}|\textstyle\sum_t\sum_k(\sum_i\frac{1}{a_{ikt}}a_{ikt}\tilde{y}_{ikt}+\sum_l\frac{1}{d_{lkt}}d_{lkt}\tilde{z}_{lkt}) && \text{(7e)}\\
\le&\delta_{yz}P'(\{\tilde{\mathbf{x}}_\mathbf{t},\tilde{\mathbf{y}}_\mathbf{t},\tilde{\mathbf{z}}_\mathbf{t},\forall t\}), && \text{(7f)}
\end{aligned}
$$

where $\delta'_{yz}=\max_{t,i,k}a_{ikt}+\max_{t,l,k}d_{lkt}$. (7a) takes the coefficients out of the summations. (7b) follows from Algorithm 2 if $\mathcal{I}'_t\ne\emptyset$. If $\mathcal{I}'_t=\emptyset$, we can still reach (7f). We now use $\mathcal{I}$ to denote $\mathcal{I}\cup\mathcal{L}$. (7c) and (7d) use Constraints (1c) and (1b), respectively. (7e) and (7f) make rearrangments.

Finally, we briefly exhibit the results for the switching cost:

$$
\begin{aligned}
&E\Big(\textstyle\sum_t\sum_i\sum_k b_{ik}(\bar{y}_{ikt}-\bar{y}_{ikt-1})^+\\
&+\textstyle\sum_t\sum_l\sum_k e_{lk}(\bar{z}_{lkt}-\bar{z}_{lkt-1})^+\Big)\\
\le&E(\textstyle\sum_t\sum_i\sum_k b_{ik}\bar{y}_{ikt}+\sum_t\sum_l\sum_k e_{lk}\bar{z}_{lkt}) && \text{(8a)}\\
\le&\delta_{wv}P'(\{\tilde{\mathbf{x}}_\mathbf{t},\tilde{\mathbf{y}}_\mathbf{t},\tilde{\mathbf{z}}_\mathbf{t},\forall t\}). && \text{(8b)}
\end{aligned}
$$

(8a) uses the definition of the function $(\cdot)^+$. (8b) follows from an analogous process as (7a) through (7f).

E. Proof of Theorem 3

Firstly, we prove that $E(\bar{x}_{it})$ is monotonically nonincreasing in $c_{it},\forall i\in\mathcal{I}$. For any t, let $\mathcal{C}(\tilde{\mathbf{x}},c_i,\mathbf{c}_{-i})$ denote the objective function value of $\tilde{\mathbf{P}}_\mathbf{t}$ with bidding prices $\{c_i,\mathbf{c}_{-i}\}$ of all the bids and the optimal fractional solution $\tilde{\mathbf{x}}$. We fix $\mathbf{c}_{-i}$. We let $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ denote the optimal fractional solution when the bid i bids the price of c_i and c'_i. We assume $c_i\ge c'_i$, and have

$$\mathcal{C}(\tilde{\mathbf{x}},c_i,\mathbf{c}_{-i})\le\mathcal{C}(\tilde{\mathbf{x}}',c_i,\mathbf{c}_{-i}),\mathcal{C}(\tilde{\mathbf{x}}',c'_i,\mathbf{c}_{-i})\le\mathcal{C}(\tilde{\mathbf{x}},c'_i,\mathbf{c}_{-i}).$$

Adding them together and rearranging the terms leads to

$$
\begin{aligned}
\mathcal{C}(\tilde{\mathbf{x}},c_i,\mathbf{c}_{-i})+\mathcal{C}(\tilde{\mathbf{x}}',c'_i,\mathbf{c}_{-i})\le\mathcal{C}(\tilde{\mathbf{x}}',c_i,\mathbf{c}_{-i})+\mathcal{C}(\tilde{\mathbf{x}},c'_i,\mathbf{c}_{-i}),\\
\mathcal{C}(\tilde{\mathbf{x}},c_i,\mathbf{c}_{-i})-\mathcal{C}(\tilde{\mathbf{x}},c'_i,\mathbf{c}_{-i})\le\mathcal{C}(\tilde{\mathbf{x}}',c_i,\mathbf{c}_{-i})-\mathcal{C}(\tilde{\mathbf{x}}',c'_i,\mathbf{c}_{-i}),
\end{aligned}
$$

which actually means $\tilde{x}_{it}(c_i-c'_i)\le\tilde{x}'_{it}(c_i-c'_i)$, following the definition of $\tilde{\mathbf{P}}_\mathbf{t}$. Based on $c_i\ge c'_i$, removing $c_i-c'_i$, we finally have $\tilde{x}_{it}\le\tilde{x}'_{it}$, i.e., $E(\bar{x}_{it})\le E(\bar{x}'_{it})$, $\forall i\in\mathcal{I}\setminus\mathcal{I}'_t$. If $i\in\mathcal{I}'_t$, we still have $E(\bar{x}_{it})\le E(\bar{x}'_{it})$, since $E(\bar{x}_{it})=E(\bar{x}'_{it})=1$.

Secondly, as ζ works as the upper bound of the integral of $\int_0^\infty E(\bar{x}_{it}(c,\mathbf{c}_{-it}))dc$, we have the following, $\forall i\in\mathcal{I}\setminus\mathcal{I}'_t$:

$$
\begin{aligned}
\textstyle\int_0^\infty E(\bar{x}_{it}(c,\mathbf{c}_{-it}))dc&=\textstyle\int_0^\zeta E(\bar{x}_{it}(c,\mathbf{c}_{-it}))dc\\
&=\textstyle\int_0^{\sum_l\sum_k f_{ikt}(d_{lkt}+e_{lk})}\tilde{x}_{it}(c,\mathbf{c}_{-it})dc\\
&\le\textstyle\sum_l\sum_k f_{ikt}(d_{lkt}+e_{lk})\\
&<\infty.
\end{aligned}
$$

We have a similar conclusion if $i\in\mathcal{I}'_t$.

Thirdly, we note that our payment already has form of $\rho_i=c_{it}E(\bar{x}_{it}(c_{it},\mathbf{c}_{-it}))+\int_{c_{it}}^\infty E(\bar{x}_{it}(c,\mathbf{c}_{-it}))dc,\forall i\in\mathcal{I}$.

Finally, we prove the individual rationality in expectation. Using the payment ρ_i, we have the expected utility of

$$u_i=\rho_i-c_{it}E(\bar{x}_{it}(c_{it},\mathbf{c}_{-it}))=\textstyle\int_{c_{it}}^\infty E(\bar{x}_{it}(c,\mathbf{c}_{-it}))dc\ge 0,$$

if the bid i wins in the auction. If the bid i loses, by definition we have $u_i=0$. Joining the two cases, we always have $u_i\ge 0$.

REFERENCES

[1] "AT&T DDoS Defense - DDoS Protection & Mitigation Service," https://www.business.att.com/products/ddos-protection.html.

[2] M. Jonker, A. Sperotto, R. van Rijswijk-Deij, R. Sadre, and A. Pras, "Measuring the adoption of ddos protection services," in *ACM IMC*, 2016.

[3] "DDoS Protection Service | Anti DDoS Mitigation | Cloudflare," https://www.cloudflare.com/ddos/.

[4] P. Zilberman, R. Puzis, and Y. Elovici, "On network footprint of traffic inspection and filtering at global scrubbing centers," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 5, pp. 521–534, 2017.

[5] D. Migault, M. A. Simplicio, B. M. Barros, M. Pourzandi, T. R. Almeida, E. R. Andrade, and T. C. Carvalho, "A framework for enabling security services collaboration across multiple domains," in *IEEE ICDCS*, 2017.

[6] N. C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: a survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 954–1001, 2017.

[7] W. Shi, L. Zhang, C. Wu, Z. Li, F. Lau, W. Shi, L. Zhang, C. Wu, Z. Li, and F. Lau, "An online auction framework for dynamic resource provisioning in cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2060–2073, 2016.

[8] Y. Zhu, S. D. Fu, J. Liu, and Y. Cui, "Truthful online auction toward maximized instance utilization in the cloud," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2132–2145, 2018.

[9] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. C. Lau, "Online auctions in iaas clouds: Welfare and profit maximization with server costs," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 1034–1047, 2017.

[10] R. Zhou, Z. Li, C. Wu, and Z. Huang, "An efficient cloud market mechanism for computing jobs with soft deadlines," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 793–805, 2017.

[11] C. Dietzel, G. Smaragdakis, M. Wichtlhuber, and A. Feldmann, "Stellar: network attack mitigation using advanced blackholing," in *ACM CoNEXT*, 2018.

[12] "BGP Cost," http://bill.herrin.us/network/bgpcost.html.

[13] T. Holterbach, S. Vissicchio, A. Dainotti, and L. Vanbever, "Swift: Predictive fast reroute," in *ACM SIGCOMM*, 2017.

[14] L. Jiao, R. Zhou, X. Lin, and X. Chen, "Online scheduling of traffic diversion and cloud scrubbing with uncertainty in current inputs," in *ACM MOBIHOC*, 2019.

[15] Z. Liu, Y. Cao, M. Zhu, and W. Ge, "Umbrella: Enabling isps to offer readily deployable and privacy-preserving ddos prevention services," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1098–1108, 2019.

[16] L. Jin, S. Hao, H. Wang, and C. Cotton, "Your remnant tells secret: residual resolution in ddos protection services," in *IEEE/IFIP DSN*, 2018.

[17] 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.

[18] 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.

[19] S. Chen, L. Jiao, L. Wang, and F. Liu, "An online market mechanism for edge emergency demand response via cloudlet control," in *IEEE INFOCOM*, 2019.

[20] N. Buchbinder, S. Chen, and J. S. Naor, "Competitive analysis via regularization," in *ACM-SIAM SODA*, 2014.

[21] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *Journal of the ACM*, vol. 53, no. 3, pp. 324–360, 2006.

[22] A. Archer and É. Tardos, "Truthful mechanisms for one-parameter agents," in *IEEE FOCS*, 2001.

[23] "Amazon EC2 Pricing," https://aws.amazon.com/ec2/pricing/.

[24] H. B. J. G. Asfandyar Qureshi, Rick Weber and B. Maggs, "Cutting the electric bill for internet-scale systems," in *ACM SIGCOMM*, 2009.

[25] "Gurobi - The fastest solver," https://www.gurobi.com/.

[26] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1378–1391, 2013.

[27] R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips, "Strengthening integrality gaps for capacitated network design and covering problems," in *ACM-SIAM SODA*, 2000.

[28] A. A. Ageev and M. I. Sviridenko, "Pipage rounding: A new method of constructing algorithms with proven performance guarantee," *Journal of Combinatorial Optimization*, vol. 8, no. 3, pp. 307–328, 2004.

[29] "AMPL | Streamlined Modeling for Real Optimization," https://ampl.com/.

[30] "COIN-OR Interior Point Optimizer IPOPT," https://www.coin-or.org/Ipopt/.