A Shapley-value Mechanism for Bandwidth On Demand between Datacenters

Weijie Shi, *Student Member, IEEE,* Chuan Wu, *Member, IEEE,* and Zongpeng Li, *Member, IEEE*

Abstract—Recent studies in cloud resource allocation and pricing have focused on computing and storage resources but not network bandwidth. Cloud users nowadays customarily deploy services across multiple geo-distributed datacenters, with significant inter-datacenter traffic generated, paid by cloud providers to ISPs. An effective bandwidth allocation and charging mechanism is needed between the cloud provider and the cloud users. Existing volume based static charging schemes lack market efficiency. This work presents the first dynamic pricing mechanism for inter-datacenter on-demand bandwidth, via a Shapley value based auction. Our auction is expressive enough to accept bids as a flat bandwidth rate plus a time duration, or a data volume with a transfer deadline. We start with an offline auction, design an optimal end-to-end traffic scheduling approach, and exploit the Shapley value in computing payments. Our auction is truthful, individual rational, budget balanced and approximately efficient in social welfare. An online version of the auction follows, where decisions are made instantly upon the arrival of each user's realtime transmission demand. We propose an efficient online traffic scheduling algorithm, and approximate the offline Shapley value based payments on the fly. We validate our mechanism design with solid theoretical analysis, as well as trace-driven simulation studies.

Index Terms—Dynamic Bandwidth Pricing, Auction Mechanism Design, Shapley Value

◆

1 INTRODUCTION

Today's Internet sees the proliferation of on-demand cloud computing services, which cloud providers provision from multiple datacenters in different geographic locations, *e.g.*, Amazon, Google and Rackspace [1]. Cloud users host their tasks or services on the cloud platform, often in multiple datacenters, to stay close to service users, exploit lower power costs in different regions, and enable service robustness in the face of network/power failures. Substantial inter-datacenter transfers have hence become the norm rather than abnormality, for migration of virtual machines [2], replication of contents like videos [3], and backup of data [4]. The bandwidth needed does not come for free from the cloud provider: except for a few large operators (*e.g.*, Google) which own the network among their datacenters, cloud providers typically rent the link between their datacenters from ISPs and are charged for their inter-datacenter traffic [5]. A proper charging mechanism is needed to share the bandwidth cost with the cloud users.

Existing studies have substantially investigated allocation and pricing mechanisms for cloud resources like computation and storage [6][7][8], but see a lack of proposals for bandwidth pricing in cloud computing. The current common practice is to charge a fixed flat rate on data volume, *e.g.*, per GB by Amazon EC2 [9].

- *Weijie Shi is with the Department of Computer Science, the University of Hong Kong. E-mail: wjshi@cs.hku.hk*
- *Chuan Wu is with the Department of Computer Science, the University of Hong Kong. E-mail: cwu@cs.hku.hk*
- *Zongpeng Li is with the Department of Computer Science, University of Calgary. E-mail: zongpeng@ucalgary.ca*

Such a flat charging model inherently lacks of market efficiency to adapt to realtime demand/supply changes, jeopardizing efficient bandwidth utilization as well as social welfare of the cloud provider and cloud users.

This paper presents the first dynamic pricing mechanism for on-the-fly requested inter-datacenter bandwidth to the authors' knowledge. We resort to an auction based mechanism, for effectively discovering the market value of the link bandwidth. The users with data transfer needs submit bids to the provider, specifying their bandwidth demands, willing-to-pay prices, and source-destination pairs of the transfer; the cloud provider decides admission control, the transfer schedule of users' data and the charges. The following properties are mandatory through this auction: (1) *Truthfulness* and (2) *Individual Rationality*, the quintessential properties which stimulate participation and voluntary revelation of valuation in auctions; (3) *Competitiveness in Social Welfare*, which is evaluated by the upper bound of the ratio between the optimal social welfare and the social welfare under the proposed algorithm, and quantifies the efficiency of the auction mechanism; (4) *Budget Balance at the Cloud Provider*, such that the payments from the users are sufficient to cover the ISP charge.

We identify the following key challenges in auction design to achieve the above properties. *First*, there are different types of bandwidth demands, *e.g.*, a flat bandwidth rate plus a duration, or an amount of data to be sent within a completion deadline; with one-hop connection or through a linked path. How can we design one unified auction framework taking care of all this variety? *Second*, the auction's underlying social welfare maximization problem is an NP-hard combinatorial problem, even with decisions on admission control and

traffic scheduling only. There is one more dimension in our decision space: the overall bandwidth to rent from the ISPs on each inter-datacenter link. Different ISP bandwidth charging models exist, *e.g.*, the 95th-percentile or the maximum-traffic charging models [10]. It is computationally difficult to decide both bandwidth capacities to rent from the ISPs and allocation of these capacities to users; the hardness escalates when this optimization appears as the winner determination problem in an auction, which renders the auction a similar flavor to double auctions, shown to be very difficult to simultaneously achieve all 4 properties listed above [11]. The challenge is already daunting even for the offline version of the auction, where the cloud provider is assumed to have complete knowledge of task arrivals in the whole system span. However, *third*, an online auction is needed in practice, to instantly make decisions upon the arrival of each user's real-time transmission demand. Our contributions in tackling the above challenges are summarized as follows:

▷ We set up an auction framework that is expressive enough to allow cloud users to specify their bandwidth demands and transmission paths, and design an efficient mechanism for traffic scheduling and charging.

▷ We investigate both an offline version and an online version of the auction, considering both the 95th-percentile and the maximum-traffic charging models of ISPs. A number of non-trivial, intriguing theoretical results are obtained.

▷ For the offline version of the auction, we design an optimal end-to-end traffic scheduling algorithm, and novelly exploit the Shapley value in deciding admission control and the payments, based on a cost sharing idea. The Shapley value has been widely used for allocating an aggregate profit/cost among the individual participants. But it has rarely been exploited for payments in auction mechanism design. We identify that calculating users' payments based on the Shapley values is more reasonable than using a classical VCG mechanism in our system, which together with the traffic scheduling algorithm, is proven to achieve all 4 desired properties and a $\delta/(\delta - \gamma)$-approximation in social welfare, where δ and γ are constants related to users' valuations.

▷ For the online version of the auction, we propose an efficient traffic scheduling algorithm, and approximate the offline Shapley value based admission control and payments on the fly. In particular, we compute an online Shapley value based on requests that have arrived, scale it according to the passed time, the total time span, the ISP charge due to the existing traffic and estimated total ISP charge. The online auction achieves all the desired properties in expectation, with $\delta/(\delta - \gamma)$-competitiveness in social welfare as well.

This theoretical work sits upon the state-of-the-art development in datacenter networks, with link multiplexing and virtual datacenter networking [12]. Especially, the recent software defined networking (SDN) paradigm enables a centralized controller to manage the entire network, with typical applications to inter-datacenter networks. Our centralized auctioneer fits nicely as a functionality module implemented on the controller, to overview traffic scheduling and charging in the system.

In the rest of the paper, we discuss related work in Sec. 2, and define the system model in Sec. 3. Sec. 4 and Sec. 5 present the offline auction and the online auction, respectively. Simulation results are presented in Sec. 6. And extension of the online auction is presented in Sec. 7 Sec. 8 concludes the paper.

2 RELATED WORK

Resource allocation and pricing in cloud computing have attracted substantial research interests in recent years (e.g., [13][14][15][16][17][18] [19][20][21][22]). Sossa *et al.* [14] study scheduling algorithms for deadline-based workflows on clouds. Hong *et al.* [15] solve the optimization problem of VM placement. The biggest differences of our work from them are to use a dynamic pricing mechanism to adapt to the dynamic market situation, and to incentivize users to follow the designed mechanism based on truthfulness guarantee. Due to their efficiency in discovering the market value of resources, auction mechanisms have been widely exploited for resource provisioning with either fixed VM types [23] or dynamically assembled VMs [7], in static systems [6] or in an online setting [8]. All these works focus on allocating computational resources of VMs such as CPU, disk and RAM, but do not deal with network bandwidth. There have been a few recent studies on bandwidth allocation and pricing in cloud computing [16][12]. But they focus on the network inside a datacenter, rather than the Internet links connecting different datacenters, where the role played by ISP charging brings another dimension of difficulty.

The major focus of auction mechanism design is on archiving truthfulness and efficiency. The celebrated VCG mechanism [24] is essentially the only type of auction that simultaneously guarantees truthfulness and optimal economic efficiency, through calculating the optimal allocation and charging bidders the opportunity cost. However, when the underlying allocation problem is NP-hard, which is the case in our paper and is common for combinatorial auctions [25], VCG becomes computationally infeasible. When approximation algorithms are applied to solving the underlying allocation problem, VCG loses its truthfulness [26]. A payment rule should be carefully designed to work in concert with the approximation algorithm to achieve truthfulness, e.g., by exploiting *critical bids* [27], or resorting to the LP decomposition technique [28], which is applicable to packing problems. We novelly apply the Shapley value method as the payment rule, which is suitable for our bandwidth allocation scenario due to the symmetry and fairness of Shapley values.

Some older literature has investigated bandwidth scheduling and pricing in the Internet using game theoretical approaches [29][30][31]. MacKie-Mason *et al.*

[32] apply a VCG auction on each router to identify a market price for each data packet. Maille *et al.* [33] enable a disparate second-price auction on each link of a tree network. Lazar *et al.* [34] study a progressive second price (PSP) auction. All these auction designs are set up in a static network with fixed bidders and persistent bandwidth demands. Differently, our auction model handles both demands for fixed bandwidth rates and data transfer with heterogeneous deadlines, in both offline and online settings. We also combine the auction mechanism analysis with the Shapley value method.

Shapley value [35] was originally proposed as an approach to allocate aggregated profit/cost to individuals in a coalition in a fair and efficient manner. It has been widely used in a variety of practical problems, including ISP settlement and analysis of ISP charging models [36][37].

Feigenbaum *et al.* [38] compare the Shapley value with the marginal cost in cost-sharing algorithms for multicast transmission. Grag *et al.* [39] propose a distributed mechanism to implement Shapley value cost-sharing. Though a common approach in cooperative game theory, it has seldom been applied to auction mechanism design. The only proposed auction using Shapley value is the Moulin Mechanism [40], used for cost sharing in public good auction. It rejects users whose valuation is lower than its Shapley value, re-calculates the aggregated cost and the Shapley value based on the remaining users, and repeats this rejection and calculation process until no more users are rejected. The approximation ratio of the Moulin mechanism in social welfare is proved to be $O(\log N)$ where N is the number of bidders [40], and this ratio is quite large but tight. Truthfulness of the Moulin mechanism relies on a *submodular* property of the cost function used in Shapley value computation, which means that the marginal cost of a user is non-increasing when new users are added to the game. The ISP charge function in our model, either the 95th-percentile or max-traffic charge, is not submodular. Instead, we design a novel Shapley value based mechanism which overcomes the drawbacks of the Moulin Mechanism.

3 SYSTEM MODEL

3.1 Network and Auction

Consider a cloud spanning multiple datacenters connected by L links, operated by one cloud provider. The topology of the network is a directed graph including these datacenters and links (see Fig. 1 in [41] for an example topology). The links are owned by an Internet Service Provider (ISP) and leased to the cloud provider. Throughout the paper, we use $[X]$ to denote the set of integers $\{1, 2, \ldots, X\}$. Let C_l denote the maximum bandwidth capacity that the ISP can provision on link $l \in [L]$. The ISP charges the cloud provider based on link usage periodically, *i.e.*, once in each accounting period. An accounting period is divided into multiple time slots. Let T be the number of time slots in each accounting period.
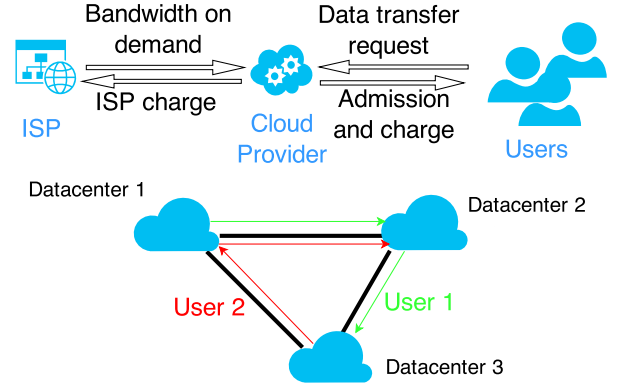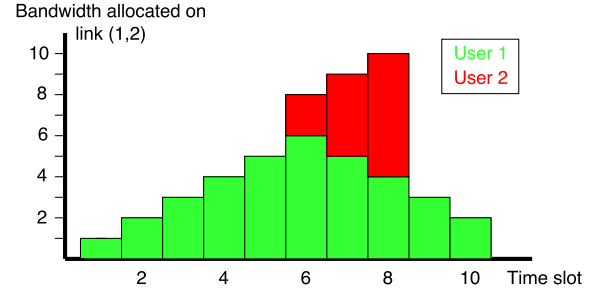

Fig. 1. An overview of the system.

Fig. 2. Example bandwidth allocation on link (1, 2) of Fig. 1.

Totally N users arrive over time, with requests for data transfer among the datacenters. User $n \in [N]$ arrives at time t_n, wishes to send data of size B_n (*e.g.*, bytes) from a source datacenter to a destination datacenter, along a network path P_n, within d_n time slots. Here P_n consists of multiple ordered links in $[L]$ such that $P_n \subseteq L$. The deadline of user n's data transfer is $t_n + d_n - 1$. User n has a secret valuation e_n for the task (*e.g.*, according to urgency and importance of the data transfer), and also submits a willingness-to-pay price, $\hat{e}_n$, for finishing her task in time. Later we will prove that it is her best strategy to bid a price equal to her valuation: $\hat{e}_n = e_n$. In summary, the bid submitted by user n is a three-tuple $(B_n, d_n, \hat{e}_n)$. The case that a user wishes to send data between multiple pairs of source/destination datacenters can be treated as multiple separate bids in our model. In addition, our bid model can also describe a user n's request for an exclusive bandwidth rate along a path for d_n consecutive time slots, where the bandwidth rate is B_n/d_n.

The provider, as the auctioneer, makes the following decisions: (1) whether to accept or to reject user n's transfer request, represented by a binary decision variable $x_n = 1$ or 0, respectively; (2) the amount of user n's data to transmit along her specified path in each time slot within the transmission deadline, $b_n(t), \forall t \in [t_n, t_n + d_n - 1]$ (we assume $b_n(t) = 0$ for $t \notin [t_n, t_n + d_n - 1]$); (3) the payment that the provider charges user n for finishing her transmission, j_n. In the case that user n requests an exclusive bandwidth rate, the transmission schedule decisions are fixed such that $b_n(t_n) = b_n(t_n + 1) = \cdots =$

$b_n(t_n + d_n - 1) = B_n/d_n$, and the decisions of the cloud provider simplify to x_n and j_n only. The utility of user n is the difference between her true valuation e_n and her payment j_n: $u_n = x_n e_n - j_n$. Denote the ISP charge function by $v(\boldsymbol{a})$, where $\boldsymbol{a} = \{a_l(t), \forall l \in [L], \forall t \in [T]\}$ is the vector of link traffic in the entire network, with $a_l(t)$ being the total traffic of the cloud on link l at time t.

Fig. 1 shows an example cloud spanning three geo-distributed datacenters, as well as the interaction between the ISP, the cloud provider and users in our system. Two example users are illustrated: user 1 arrives at time 1, requesting to send data at the volume of $B_1 = 35$ from datacenter 1 to datacenter 2 and then to datacenter 3, within $d_1 = 10$ time slots; user 2 arrives at time 6, requesting to transmit data $B_2 = 12$ from datacenter 3 to datacenter 1 and then to datacenter 2 within $d_2 = 3$. Fig. 2 shows an example bandwidth allocation by the cloud provider on link (1,2) over time, *e.g.*, the data transmission for user 2 (in red) is scheduled to happen in time slots 6, 7 and 8 with data volume $b_2(6) = 2, b_2(7) = 4$ and $b_2(8) = 6$, respectively.

The auction targets maximization of the social welfare, which is the sum of the cloud provider's revenue, *i.e.*, aggregate payment from the users minus link bandwidth charge paid to the ISP, $\sum_{n \in [N]} j_n - v(\boldsymbol{a})$, and utilities of all the users, $\sum_{n \in [N]} (x_n e_n - j_n)$. Since payments from the users and those received at the provider cancel each other out, the social welfare equals the total valuation of accepted users (users with $x_n = 1$) minus the overall ISP charge, $\sum_{n \in [N]} x_n e_n - v(\boldsymbol{a})$.

In our system, we assume that the intermediate datacenters along an end-to-end transmission path P_n do not store the data, but immediately forward the data received to the next hop datacenter. We ignore the end-to-end delay along path P_n, since the propagation delay even along a cross-continent link is usually less than several seconds, while the typical time slot length used in ISP bandwidth accounting is 5 minutes [42]. Hence, the amount of user n's traffic on any link l along her path P_n is the same $b_n(t)$ in a given time slot t, and the total traffic on link l at t can be computed as $a_l(t) = \sum_{n \in \widetilde{P}_l} b_n(t)$, where $\widetilde{P}_l$ represents the set of users whose paths traverse link l, such that $n \in \widetilde{P}_l$ if and only if $l \in P_n$. The social welfare optimization problem, as the winner determination problem that the cloud provider solves to produce the auction outcome, is formulated as follows (assuming truthful bidding):

$$\max \sum_{n \in [N]} x_n e_n - v(\boldsymbol{a}) \tag{1}$$

$$\text{s.t.} \quad a_l(t) = \sum_{n \in \widetilde{P}_l} b_n(t) \quad \forall l \in [L], t \in [T] \tag{1a}$$

$$x_n B_n = \sum_{t=t_n}^{t_n + d_n - 1} b_n(t) \quad \forall n \in [N] \tag{1b}$$

$$a_l(t) \le C_l \quad \forall l \in [L], t \in [T] \tag{1c}$$

$$x_n \in \{0, 1\} \quad \forall n \in [N] \tag{1d}$$

$$b_n(t) \ge 0 \quad \forall n \in [N], t \in [T] \tag{1e}$$

(1b) specifies that if user n's task is admitted, her total

amount of data transmitted during t_n to $t_n + d_n - 1$ should be B_n. (1c) is the maximum link capacity constraint.

We define the ISP charge function $v(\boldsymbol{a})$ as the aggregated ISP charge incurred by traffic of all accepted users in the entire network. Next we also define an alternative form of $v(\cdot)$, which will be useful later. Let $v(\boldsymbol{b}, S)$ denote the aggregate ISP charge under the traffic schedule $\boldsymbol{b} = \{b_n(t), \forall n \in [S], t \in [T]\}$, incurred by a subset of users $S \subseteq [N]$, by assuming $b_n(t) = 0$ for all users $n \notin S$. Hence $v(\boldsymbol{a})$ is equivalent to $v(\boldsymbol{b}, [N])$. We may omit $\boldsymbol{b}$ when traffic schedule $\boldsymbol{b}$ is known and fixed, and denote the ISP charge function by $v(S)$.

Theorem 1. *The social welfare maximization problem (1) is NP-hard, even without the ISP charge ($v(\boldsymbol{a}) = 0$).*

The proof is given in Appendix A.

3.2 Target Properties

We seek to achieve the following critical properties in our auction mechanism design.

Definition 1. (Truthfulness in bidding price) *Any user n cannot increase her utility by bidding a price $\hat{e}_n$ different from her true valuation e_n, i.e., fixing other users' bids, and assuming that her data size B_n and deadline d_n are known to the auctioneer, user n's utility is maximized by bidding the truth valuation.*

Definition 2. (Individual rationality) *Any user's utility is non-negative, whether or not her request is admitted, i.e., $u_n \ge 0, \forall n \in [N]$.*

Definition 3. (Budget balance) *The revenue of the cloud provider, which is the total collected payment from users minus the overall ISP bandwidth charge, is non-negative, i.e., $u_P = \sum_{n \in [N]} j_n - v(\boldsymbol{a}) \ge 0$.*

Definition 4. (Competitiveness) *The auction is called c-competitive if for any arriving pattern of user requests, the ratio between the optimal social welfare, S_{opt}, the optimal objective value of (1), and the social welfare achieved by the auction mechanism, S_{auc}, is no larger than c, i.e., $\frac{S_{opt}}{S_{auc}} \le c$.*

3.3 ISP Charging Models

We consider the most prevalent bandwidth charging method adopted by many ISPs, the 95th-percentile charge [43], [37]. An accounting period, usually 1 month, is split into time slots, typically 5 minutes each, and the number of bytes transmitted in each time slot along a link l is recorded. The charge on link l is based on the 95th-percentile of these recorded volumes in an accounting period. Let α_l be the unit bandwidth price of link l (*e.g.*, \$/byte), and $\widetilde{a}_l(t), \forall t$, be the ranked sequence of $a_l(t), \forall t$, such that $\widetilde{a}_l(1) \ge \widetilde{a}_l(2) \ge \cdots \ge \widetilde{a}_l(T)$. Then the charge of link l is $v_{95}^{(l)}(\boldsymbol{a}_l) = \alpha_l \widetilde{a}_l(\lfloor 0.05T \rfloor + 1)$. The total ISP charge for all links is $v_{95}(\boldsymbol{a}) = \sum_{l \in [L]} v_{95}^{(l)}(\boldsymbol{a}_l)$.

We also investigate another ISP charging model, the maximum-traffic charge, which is a simplification of the

95th-percentile charge. The maximum volume among the recorded slots is charged, such that the charge of link l in an accounting period is $v_{max}^{(l)}(\boldsymbol{a}_l) = \alpha_l \widetilde{a}_l(1)$, and the total ISP charge is $v_{max}(\boldsymbol{a}) = \sum_{l \in [L]} v_{max}^{(l)}(\boldsymbol{a}_l)$.

For the example in Fig. 2 where two users share link (1,2), the maximum-traffic charge for the 10 time slots is based on the traffic at time slot 8, which is 10. The 95th-percentile charge is based on the data volume in the time slot with the second highest traffic, which is 9 at time slot 7.

TABLE 1
Key Notation

[X]	integer set $\{1, 2, \ldots, X\}$
N	# of users
T	# of time slots in an accounting period
L	# of links
C_l	maximum capacity of link l
B_n	user n's data size
P_n	the path of user n's transmission
d_n	user n's allowable delivery delay
t_n	user n's arriving time
x_n	0 or 1, user n is accepted or rejected
$b_n(t)$	traffic of user n sent at time t
$a_l(t)$	overall traffic on link l at time t
e_n	user n's valuation
$\hat{e}_n$	user n's bid price
j_n	user n's payment
u_n	user n's utility
u_P	cloud provider's revenue
$v(\boldsymbol{a})$	ISP charge under overall traffic $\boldsymbol{a} = \{a_l(t)\}$
$v(S)$	ISP charge with a subset S of users
S_{opt}	optimal social welfare
S_{auc}	social welfare under the auction
$\widetilde{P}_l$	set of users whose paths traverse l
ϕ_n	Shapley value of user n
δ	constant in Assumption 2
γ	constant in Assumption 3
D	maximum allowed delay of all data transfers
$\mathcal{N}$	set of users with $e_n \geq \gamma\phi_n$
α_l	unit bandwidth price of link l

3.4 Key Assumptions

We assume that all users' requests can be served, if the cloud provider is willing to pay the ISP any amount of charge. This implies that the maximum capacity of the links is sufficient for all the data transfers, which is reasonable, since the ISP may otherwise improve its infrastructure to accommodate the extra demand, which brings more revenue to the ISP under the 95th-percentile or the maximum-traffic charge model.

If all users' requests are accepted, the total valuation of all users should typically be much larger than the overall ISP's charge. Otherwise, the cloud provider has no incentive to rent the links for dispatching user's traffic in the first place.

In the following sections, we will design a mechanism to allocate the cloud provider's cost due to the ISP charge proportionally and fairly among the users. Suppose such a mechanism exists for now, where ϕ_n is user n's cost share, and the sum of users' shares equals the total ISP charge, i.e., $\sum_n \phi_n = v(\boldsymbol{a})$, if all the requests are accepted. We expect that the valuation e_n of most users is larger

than the respective proportional cost ϕ_n. The number of the rest users, whose valuations are smaller, should be small, and the sum of the cost shares of these users should be limited.

We formally formulate the above into the following assumptions on users' valuations and cost shares.

Assumption 1. *There exists a feasible solution of the social welfare maximization problem (1), which admits all users' transfer requests, i.e., $x_n = 1, \forall n \in [N]$.*

Assumption 2. *The total valuation of all users is at least δ times the overall ISP charge, i.e., $\sum_{n \in [N]} e_n \geq \delta \sum_{n \in [N]} \phi_n$, where δ is a positive constant.*

Assumption 3. *Let $\mathcal{N}$ denote the set of users whose valuation e_n is at least γ times her share of the ISP charge, ϕ_n, i.e., $e_n \geq \gamma\phi_n$. The aggregate cost share of all users not in $\mathcal{N}$ is no larger than $(1 - 1/\gamma)$ times the overall ISP charge, i.e., $\sum_{n \notin \mathcal{N}} \phi_n \leq (1 - 1/\gamma) \sum_{n \in [N]} \phi_n$, where $\gamma > 1$ is a constant. Let ϕ_n' be the online estimated ISP charge (will be introduced in Sec. 5), and let $\mathcal{N}' = \{n | e_n \geq \gamma\phi_n'\}$. Then $\mathcal{N}'$ has the same property: $\sum_{n \notin \mathcal{N}'} \phi_n \leq (1 - 1/\gamma) \sum_{n \in [N]} \phi_n$.*

Key notation is listed in Table 1 for ease of reference.

4 OFFLINE AUCTION

We first consider the offline scenario where all users' transmission requests (arrival times and detailed demands) are known within the whole time span, and design an efficient mechanism to decide whether to admit each user's request, how to schedule data transfer over time, and the payment of each user. It is natural to start with a VCG-like mechanism in the design, which is well known for achieving both social welfare maximization and truthfulness. The VCG auction [24] decides winners and resource allocation among winners by solving an underlying social welfare maximization problem, and charges each winner by their opportunity cost. The exact optimal solution to the social welfare maximization problem is typically required, in order to guarantee truthfulness of the VCG auction. In our model, the social welfare maximization problem is NP hard, where a VCG mechanism requires exponential running time and hence becomes computationally infeasible. Instead, we seek a novel truthful mechanism inspired by the cost sharing idea of the Shapley value. In what follows, we present the basics of the Shapley value first, and then our Shapley-value based auction design.

4.1 Shapley Value

The Shapley value originates from the cooperative game theory, which denotes the share of an individual in a coalition when the total surplus generated by the coalition is allocated among the individuals, following a uniquely designed distribution method [35]. The idea of the distribution method is to estimate the surplus contributed by an individual by averaging her marginal

contribution over all possible permutations of the individuals. A number of nice properties are achieved by the Shapley values. We adapt the Shapley value idea for cost sharing, by distributing the ISP's bandwidth charge that the users' traffic aggregately leads to, among the users. In our model, the Shapley value, *i.e.*, each user's cost share, when all uses are accepted (and hence share the ISP charge), is referred to as the *offline Shapley value*, and defined as follows.

Definition 5. *Assuming all users' transfer requests are admitted and the traffic schedule $\{b_n(t), \forall n \in [N], \forall t \in [T]\}$ is known, the offline Shapley value ϕ_n of user n is defined as the average marginal ISP charge incurred by user n's traffic*

$$\phi_n = \frac{1}{N!} \sum_{\pi \in \Pi_N} (v(S(\pi, n)) - v(S(\pi, n) \backslash n)),$$

where Π_N is the set of all $N!$ permutations of N users, $S(\pi, n)$ is the set of users that are no later than user n in permutation π, $v(S(\pi, n))$ is the ISP charge (under either the 95th-percentile or max-traffic charging method) supposing only requests from the subset $S(\pi, n)$ of users are served, and $v(S(\pi, n) \backslash n)$ is the ISP charge with one less user − user n.

Again we use the example in Fig. 2 to illustrate how to calculate the Shapley value of user 2. There are two permutations in this example: (i) [1,2], and (ii) [2,1]. In case (i), user 2's contribution to the ISP charge (under the maximum charge model) is 6 at time 8. In case 2, the ISP charge due to sending user 1's data only is 6; after including user 2's transmission, the ISP charge increases to 10, and hence the marginal ISP charge due to user 2 is 4. The Shapley value of user 2 is the average of the two cases, which equals 5.

Though an intuitive idea for cost sharing, the Shapley value has been proven to be the only distribution which achieves all the following three nice properties [44]. (1) *Efficiency*: the aggregate Shapley value of all users is guaranteed to be equal to the overall ISP charge, *i.e.*, $\sum_{n \in [N]} \phi_n = v([N])$. (2) *Symmetry*: if two users have exactly the same traffic pattern (source/destination, path, transfer schedule), their Shapley values should be equal. (3) *Fairness*: a user's influence on another user in Shapley value computation is symmetric, *i.e.*, $\forall n, m \in S$, $\phi_n(S) - \phi_n(S \backslash m) = \phi_m(S) - \phi_m(S \backslash n)$, $\forall S \subseteq [N]$, where $\phi_n(S)$ is the Shapley value computed when only traffic incurred by users in set S is considered.

Theoretically, calculating a Shapley value involves $O(N!)$ time complexity. However, it is proved that a sampling over possible permutations gives an efficient and unbiased estimator of the Shapley value [37]. We choose a polynomial number of permutations, Π, randomly from the set of all $N!$ permutations of users in $[N]$, and calculate the average marginal cost of each user over Π to obtain her estimated Shapley value.

$$\phi_n^\Pi = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} (v(S(\pi, n)) - v(S(\pi, n) \backslash n)) \qquad (2)$$

Theorem 2, which follows immediately from Proposition 2 in [37], shows that the estimated Shapley value still retains the efficiency property, if all users are computed using the same set of permutations Π.

Theorem 2. *If the estimated Shapley values in (2) for all users are computed using the same set of permutations Π, the sum of the estimate Shapley values of all users equals the total ISP charge,* i.e., $\sum_{n \in [N]} \phi_n^\Pi = v([N])$.

We compute the Shapley value based on (2) throughout the rest of the paper, but omit Π from the notation ϕ_n^Π.

4.2 Shapley Value Based Offline Auction

We design a Shapley value based admission control and payment mechanism, based on the idea of sharing the overall ISP charge, that the cloud provider is subject to, among all the users who bring the traffic on the links. The key idea of the mechanism is: if a user's bid price is no larger than her share of the overall ISP charge times a factor, then the user's transfer request should be rejected; otherwise, the user's request is accepted and she is charged at her share of the ISP charge times the factor.

The detailed auction design is given in Alg. 1. The algorithm decides admission control and payments of the users, assuming a given efficient traffic scheduling algorithm, which we will detail in Sec. 4.3. We first suppose all users' requests were accepted, *i.e.*, $x_n = 1, \forall n \in [N]$, and schedule the data transfer $\boldsymbol{b} = \{b_n(t), \forall n \in [N], \forall t \in T\}$ using the traffic scheduling algorithm. We compute the Shapley value ϕ_n for each user n according to this traffic schedule $\boldsymbol{b}$, and compare the user's bid price with $\gamma \phi_n$ to decide the acceptance/rejection and payment, where γ is a positive constant in Assumption 3 in Sec. 3.4. Finally, based on the admission control decisions $x_n, \forall n \in [N]$, we run the traffic scheduling algorithm in Sec. 4.3 again to re-schedule data transfer for all accepted users, to further optimize the traffic schedule for reduced ISP charge.

The payments in our auction are proportional to the respective Shapley values (with the same coefficient γ), such that they are fair among the users, according to the portions of overall ISP charge that their traffic incurs on the links that their data transfer traverses. It is also a reasonable method for the cloud provider to charge the users, that associates the payments with her cost (due to the ISP's charge). The parameter γ decides the revenue obtained by the cloud provider, and is tunable by her. It should be no smaller than 1, such that the overall payment from the users would be no lower than the ISP charge, in order to guarantee budget balance at the provider (based on the efficiency property achieved by Shapley values). It should not be too large though, in order to retain as many users as possible in the cloud service, with better social welfare.

In addition, although the Shapley values used for deciding the payments are not exactly the Shapley values

---

Algorithm 1 Offline Auction Mechanism

1: Calculate a feasible traffic schedule $\boldsymbol{b} = \{b_n(t), \forall n \in [N], \forall t \in [T]\}$, using the offline traffic scheduling algorithm in Sec. 4.3 with $x_n = 1, \forall n \in [N]$

2: **for all** users $n \in [N]$ **do**

3:     Calculate the offline Shapley value ϕ_n based on $\boldsymbol{b}$

4:     **if** user n's bid price $\hat{e}_n \geq \gamma\phi_n$ **then**

5:         Accept user n by setting $x_n = 1$

6:         Charge user n by payment $j_n = \gamma\phi_n$

7:     **else**

8:         Reject user n by setting $x_n = 0$

9:     **end if**

10: **end for**

11: Re-compute the traffic schedule $\boldsymbol{b}$ using the offline traffic scheduling algorithm in Sec. 4.3, with $x_n, \forall n \in [N]$, computed above

---

computed based only on traffic of admitted requests, we show in Theorem 3 that our auction achieves all four target properties of the mechanism design. The proof is given in Appendix B.

Theorem 3. *The Shapley value based offline auction described in Alg. 1 is computationally efficient, budget balanced, individually rational, truthful in bidding price, and $\frac{\delta}{\delta-\gamma}$-competitive in social welfare, where δ and γ are constants given in Assumptions 2 and 3.*

For example, if $\delta > 10$ and $1 < \gamma < 2$, the competitive ratio $\frac{\delta}{\delta-\gamma}$ is no larger than 1.25. Note that we only require a feasible traffic schedule $\boldsymbol{b}$ to be computed in line 1 of Alg. 1 by the offline traffic scheduling algorithm, and can rest assured that Theorem 3 is correct. We give an algorithm which produces a feasible traffic schedule in the following.

4.3 Offline Traffic Scheduling Algorithm

Given the admission decisions, $x_n, \forall n \in [N]$, we solve the following linear program (LP) to compute a feasible traffic schedule to the social welfare maximization problem in (1):

$$\max \sum_{n \in [N]} x_n e_n - \sum_l \alpha_l c_l \tag{3}$$

$$\text{s.t.} \quad x_n B_n = \sum_{t=t_n}^{t_n+d_n-1} b_n(t) \quad \forall n \in [N] \tag{3a}$$

$$\sum_{n \in \tilde{P}_l} b_n(t) \leq c_l \quad \forall l \in [L], \forall t \in [T] \tag{3b}$$

$$b_n(t) \geq 0 \quad \forall n \in [N], \forall t \in [T] \tag{3c}$$

$$c_l \leq C_l \quad \forall l \in [L] \tag{3d}$$

Here, $c_l, \forall l \in [L]$, and $b_n(t), \forall n \in [N], \forall t \in [T]$, are decision variables, where c_l denotes the maximum traffic on link l among the time slots within the accounting period T. We can see that the objective function is the same as that in (1), except for computing the ISP charge

$v(\boldsymbol{a})$ using the maximum-traffic model. The constraints are similar too, except for replacing $a_l(t), \forall t \in [T]$, by c_l. It is easy to see that if there exists a feasible solution $\boldsymbol{b}$ to LP (3) at the given x_n's, then this $\boldsymbol{b}$ and x_n's render a feasible solution to problem (1). Hence by solving LP (3) using any efficient LP solver, we can find a feasible solution to problem (1) with the given x_n's.

We note that solving LP (3) actually produces an optimal traffic schedule that maximizes the social welfare (or equivalently minimizes the ISP charge) with given x_n's, under the maximum-traffic ISP charge model. Nevertheless, we only require a feasible traffic schedule in Alg. 1, and hence Theorem 3 is correct no matter whether the ISP charge is based on the 95th-percentile or maximum-traffic model.

Remark: We have presented the offline auction focusing on the request model that a user specifies an amount of data and the transmission deadline. In the case that requests for exclusive bandwidths exist, Alg. 1 works in a simplified fashion by setting $b_n(t)$'s of those users directly to the respective demanded bandwidth rates. The existence of a feasible traffic schedule is guaranteed by Assumption 1.

The overall messaging complexity of the offline auction is $O(N)$. Suppose we use $O(N^2)$ random permutations when calculating the Shapley value, which is adopted in our simulations. We also use a heap to record the maximum traffic. The time complexity of the offline auction is $O(N^3 D \log T + M)$, where M is the time complexity of solving LP (3).

5 ONLINE AUCTION

We now design an online version of the auction in Alg. 1, which makes decisions on admission, traffic scheduling and payment of each user's request upon its arrival.

5.1 Online Auction based on Estimation of Offline Shapley Value

The online auction follows a similar idea as Alg. 1, in order to retain the nice properties achieved by the offline auction: the cloud provider calculates the Shapley value of each user upon the arrival of her request, and rejects the user if her bid price is low relative to the Shapley value. In the online auction, we also stick to the offline Shapley value defined in Definition 5, computed as the share of each user in the overall ISP charge incurred when all users' transfer requests are scheduled, since we are able to show nice properties achieved by an auction based on such a Shapley value. Again we assume a known online traffic scheduling algorithm which produces a feasible traffic schedule, to be discussed in the next Sec. 5.2, that essentially divides the data to be transmitted evenly among all the allowed time slots. The challenge, however, is that it is impossible to precisely calculate such an offline Shapley value on the spot without knowing any future request arrivals. In our design, we estimate a user's offline Shapley value as

best as we can. Here we present two methods to estimate the offline Shapley value, assuming that we can estimate the probability distribution of user arrivals using some regression techniques [45].

When a new user n arrives at t_n, our first estimation method is based on the Monte Carlo simulation [46]: we randomly simulate future user arrivals according to the arrival distribution and then calculate the offline Shapley value of user n using all the produced user arrivals. Using this method, we are guaranteed to obtain the expectation of the offline Shapley value (based on the estimated probability distribution of user arrivals), and the precision is higher if we repeat the simulation for more times.

Our second estimation method is according to an efficient formula, which can approximately calculate the expectation of the offline Shapley value. We first calculate user n's Shapley value ϕ'_n according to the ISP charge incurred by all the users that have arrived so far, using a formula similar to (2), assuming that all their requests are scheduled. This value, ϕ'_n, can be considered as the amortized cost of user n supposing no new users would arrive after t_n. Then we scale ϕ'_n by a factor $\bar{v}([N])t_n/(v([1,n])T)$ to obtain an estimate $\hat{\phi}_n$ of the offline Shapley value, i.e., $\hat{\phi}_n = \frac{\phi'_n \bar{v}([N])t_n}{v([1,n])T}$. Here $v([1,n])$ is the overall ISP charge due to the traffic of the first n arrived users, and $\bar{v}([N])$ is the expectation of the overall ISP charge due to traffic of all N users that may arrive in T. The intuition behind this scaling is as follows: Assume the probability distribution of user request arrival is the same in any time slot, i.e., the probability for a specific number of users with a specific transfer request to arrive is the same among different time slots. Then in expectation, t_n/T fraction of all the users arrive no later than t_n. For the upcoming users after t_n, their transfer data sizes and allowable transmission delays follow the same distributions as those of the n earlier users, based on the assumption above. So the cost share of user n will eventually be amortized by T/t_n times more users, and hence we scale the Shapley value ϕ'_n down by a factor of t_n/T to estimate the effect of more arriving users in user n's offline Shapley value. Meanwhile, the total ISP charge will increase from $v([1,n])$ to $\bar{v}([N])$, so we also multiple the ratio $\bar{v}([N])/v([1,n])$ in the estimation.

To estimate $\bar{v}([N])$, the provider can use the history records in previous accounting periods, and take the average ISP charge per accounting period as the estimate. Or another approach is as follows: Let $P(b,d,m)$ denote the probability that m users with a per-time-slot traffic b and an allowable delay no smaller than d arrive in time slot t which is known. First, we calculate the probability that m users are transmitting traffic b in time slot t: $p(b,m)$. These m users should arrive during time slots $[t-D+1,t]$. So $p(b,m) = \sum_{m_1+m_2+...+m_D=m} \prod_{d=1}^{D} P(b,d,m_d)$. Then the probability that the total traffic at time t is B is: $p'(B) = \sum_{m_1,m_2,...,m_{b_{max}}} \prod_b p(b,m_b)$, where b_{max} is the upper

bound of the per-time-slot traffic, and $m_1 + 2m_2 + ... + b_{max}m_{b_{max}} = B$. Next, the probability that the peak traffic over time slots $[t_n, T]$ is no more than B is: $\xi(B) = (\sum_{i=0}^{B} p'(i))^{T-t_n}$. The probability that the peak traffic is exactly B is: $\xi'(B) = \xi(B+1) - \xi(B)$. Finally, we obtain $\bar{v}([N]) = \sum_{i=0}^{\widetilde{B}} \widetilde{B}\xi'(i) + \sum_{i=\widetilde{B}}^{\infty} i\xi'(i)$, where $\widetilde{B}$ is the peak per-time-slot traffic during $[1, t_n]$.

The sketch of our online algorithm is given in Alg. 2, where we adopt the second method above to estimate the offline Shapley value of each user upon her arrival. The messaging complexity of this algorithm is still $O(N)$, and its time complexity is $O(N^4 D \log T)$.

Theorem 4. *The online auction described in Alg. 2 is computationally efficient, individually rational, truthful in both bidding price and bidding data size, budget balanced in expectation, and $\frac{\delta}{\delta-\gamma}$-competitive in social welfare in expectation, if the expectation of the offline Shapley value of each user can be estimated upon her arrival.*

For the online auction, we can prove its truthfulness in both bidding price and bidding data size (and only assume that the deadline is known to the auctioneer), which is stronger than the property of the offline auction. The proof of Theorem 4 is given in Appendix C.

5.2 Online Traffic Scheduling Algorithm

We design a simple but effective online algorithm to provide a traffic schedule $\{b_n(t), \forall t \in [T]\}$, upon the arrival of user n, utilizing the idea of simple smoothing [42]: We distribute user n's data traffic equally over the entire transmission time window $[t_n, t_n + d_n - 1]$, such that $b_n(t) = B_n/d_n$, $\forall t \in [t_n, t_n + d_n - 1]$. This simple algorithm provides us a simple feasible solution to (1), since we assume the maximum capacity of the ISP's links is sufficient for all the data transfers if the cloud provider is willing to pay an ISP charge at any amount (Assumption 1).

Similar to the case in Sec. 4.3 where the solution we find further maximizes social welfare under the maximum-traffic charging model, here we move a step further to show that the solution produced by such a

---

Algorithm 2 Online Auction Mechanism

1: ▷ **Upon arrival of user n's transfer request:**
2: Calculate the traffic schedule $b_m(t), \forall m \in [1,n], \forall t \in [T]$, using the traffic scheduling algorithm in Sec. 5.2
3: Calculate the Shapley value ϕ'_n based on $b_m(t), \forall m \in [1,n], \forall t \in [T]$
4: Calculate the estimation of the offline Shapley value: $\hat{\phi}_n = \frac{\phi'_n \bar{v}([N])t_n}{v([1,n])T}$
5: **if** user n's bid price $\hat{e}_n \geq \gamma\hat{\phi}_n$ **then**
6:     Accept user n by setting $x_n = 1$
7:     Charge user n by payment $j_n = \gamma\hat{\phi}_n$
8: **else**
9:     Reject user n by setting $x_n = 0$
10: **end if**

simple-smoothing algorithm is also efficient in minimizing the ISP charge part of the social welfare, under the maximum-traffic charging model. The intuition is that by evenly distributing the traffic, we may avoid high traffic peaks.

Lemma 1. *Our simple smoothing based online traffic scheduling algorithm achieves a $2\mathbb{H}_D \in O(\log D)$ competitive ratio in the peak traffic on each link, where $D = \max_{n \in N}\{d_n\}$ is the maximal allowable delay among all users' requests, and $\mathbb{H}_D = \sum_{d=1}^{D} 1/d$.*

Lemma 1 shows that if the optimal solution of (1) leads to a peak traffic C along a link in the system, our online scheduling algorithm achieves a peak traffic at most $2\mathbb{H}_D C$ on the link. The proof of Lemma 1 is given in Appendix D. This competitive ratio is also asymptotically tight. Consider D users who arrive at $t_1 = 1, t_2 = 2, \ldots, t_D = D$, respectively, and maximum transfer completion time $d_1 = D, d_2 = D-1, \ldots, d_D = 1$, respectively. The data size of all these requests is $B_1 = \cdots = B_D = 1$. The minimal peak traffic is 1 when user 1's data is transferred in time slot 1, user 2's data is transmitted in time slot 2, and so on. With our online scheduling algorithm, the peak traffic $\sum_{d=1}^{D} 1/d$ occurs at time D.

Based on Lemma 1, we can bound the ratio of the overall ISP charge for traffic in the entire network incurred by our online scheduling algorithm over that computed with the optimal solution to (1), as follows, under the maximum-traffic charging model. The proof of Theorem 5 is given in Appendix E.

Theorem 5. *The simple smoothing based online traffic scheduling algorithm achieves a $2\alpha\mathbb{H}_D$ competitive ratio in overall ISP charge in the system, under the maximum-traffic charging model, where α is the ratio between the highest and lowest unit bandwidth prices among all links, and $\mathbb{H}_D = \sum_{d=1}^{D} 1/d$.*

Finally, we readily see that Alg. 2 and all results derived in this section also apply to the case that some users request exclusive bandwidth rates.

6 PERFORMANCE EVALUATION

6.1 Simulation Setup

We evaluate the performance of our bandwidth auctions through trace-driven simulations. We simulate the topology of Google datacenter network [41], which includes 12 datacenters and 16 links between the datacenters. The capacity of the links is assumed to be always sufficient for data transfer. A unit bandwidth price α_l for each link is generated randomly in the range $[1, 2]$. We uniformly randomly choose one time slot as the arrival time t_n of each request, such that each user arrives at each time slot with an equal probability. We generate our data transfer workloads in two ways: (1) assigning each user with a transfer data size B_n based on the recorded Wikipedia page view counts (available at [47]) during

her arrival hour, multiplied by a random coefficient in $[1, 4]$; (2) uniformly randomly generating a data transfer size in the range of $[10^4, 10^5]$. The source and destination datacenters of each request are randomly picked among all the datacenters, and the shortest path between the two is indicated for data transfer. The allowable delay d_n is randomly set within $[1, D]$. The valuation e_n of user n is set proportional to the data size B_n, negatively correlated with the allowable delay d_n, and scaled by a random coefficient. Based on the data we generated, we have $\delta = 10$. By default, $\gamma = 2$, and the percentage of flat bandwidth requests among all requests is zero.

We evaluate the performance of our auctions against the optimal performance achieved by the optimal solution of the social welfare maximization problem (1). The evaluations are based on the maximum-traffic charging model, because problem (1) under the 95th-percentile charging model cannot be formulated into a standard mixed integer program and cannot be readily solved by an optimization solver. We run each experiment for 10 times (5 following the Wikipedia workload and 5 with randomly generated workload) and present the average result.

6.2 Performance of the Offline Auction

We first compare the social welfare achieved by our offline auction Alg. 1 and the optimal social welfare by solving (1) exactly, with $T = 1000$. In Fig. 3, we show the ratio of the optimal social welfare over the social welfare derived by Alg. 1, averaged over 10 times of the experiment, together with the best and worst ratios obtained among the 10 trials. Our theoretical upper bound of the approximation ratio is $\frac{\delta}{\delta-\gamma} = 1.25$. Fig. 3 shows that the average ratio that our offline auction actually achieves is much better than the theoretical bound. The impact of D on the ratio is not obvious. When N is larger, the performance tends to downgrade a bit. This is because with a larger N, more random permutations are required in computing Shapley value with Eqn. (2), in order to obtain an accurate estimation, *i.e.*, using $O(N^2)$ random permutations may not be sufficient. This calls for a more efficient permutation selection method (rather than random selection among the set of all possibly permutations over $[1, N]$), to achieve more accurate estimation with fewer permutations, whose in-depth study we leave as future work.

Next we study the impact of the percentage of flat bandwidth requests on the social welfare ratio. Given the data sizes in delay-tolerant data transfer requests are in the range of $[100, 10^5]$ based on the Wikipedia traces, we set the bandwidth rate demanded in a flat bandwidth request n by randomly picking from $[\frac{100}{d_n}, \frac{10^5}{d_n}]$. In addition, $T = 400, N = 2000$, and $D = 10$ in this set of experiments. Fig. 4 shows that the performance of our offline auction remains stable at any ratio of the flat bandwidth requests and the delay-tolerant data transfer requests, demonstrating that the offline auction Alg. 1 can handle any mixed request patterns well.
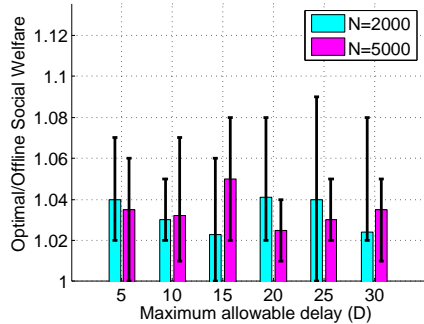
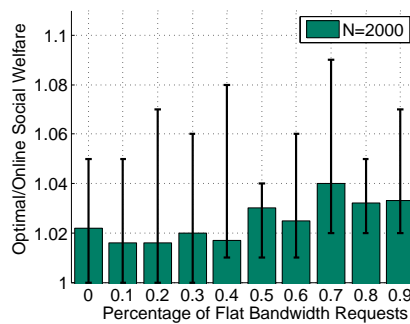Fig. 3. Performance of the offline auction

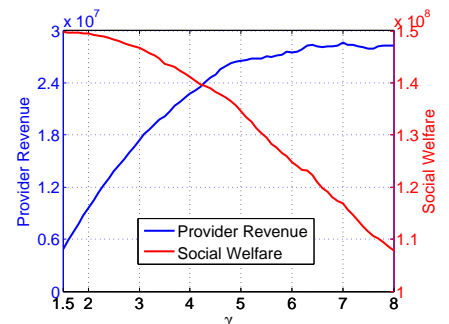Fig. 4. Performance of offline auction with different percentages of flat bandwidth requests

Fig. 5. Impact of γ on the provider's revenue and the social welfare
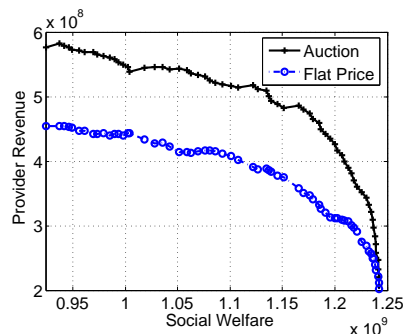

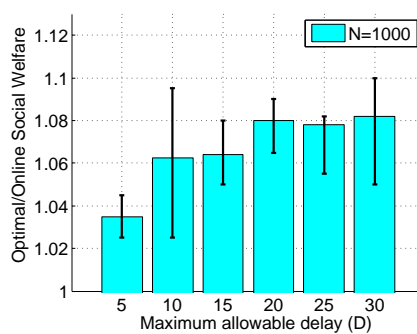Fig. 6. Comparison of the provider's revenue between our offline auction and flat pricing

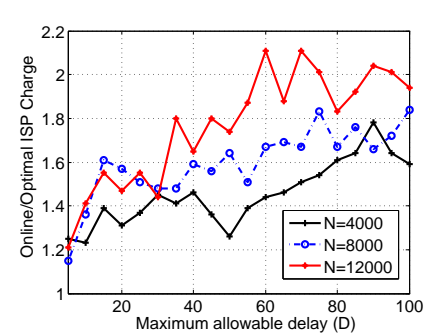Fig. 7. Performance of the online auction

Fig. 8. Performance of online scheduling algorithm

We also study the impact of parameter γ in Alg. 1 on the provider's revenue and the social welfare. Here we set $T = 100, N = 2000, D = 15$, and vary γ used in deciding the acceptance of users $\hat{e}_n \geq \gamma\phi_n$ and the payment $j_n = \gamma\phi_n$ between 1.5 and 8. Fig. 5 shows that with the increase of γ, the social welfare decreases, and the provider's revenue in general increases, revealing a tradeoff between social welfare and the provider's revenue. With larger γ, the admission criterion becomes higher, and hence more users are rejected. Both the total valuation and the ISP charge decrease due to more rejected users. Since most users' valuations are much larger than their incurred ISP charges, the decrease of the total valuation is dominating, leading to reduced social welfare. For the provider, she accepts fewer requests but charges a higher payment from each winner. When γ is relatively small, the increase of payment with the increase of γ is dominating, and the provider's revenue increases; when γ is larger, the impacts of per-winner payment increase and winner number decrease roughly cancel out, and the provider's revenue becomes more stable. We also notice that for $\gamma > 5$, the provider's revenue increases very little with the decrease of social welfare. This implies that the provider can set γ at a reasonable level (about 4 in this case) to achieve a satisfying level of revenue, while looking after the benefit of most users (social welfare).

Finally we compare the performance of our offline auction with that of a flat pricing policy. Under the flat pricing policy, each user is charged by her total data size B_n times a flat price rate. The provider also faces the social welfare and revenue trade-off when using flat pricing: when the price is higher, the provider may gain more revenue for serving each user while driving more users away, affecting social welfare. So we compare the provider's revenue achieved under the two pricing methods, our auction and flat pricing, when the same total social welfare is achieved. Under our auction mechanism, we can calculate the revenue achieved at any given social welfare, just like what we do in Fig. 5. For flat pricing, given an amount of social welfare, we find the unique corresponding flat price by gradually increasing a price rate until the target social welfare is achieved; then the revenue under such a flat price can be calculated. As we can see in Fig. 6, under either pricing mechanism, if more social welfare is desired, the provider has to give up some revenue. At any given amount of social welfare, the revenue under our auction mechanism is much larger than that under flat pricing, revealing the better performance of our mechanism.

6.3 Performance of the Online Auction

We now evaluate our online auction Alg. 2, and show in Fig. 7 the ratio of the social welfare achieved by Alg. 2 over the optimal social welfare by solving (1) exactly, with $T = 100$. The average ratio is smaller than the theoretical approximation ratio 1.25 in all cases.

Finally, we evaluate the ratio between the overall ISP charge incurred by our online scheduling algorithm in Sec. 5.2) and the optimal ISP charge, derived based on the optimal solution to (1). The ratio between the maximum and minimum unit bandwidth prices on all links is set to $\alpha = 2$, and we use $T = 1000$. Fig. 8 shows that the performance of our online scheduling algorithm slowly downgrades with larger D, which is consistent with our analysis in Theorem 5. Nevertheless, for large D, the ratio is around 2, still much lower than the theoretical upper bound given in Theorem 5. It also shows that the extreme worst cases (like the example we present in Sec. 5.2) rarely happen in practical scenarios.

7 EXTENSION

Recall that our offline auction Alg. 1 is truthful in bidding price and our online auction Alg. 2 is truthful in bidding price and bidding data size. Among the three parts in a user's bid, truthfulness in the deadline d_n is still missing. We now propose an extension of the online auction mechanism, which achieves fully truthfulness in all parameters in a user's bid.

Intuitively, a user cannot bid a later deadline than her true need, which will lead to incomplete transmission by the time of her said deadline. Then the only possibility for untruthfulness in the deadline is that a user may report an earlier deadline than her true need. In order to prevent such manipulation, our approach is to let the auctioneer decide an optimal transfer completion time for the user, which brings the lowest price for the user. Then the user has no incentive to report an earlier deadline since the bandwidth allocation at the auctioneer is already to achieve the best utility for her. The modified online auction is given in Alg. 3. Let $\hat{\phi}_n(d)$ denote the estimation of the offline Shapley value of user n in Alg. 2 under deadline d, which decides the threshold for acceptance/rejection in our mechanism. Our extended online auction works by running Alg. 2 with different deadlines d, for $d = 1, \dots, d_n$, and choosing the smallest threshold: $\hat{\phi}_n^{min} = \min_{d \in [d_n]}\{\hat{\phi}_n(d)\}$, as well as the corresponding traffic schedule. The auction accepts a user if $\hat{e}_n \geq \gamma \hat{\phi}_n^{min}$.

We prove the properties achieved by Alg. 3 in Theorem 6, with proof given in Appendix F.

Theorem 6. *The extended online auction Alg. 3 is computationally efficient, individually rational, truthful in bidding price, deadline and data size, as well as budget balanced in expectation.*

We note that since the online scheduling algorithm is modified in this extension mechanism, the competitive

---

Algorithm 3 Extended Online Auction Mechanism

1: ▷ **Upon arrival of user n's transfer request:**
2: **for all** $d \in [d_n]$ **do**
3:     Calculate the traffic schedule $b_m(t), \forall m \in [1, n], \forall t \in [T]$, using the traffic scheduling algorithm in Sec. 5.2, supposing user n's deadline is d instead of d_n
4:     Calculate the Shapley value $\phi_n'(d)$ based on traffic schedule $b_m(t), \forall m \in [1, n], \forall t \in [T]$
5:     Calculate the estimation of the offline Shapley value: $\hat{\phi}_n(d) = \frac{\phi_n'(d)\bar{v}([N])t_n}{v([1,n])T}$
6: **end for**
7: Choose the smallest offline Shapley value $\hat{\phi}_n^{min} = \min_{d \in [d_n]}\{\hat{\phi}_n(d)\}$, and the corresponding optimal deadline d^{min}.
8: **if** user n's bid price $\hat{e}_n \geq \gamma \hat{\phi}_n^{min}$ **then**
9:     Accept user n by setting $x_n = 1$
10:    Charge user n by payment $j_n = \gamma \hat{\phi}_n^{min}$
11:    Schedule user n's traffic in d^{min} time slots
12: **else**
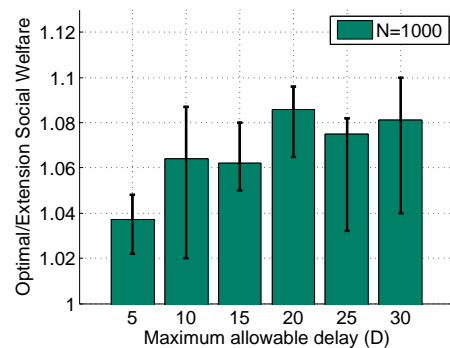13:   Reject user n by setting $x_n = 0$
14: **end if**

Fig. 9. Performance of the extended online auction

ratio in social welfare is no longer guaranteed. We instead evaluate the performance of Alg. 3 in social welfare using trace-driven simulations under the same setup as in previous experiments. Fig. 9 shows the ratio of the optimal social welfare over the social welfare derived by Alg. 3, averaged over 10 times of the experiment, together with the best and worst ratios obtained among the 10 trials. Here $T = 100$. We observe that the ratios are very close to those achieved by the online auction Alg. 2 in Fig. 7. The reason behind the unaffected performance is as follows: The search for the optimal completion time leading to the lowest Shapley value is in line with the minimization of the ISP charge; if a user's traffic would increase the maximum traffic in the billing time slot, such a traffic schedule will result in a relatively high Shapley value, and thus is unlikely to be chosen. So in the long term, traffic schedules which smooth the peak traffic are more likely to be chosen, and the total ISP charge and hence the social welfare remain at similar levels.

8 CONCLUSION

This paper presents a novel Shapley value based auction mechanism for dynamic pricing of inter-datacenter on-demand bandwidth. Targeting truthfulness, individual rationality, budget balance and competitiveness in social welfare, our mechanism design is divided into two steps: we first propose the offline version of the auction, which exploits the Shapley values in admission control and payment computation together with an optimal traffic scheduling algorithm, achieving all four desired properties; we then design the online auction approximating the offline mechanism, retaining the nice properties. As the first pricing mechanism for inter-datacenter bandwidth, our design has been targeting a generic system model, by allowing two representative types of bandwidth demands and two typical ISP charging models. There yet exist other dimensions that our model can be further extended on, *e.g.*, involving routing decisions for selecting (possibly) multiple paths for each transfer task. We also plan to extend this work by addressing computational resource and bandwidth resource together in cloud resource allocation, and seek to build a more comprehensive auction framework for cloud providers. We seek to explore such extensions in our future work.

REFERENCES

[1] *Data Center Map*, 2014. http://www.datacentermap.com/datacenters.html.

[2] K. K. Ramakrishnan, Prashant Shenoy, and Jacobus Van der Merwe. Live Data Center Migration across WANs: a Robust Cooperative Context Aware Approach. In *Proc. of the 2007 SIGCOMM workshop on Internet Network Management*, 2007.

[3] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. C. M. Lau. Scaling Social Media Applications onto Geo-distributed Clouds. In *Proc. of IEEE INFOCOM*, 2012.

[4] Nikolaos Laoutaris, Michael Sirivianos, Xiaoyuan Yang, and Pablo Rodriguez. Inter-datacenter Bulk Transfers with NetStitcher. In *Proc. of ACM SIGCOMM*, 2011.

[5] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The Cost of a Cloud: Research Problems in Data Center Networks. In *Proc. of ACM SIGCOMM*, volume 39, pages 68–73, 2008.

[6] Qian Wang, Kui Ren, and Xiaoqiao Meng. When Cloud Meets eBay: Towards Effective Pricing for Cloud Computing. In *Proc. of IEEE INFOCOM*, 2012.

[7] Linquan Zhang, Zongpeng Li, and Chuan Wu. Dynamic Resource Provisioning in Cloud Computing: A Randomized Auction Approach. In *Proc. of IEEE INFOCOM*, pages 433–441, 2014.

[8] Weijie Shi, Linquan Zhang, Chuan Wu, Zongpeng Li, and Francis C. M. Lau. An Online Auction Framework for Dynamic Resource Provisioning in Cloud Computing. In *Proc. of ACM SIGMETRICS*, 2014.

[9] *Amazon EC2 Pricing*, 2015. http://aws.amazon.com/ec2/pricing/.

[10] Nikolaos Laoutaris, Georgios Smaragdakis, Pablo Rodriguez, and Ravi Sundaram. Delay Tolerant Bulk Data Transfers on the Internet. In *Proc. of ACM SIGMETRICS*, volume 37, pages 229–238, 2009.

[11] Dengji Zhao, Dongmo Zhang, and Laurent Perrussel. Multi-unit Double Auction under Group Buying. In *ECAI*, 2012.

[12] Hitesh Ballani, Paolo Costa, Thomas Karagiannis, and Ant Rowstron. Towards Predictable Datacenter Networks. In *Proc. of ACM SIGCOMM*, 2011.

[13] Guiyi Wei, Athanasios V Vasilakos, Yao Zheng, and Naixue Xiong. A Game-Theoretic Method of Fair Resource Allocation for Cloud Computing Services. *The Journal of Supercomputing*, 2010.

[14] M Rodriguez Sossa and Rajkumar Buyya. Deadline based Resource Provisioning and Scheduling Algorithmfor Scientific Workflows on Clouds. *IEEE Transactions on Cloud Computing*, 2014.

[15] H Hong, D Chen, C Huang, K Chen, and C Hsu. Placing Virtual Machines to Optimize Cloud Gaming Experience. *IEEE Transactions on Cloud Computing*, 2015.

[16] J. Guo, F. Liu, X. Huang, J. C.S. Lui, M. Hu, Q. Gao, and H. Jin. On Efficient Bandwidth Allocation for Traffic Variability in Datacenters. In *Proc. of IEEE INFOCOM*, 2014.

[17] Hong Zhang, Bo Li, Hongbo Jiang, Fangming Liu, Athanasios V Vasilakos, and Jiangchuan Liu. A Framework for Truthful Online Auctions in Cloud Computing with Heterogeneous User Demands. In *Proc. of IEEE INFOCOM*, 2013.

[18] M. Polverini, A. Cianfrani, S. Ren, and A. V. Vasilakos. Thermal-Aware Scheduling of Batch Jobs in Geographically Distributed Data Centers. *IEEE Transactions on Cloud Computing*, 2014.

[19] L. Mashayekhy, M. M. Nejad, D. Grosu, and A. V. Vasilakos. Incentive-Compatible Online Mechanisms for Resource Provisioning and Allocation in Clouds. In *Proc. of IEEE CLOUD*, 2014.

[20] D. Niyato, A. V. Vasilakos, and Z. Kun. Resource and Revenue Sharing with Coalition Formation of Cloud Providers: Game Theoretic Approach. In *Proc. of IEEE CCGRID*, 2011.

[21] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos. Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of the Art, and Future Directions. *Proceedings of the IEEE*, 102(1):11–31, 2014.

[22] L. Wang, F. Zhang, J. A. Aroca, A. V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu. GreenDCN: A General Framework for Achieving Energy Efficiency in Data Center Networks. *IEEE Journal On Selected Areas in Communications*, 32(1):4–15, 2014.

[23] Ganesha Shanmuganathan, Ajay Gulati, and Peter Varman. Defragmenting the Cloud Using Demand-Based Resource Allocation. In *Proc. of ACM SIGMETRICS*, 2013.

[24] William Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of finance*, 16(1):8–37, 1961.

[25] Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. Computationally Manageable Combinational Auctions. *Management science*, 44(8):1131–1147, 1998.

[26] Ahuva Mu'Alem and Noam Nisan. Truthful Approximation Mechanisms for Restricted Combinatorial Auctions. *Games and Economic Behavior*, 2008.

[27] Daniel Lehmann, Liadan Ita O'callaghan, and Yoav Shoham. Truth Revelation in Approximately Efficient Combinatorial Auctions. *Journal of the ACM (JACM)*, 49, 2002.

[28] Ron Lavi and Chaitanya Swamy. Truthful and near-Optimal Mechanism Design via Linear Programming. In *Proc. of FOCS*, 2005.

[29] D. Xu, X. Liu, and A. V. Vasilakos. Traffic-Aware Resource Provisioning for Distributed Clouds. *IEEE Cloud Computing*, 2015.

[30] Z. Feng, Y. Zhu, Q. Zhang, L. M Ni, and A. V. Vasilakos. TRAC: Truthful Auction for Location-Aware Collaborative Sensing in Mobile Crowdsourcing. In *Proc. of IEEE INFOCOM*, 2014.

[31] K. Chen, C. Hu, X. Zhang, K. Zheng, Y. Chen, and A. V. Vasilakos. Survey on Routing in Data Centers: Insights and Future Directions. *IEEE Network*, 2011.

[32] Jeffrey K MacKie-Mason and Hal R Varian. Pricing the Internet. *Public Access to the Internet*, 1994.

[33] Patrick Maillé and Bruno Tuffin. Pricing the Internet with Multi-bid Auctions. *IEEE/ACM Transactions on Networking*, 14(5):992–1004, 2006.

[34] Aurel A Lazar and Nemo Semret. Design and Analysis of the Progressive Second Price Auction for Network Bandwidth Sharing. *Telecommunication Systems*, 13, 2001.

[35] Lloyd S Shapley. A Value for N-person Games. Technical report, Rand Corp Santa Monica (No. RAND-P-295), 1952.

[36] R. T.B. Ma, D.M. Chiu, J. C.S. Lui, M. Vishal, and D. Rubenstein. Internet Economics: The use of Shapley value for ISP settlement. *IEEE/ACM Transactions on Networking*, 18(3):775–787, 2010.

[37] Rade Stanojevic, Nikolaos Laoutaris, and Pablo Rodriguez. On Economic Heavy Hitters: Shapley Value Analysis of 95th-Percentile Pricing. In *Proc. of ACM IMC*, 2010.

[38] Joan Feigenbaum, Christos Papadimitriou, and Scott Shenker. Sharing the Cost of Muliticast Transmissions. In *Proc. of ACM STOC*, pages 218–227, 2000.

[39] Nandan Garg and Daniel Grosu. A Faithful Distributed Mechanism for Sharing the Cost of Multicast Transmissions. *IEEE Transactions on Parallel and Distributed Systems*, 2009.

[40] Shahar Dobzinski, Aranyak Mehta, Tim Roughgarden, and Mukund Sundararajan. Is Shapley Cost Sharing Optimal? In *Algorithmic Game Theory*, pages 327–336. Springer, 2008.

[41] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. B4: Experience with a Globally-Deployed Software Defined WAN. In *Proc. of ACM SIGCOMM*, 2013.

[42] Leana Golubchik, Samir Khuller, Koyel Mukherjee, and Yuan Yao. To Send or not to Send: Reducing the Cost of Data Transmission. In *Proc. of IEEE INFOCOM*, 2013.

[43] Xenofontas Dimitropoulos, Paul Hurley, Andreas Kind, and Marc Ph Stoecklin. On the 95-percentile Billing Method. In *Passive and Active Network Measurement*, pages 207–216. 2009.

[44] Roger B Myerson. Graphs and Cooperation in Games. *Mathematics of operations research*, 2(3):225–229, 1977.

[45] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to Linear Regression Analysis*. John Wiley & Sons, 2012.

[46] Patrick Stuedi and Gustavo Alonso. Recall and Precision in Distributed Bandwidth Allocation. In *Quality of Service, 2007 Fifteenth IEEE International Workshop on*. IEEE, 2007.

[47] *Wikipedia Stats*, 2013. http://dumps.wikimedia.org/other/pagecounts-ez/.

APPENDIX A
PROOF OF THEOREM 1

Proof. We prove the computational complexity of the social welfare maximization problem by a reduction to the 0-1 knapsack problem, which is NP-hard. Suppose the ISP charge is 0, and all users allow 0 delay (immediate transmission), and only consider one time slot. Then the social welfare maximization problem becomes maximizing the sum of the valuation of served users, under the constraint that the sum of their data size does not exceed the bandwidth limit. This is exactly the form of the 0-1 knapsack problem. □

APPENDIX B
PROOF OF THEOREM 3

Proof. It is ready to see that each step of Alg. 1 involves polynomial computation complexity.

First we calculate the revenue of the provider. The actual ISP charge is no larger than the ISP charge when all users are served, *i.e.*, $\sum_{n \in [N]} \phi_n$. The payment collected is $\gamma(\sum_{n \in \mathcal{N}} \phi_n)$, where $\mathcal{N}$ is the set of accepted users. So:

$$
\begin{aligned}
u_P \quad & \geq \gamma(\sum_{n \in \mathcal{N}} \phi_n) - \sum_{n \in [N]} \phi_n \\
& = \gamma(\sum_{n \in [N]} \phi_n - \sum_{n \notin \mathcal{N}} \phi_n) - \sum_{n \in [N]} \phi_n \\
& \geq \gamma(\sum_{n \in [N]} \phi_n - (1 - 1/\gamma) \sum_{n \in [N]} \phi_n) - \sum_{n \in [N]} \phi_n = 0
\end{aligned}
$$

So budget balance at the provider is guaranteed. The utility of rejected users is always 0. The utility of accepted users is $u_n = e_n - \gamma\phi_n \geq 0$ according to the offline auction mechanism. So the auction is individually rational.

For user n, if she is accepted with truthful bidding e_n, submitting her bid price $\hat{e}_n$ different from her valuation e_n does not affect the traffic scheduling algorithm and the traffic $b_n(t)$. Thus her Shapley value ϕ_n is the same either bidding e_n or $\hat{e}_n$. So her utility is not affected. For a rejected user, for the similar reason, her Shapley value ϕ_n cannot be decreased by false valuation bidding. So

the utility is also not increased if she is still rejected with bidding $\hat{e}_n$. And her utility becomes negative if accepted with false bid.

The optimal social welfare is no larger than the social welfare achieved in the following fictional scenario: all users are accepted and the ISP charge is 0. So $S_{opt} \leq \sum_{n \in [N]} e_n$. The total valuation under our auction is $\sum_{n \in \mathcal{N}} e_n$, and for $n \notin \mathcal{N}$, $e_n < \gamma\phi_n$. The actual ISP charge is no larger than $\sum_{n \in [N]} \phi_n$. So

$$
\begin{aligned}
S_{auc} & \geq \sum_{n \in \mathcal{N}} e_n - \sum_{n \in [N]} \phi_n \\
& \geq \sum_{n \in [N]} e_n - \sum_{n \notin \mathcal{N}} \gamma\phi_n - \sum_{n \in [N]} \phi_n \\
& \geq \sum_{n \in [N]} e_n - \gamma(1 - 1/\gamma) \sum_{n \in [N]} \phi_n - \sum_{n \in [N]} \phi_n \\
& = \sum_{n \in [N]} e_n - \gamma \sum_{n \in [N]} \phi_n
\end{aligned}
$$

Then the ratio between optimal social welfare that is derived by solving (1) exactly and the social welfare achieved by our offline auction is:

$$
\begin{aligned}
\frac{S_{opt}}{S_{auc}} & \leq \frac{\sum_{n \in [N]} e_n}{\sum_{n \in [N]} e_n - \gamma \sum_{n \in [N]} \phi_n} \\
& = 1 + \frac{\gamma \sum_{n \in [N]} \phi_n}{\sum_{n \in [N]} e_n - \gamma \sum_{n \in [N]} \phi_n} \\
& \leq 1 + \frac{\gamma \sum_{n \in [N]} \phi_n}{\delta \sum_{n \in [N]} \phi_n - \gamma \sum_{n \in [N]} \phi_n} = \frac{\delta}{\delta - \gamma}
\end{aligned}
$$

□

APPENDIX C
PROOF OF THEOREM 4

Proof. Similar to the proof in Appendix B, the online auction described in Alg. 2 has polynomial time complexity. The expectation of the provider's revenue is $E[u_P] \geq E[\gamma(\sum_{n \in \mathcal{N}'} \hat{\phi}_n) - \sum_{n \in [N]} \phi_n] \geq 0$, based on the assumption that $\sum_{n \notin \mathcal{N}'} \phi_n \leq (1 - 1/\gamma) \sum_{n \in [N]} \phi_n$ and $E[\hat{\phi}_n] = \phi_n$. So budget balance in expectation at the provider is guaranteed. The utility of the accepted users $u_n = e_n - \gamma\hat{\phi}_n \geq 0$ according to the online auction mechanism. So the auction is individually rational. Bidding a different valuation $\hat{e}_n$ does not change the traffic allocation $b_m(t)$. Also the overall ISP charge due to the traffic of the first n arrived users $v([1, n])$ and the expectation of overall ISP charge $\bar{v}([N])$ are not affected by the bidding price $\hat{e}_n$, since $v([1, n])$ depends on the previous users and $\bar{v}([N])$ depends on the user distribution estimation (e.g., using some regression techniques). Therefore the estimated Shapley value $\hat{\phi}_n$ is not affected by $\hat{e}_n$. So this mechanism is truthful in bidding price. Next we prove that the user cannot decrease the estimated Shapley value $\hat{\phi}_n$ by misreporting her data size B_n. First the user cannot bid a smaller data size B_n because otherwise the transmission task is not fully served. If a user submits a larger B_n, then her traffic at any time slot is larger according to our online scheduling algorithm. If

the ISP charge due to the first n users $v([1,n])$ is not increased under the false B_n value, then $\hat{\phi}_n$ remains the same. Otherwise, the billing time slot is one of the time slots during user n's stay. According to the property of Shapley value, her Shapley value will increase more than the increase of the ISP charge: $\phi'_n/v([1,n])$ is larger, and so is $\hat{\phi}_n$, which can only decrease the utility of user n. So this mechanism is also truthful in bidding data size. The optimal social welfare $S_{opt} \leq \sum_{n \in [N]} e_n$. The expectation of social welfare under the online auction is:

$$E[S_{auc}] \geq E[\sum_{n \in \mathcal{N}'} e_n - \sum_{n \in [N]} \phi_n]$$

$$\geq \sum_{n \in [N]} e_n - \sum_{n \notin \mathcal{N}'} \gamma E[\hat{\phi}_n] - \sum_{n \in [N]} \phi_n$$

$$\geq \sum_{n \in [N]} e_n - \gamma \sum_{n \in [N]} \phi_n$$

So the competitive ratio is $E[\frac{S_{opt}}{S_{auc}}] \leq \frac{\delta}{\delta - \gamma}$. $\qquad\square$

APPENDIX D
PROOF OF LEMMA 1

Proof. First we formulate the minimal peak traffic problem on a single link (4), as well as its dual (5).

$$\min C \qquad (4)$$

$$\text{s.t.} \quad B_n = \sum_{t=t_n}^{t_n+d_n-1} b_n(t) \quad \forall n \in [N] \qquad (4a)$$

$$\sum_{n \in [N]} b_n(t) \leq C \quad \forall t \in [T] \qquad (4b)$$

$$b_n(t) \geq 0 \quad \forall n \in [N], t \in [T] \qquad (4c)$$

$$\max \sum_{n \in [N]} z_n B_n \qquad (5)$$

$$\text{s.t.} \quad \sum_{t \in [T]} y(t) \leq 1 \qquad (5a)$$

$$z_n \leq y(t) \quad \forall n \in [N], t \in [t_n, t_n+d_n-1] \qquad (5b)$$

$$y(t) \geq 0 \quad \forall t \in [T] \qquad (5c)$$

For an arbitrary user set $[N]$, suppose the highest traffic appears at time T' in our online algorithm. Let $\mathbb{N}$ be the set of user involved in the traffic at time T', *i.e.* $n \in \mathbb{N}$ iff $T' \in [t_n, t_n+d_n-1]$. Then the peak traffic of our algorithm is $\sum_{n \in \mathbb{N}} B_n/d_n$. Note that for any user $n \in \mathbb{N}$, its delay satisfies $D \geq d_n \geq T' - t_n + 1$ and its arriving time $t_n \geq T' - D$. Now we give a feasible solution for the dual (5). Let $w = 2\mathbb{H}_D$. $y(t) = 1/(|t - T'|w)$ for $t \in [T' - D, T' + D]$, and 0 otherwise. For $n \in \mathbb{N}$, $z_n = 1/(d_n w)$, and 0 otherwise. We verify that this solution is feasible. $\sum_{t \in [T]} y(t) \leq 2\sum_{d=1}^{d=D}(1/dw) \leq 1$. For $\forall n \in \mathbb{N}$, we need to show that $z_n \leq \min_{t \in [t_n, t_n+d_n-1]} y(t)$. Note the distribution of the value of $y(t)$, we can conclude that $\min_{t \in [t_n, t_n+d_n-1]} y(t)$ equals either $y(t_n)$ or $y(t_n + d_n - 1)$. Since $y(t_n) = 1/(T' - t_n)w$, and $y(t_n + d_n - 1) = 1/(t_n + d_n - T' - 1)w$ (if $t_n + d_n - 1 \geq T'$, otherwise this case can be ignored), we only need to show $T' - t_n \leq d_n$, and $t_n + d_n - T' - 1 \leq d_n$. These 2 inequalities holds obviously. So this solution is feasible, and the value of the objective function of the dual (5) is $\sum_{n \in \mathbb{N}} B_n/(d_n w)$. $\qquad\square$

APPENDIX E
PROOF OF THEOREM 5

Proof. The key issue we need to prove here is: for every link, if the peak traffic of the optimal solution is C, then the peak traffic in our algorithm is no larger than $2\mathbb{H}_D C$. If we take all the users involved in this link, and create a new network with only one link and these users. Then the minimal peak traffic is no larger than C in the new network. So our algorithm running on single link network does not exceed $2\mathbb{H}_D C$. The schedules for the single link network and the original network are the same for these users, and so are the peak traffic. Then take the effect of unit prices into consideration, we get the bound on total ISP charge. $\qquad\square$

APPENDIX F
PROOF OF THEOREM 6

Proof. The truthfulness in bidding price and data size is due to the same reason in Appendix C. Suppose a user makes a smaller deadline bid. Her threshold $\hat{\phi}_n^{min}$ cannot be smaller than under the truthful bid. Because under the truthful bid, $\hat{\phi}_n^{min}$ is the smallest taken on $d = 1 \ldots d_n$. Thus the extended online auction is also truthful in bidding deadline.

$\qquad\square$
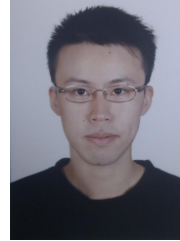
Weijie Shi received his B.E. degree in 2012, from Department of Computer Science and Technology, Tsinghua University, China. He is currently a PhD candidate in the Department of Computer Science, the University of Hong Kong. His research interests include cloud computing and mechanism design.
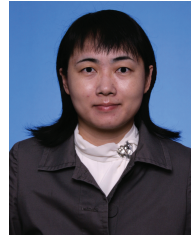
Chuan Wu received her B.E. and M.E. degrees in 2000 and 2002 from Department of Computer Science and Technology, Tsinghua University, China, and her Ph.D. degree in 2008 from the Department of Electrical and Computer Engineering, University of Toronto, Canada. She is currently an associate professor in the Department of Computer Science, The University of Hong Kong, China. Her research interests include cloud computing and online/mobile social network.

Zongpeng Li received his B.E. degree in Computer Science and Technology from Tsinghua University (Beijing) in 1999, his M.S. degree in Computer Science from University of Toronto in 2001, and his Ph.D. degree in Electrical and Computer Engineering from University of Toronto in 2005. He is currently an associate professor in the Department of Computer Science in the University of Calgary. His research interests are in computer networks, particularly in network optimization, multicast algorithm design, network game theory and network coding.