iDaaS: Inter-Datacenter Network as a Service

Wenxin Li, Deke Guo, Keqiu Li, *Senior Member, IEEE*, Heng Qi, and Jianhui Zhang

Abstract—Increasing number of Internet-scale applications, such as video streaming, incur huge amount of wide area traffic. Such traffic over the unreliable Internet without bandwidth guarantee suffers unpredictable network performance. This result, however, is unappealing to the application providers. Fortunately, Internet giants like Google and Microsoft are increasingly deploying their private wide area networks (WANs) to connect their global datacenters. Such high-speed private WANs are reliable, and can provide predictable network performance. In this paper, we propose a new type of service—inter-datacenter network as a service (iDaaS), where traditional application providers can reserve bandwidth from those Internet giants to guarantee their wide area traffic. Specifically, we design a bandwidth trading market among multiple iDaaS providers and application providers, and concentrate on the essential *bandwidth pricing* problem. The involved challenging issue is that the bandwidth price of each iDaaS provider is not only influenced by other iDaaS providers, but also affected by the application providers. To address this issue, we characterize the interaction between iDaaS providers and application providers using a Stackelberg game model, and analyze the existence and uniqueness of the equilibrium. We further present an efficient bandwidth pricing algorithm by blending the advantage of a geometrical Nash bargaining solution and the demand segmentation method. For comparison, we present two bandwidth reservation algorithms, where each iDaaS provider's bandwidth is reserved in a weighted fair manner and a max-min fair manner, respectively. Finally, we conduct comprehensive trace-driven experiments. The evaluation results show that our proposed algorithms not only ensure the revenue of iDaaS providers, but also provide bandwidth guarantee for application providers with lower bandwidth price per unit.

Index Terms—Inter-datacenter network, WANs, bandwidth price, Stackelberg game

1 Introduction

Large-scale Internet applications, such as video streaming and cloud computing, provide service to hundreds of millions of users. The enormous, and growing user demand has motivated application providers to place their application instances across multiple geographical regions, such as Netflix [1]. Accordingly, a large volume of wide area traffic will exhibit across different regions, due to the routine background computation and periodic data backup tasks. As revealed in [2], the wide area traffic accounts for up 45 percent of the total traffic of a typical business provider. A recent survey further highlights that the amount of such wide area traffic will double or triple in the next two to four years [3].

Traditionally, most application providers acquire bandwidth from Internet Service Providers (ISPs) for their wide area traffic. They, however, suffer unpredictable and unreliable network performance since the network bandwidth is shared by all traffic in a best effort manner in today's Internet. Nowadays, they can be freed from such performance issues with the success of private wide area networks (WANs) hosted by some Internet giants. For instance, B4, a private WAN connecting Google's datacenters across the planet, is highly reliable and can provide guaranteed network performance [4]. In addition to the performance advantage, each datacenter in the WAN can actually perform well as a router with its abundant resources [5].

Bearing these points in mind, we design a new type of service, inter-datacenter network as a service (iDaaS), for companies like Google and Microsoft, which deploy and host such private WANs [6]. In this setting, application providers acquire bandwidth from iDaaS providers for their wide area traffic. Fig. 1 plots an illustrative example for this novel service in a scenario of a single iDaaS provider and multiple application providers. Each application provider can send its bandwidth request to the closest datacenter via the front end server or border router offered by the iDaaS provider. On receiving the bandwidth request, iDaaS provider will open a tunnel and allocate the required bandwidth, such that the application provider's wide area traffic can be delivered along the inter-datacenter links with guaranteed bandwidth.

To enable the widely usage of this new type of service, we believe that a bandwidth market is essential between application providers and iDaaS providers. The commodities to be traded in such a market consist of a series of available bandwidth from many iDaaS providers, which can be reserved to guarantee the application performance. Consequently, an emerging *bandwidth pricing problem* dominates the utility of such a market. However, current pricing methods, i.e., the simple pay-as-you-go model based on the number of bytes transferred [7], are insufficient to characterize behaviors of bandwidth guarantee, as shown in Figs. 2 and 3.

Pricing in such a bandwidth trading market is the principal challenge. On the one hand, each iDaaS provider normally sets its bandwidth price per unit based on the total reserved

- *W. Li, K. Li, H. Qi, and J. Zhang are with the School of Computer Science and Technology, Dalian University of Technology, No 2, Linggong Road, Dalian 116023, China. E-mail: {liwenxin, zhangjh}@mail.dlut.edu.cn, {keqiu, hengqi}@dlut.edu.cn.*
- *D. Guo is with the College of Information System and Management, National University of Defense Technology, Changsha 410073, P.R. China. E-mail: guodeke@gmail.com.*

Manuscript received 12 Mar. 2015; revised 7 Oct. 2015; accepted 23 Nov. 2015. Date of publication 4 Dec. 2015; date of current version 13 June 2018. (Corresponding author: Heng Qi.)
Recommended for acceptance by I. Brandic.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2015.2505731

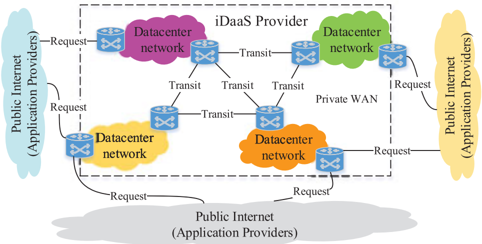


Fig. 1. An illustrative example of the new service iDaaS. There are multiple application providers and a single iDaaS provider, which hosts a private WAN connecting a large number of distributed datacenters.

bandwidth (workload) [8]. As iDaaS providers compete with each other for bandwidth demand with the aim of maximizing the revenue, the bandwidth price per unit of one iDaaS provider can be affected by others. On the other hand, given the bandwidth price, each application provider seeks the optimal bandwidth reservation strategy to minimize its payment and still desires to be completely served in terms of its demand. As such, each iDaaS provider's workload is determined by the application providers. This means that all application providers may together influence the price of an iDaaS provider. Based on the above guidelines, we believe that an efficient bandwidth pricing policy must benefit both iDaaS providers and application providers.

Accordingly, we are motivated to propose a global controller to make the price in the bandwidth trading market between iDaaS providers and application providers. Specifically, we model the interaction between iDaaS providers and application providers as a two-stage Stackelberg game [9]. In the first stage, iDaaS providers cooperate with each other in a Nash bargaining game, and make decisions on the price and size of bandwidth they are willing to allocate, based on the total bandwidth demand issued by all application providers. In the second stage, application providers compete with each other in a non-cooperative game, and decide on how much bandwidth they will reserve from each iDaaS provider. Theoretical analysis shows that there exists a unique Nash equilibrium (NE) in the non-cooperative game.

In order to compute the bandwidth price and bandwidth reservation in an efficient way, we propose a bandwidth pricing algorithm by blending the advantage of the demand segmentation method and the geometrical Nash bargaining solution [10]. For comparison, we particularly present two bandwidth reservation algorithms, which reserve each iDaaS provider's bandwidth in the weighted fair manner and max-min fair manner, respectively. Finally, we use

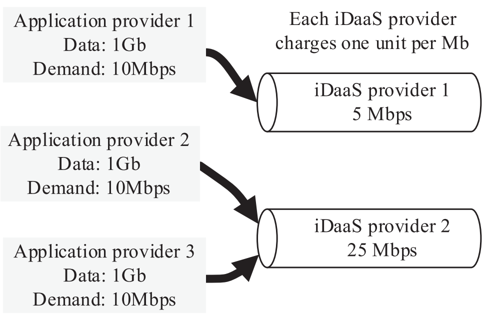


Fig. 2. When pricing based on the number of bytes transferred, application provider 1 cannot get fully served, though it pays the same money as application provider 2 and 3.

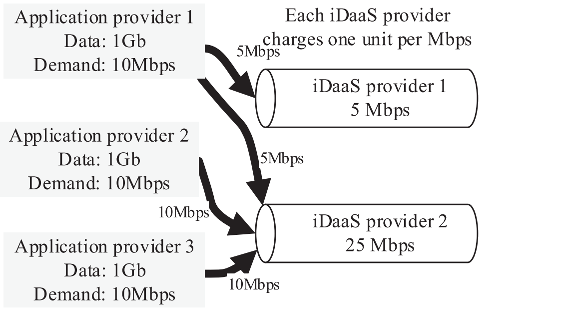


Fig. 3. When pricing the bandwidth guarantee, all the three application providers get fully served if we let half of the demand of application provider 1 be served by iDaaS provider 2.

comprehensive trace-driven simulations to demonstrate the efficiency of our algorithms, in the market of multiple iDaaS providers and application providers.

The major contributions of this paper are as follows:

- We make the first attempt to propose a new type of service—inter-datacenter network as a service, for Internet giants (iDaaS providers) that host large private wide area network to connect their global datacenters. In particular, we study a bandwidth market consisting of multiple iDaaS providers and application providers, and concentrate on the essential bandwidth pricing problem.
- To benefit both iDaaS providers and application providers, we model the interaction between iDaaS providers and application providers as a two-stage Stackelberg game. It contains a cooperative game among iDaaS providers and a non-cooperative game among application providers. We perform a theoretical analysis with respect to the Nash equilibrium of the non-cooperative game.
- We design an efficient bandwidth pricing algorithm based on the geometrical Nash bargaining solution and demand segmentation method. We further put forward a weighted fair bandwidth reservation algorithm and a max-min fair bandwidth reservation algorithm.
- We conduct comprehensive trace-driven experiments. The experimental results verify the efficiency of our algorithms in terms of both iDaaS provider's revenue, and application provider's utility of getting fully served with less payment. In addition, the evaluation results show that the per unit bandwidth price decreases as the bandwidth demand increases, which are close to real-life situations.

The rest of this paper is organized as follows. In Section 2, we discuss the background and present our system model. In Section 3, we apply a two-stage Stackelberg game to model the interaction between providers and application providers. In Section 4, we present our proposed bandwidth pricing and bandwidth reservation algorithms. In Section 5, we present the performance evaluation. Related work are presented in Section 6. Finally, Section 7 conclude this paper.

2 Background and System Model

2.1 Background

It is well-known that existing Internet suffers from the reliability and network performance issues [12], [13]. In spite of

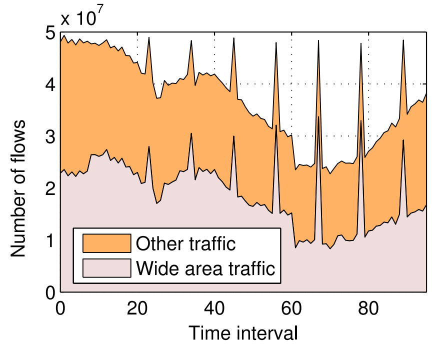


Fig. 4. The number of flows for the extracted wide area traffic in the Yahoo! datasets [11].

TABLE 1
Notations and Definitions

Notation	Definition
$\mathcal{N}$	the set of iDaaS providers, which are indexed by $i = 1, 2, \ldots, N$
$\mathcal{M}$	the set of application providers, which are indexed by $j = 1, 2, \ldots, M$
v_i	the WAN bandwidth capacity of iDaaS provider i
$x_{i,j}$	the amount of bandwidth reserved by application provider j from iDaaS provider i
$P_i(\cdot)$	the bandwidth price per unit of iDaaS provider i, which is a function of $x_i = \sum_{j \in \mathcal{M}} x_{i,j}$
d_j	the bandwidth demand of application provider j
$Q_i(\cdot)$	the utility function of iDaaS provider i, which is a function of x_i
$U_j(\cdot, \cdot)$	the utility function of application provider j, which is a function of $x_{i,j}$ and x_i

this, large-scale application providers are still relying on such unreliable Internet for their wide area traffic, which can result in highly unpredictable network performance. Benson et al. recently reported that such wide area traffic accounts for 40-90 percent of the total traffic in a typical business [14]. A recent survey further highlights that the requirement of such wide area traffic will double or triple in the next two to four years [3].

To have an intensive understanding of the application provider's wide area traffic, we detailed analyze some network datasets provided by Yahoo! [11]. Such datasets not only contain traffic between Yahoo! datacenters and clients (D2C traffic), but also contain traffic between different Yahoo! datacenters (D2D traffic). Each record in the NetFlow data includes the following fields: 1) timestamp, 2) source and destination IP address, 3) source and destination port, 4) protocol, 5) number of packets and bytes transferred from the source to the destination. Since all IP addresses in the datasets are permuted to hide the identities of Yahoo! application providers, we analyze the Yahoo! datasets based on the study of [2]. They reported that there are 17 popular server ports for the D2C traffic. For the detailed information of these 17 ports, we refer readers to the TABLE III in [2]. Here, we simply extract the traffic, not passing through those 17 ports, to denote the wide area traffic. Accordingly, we plot the number of flows for the extracted wide area traffic in a 96-interval period of time (each interval is 15 minutes), as shown in Fig. 4. We can easily check that the extracted wide area traffic accounts for around 50 percent on average throughout the day.

In view of the enormous, and rapidly growing wide area traffic, the phenomenon of unpredictable performance can further be worse in the current Internet. In contrast, we find that private wide area networks deployed by some Internet giants among their geographically distributed datacenters, i.e., Google B4 [4], are highly reliable and can provide guaranteed network performance. In addition, most of the private WAN links are provisioned to 30-40 percent average utilization. Even in the Google B4, the average WAN link utilization is 70 percent [4]. This implies that those Internet giants can always have some redundant bandwidth, which can be rent out to application providers with large amount of wide area traffic. Moreover, the private WAN link bandwidth utilization could further be improved if those Internet giants support the iDaaS service. As such, datacenters only need to access the traffic, push them to the private wide area network, and finally forward them to the destinations. Each datacenter is exactly similar to a router [5]. Moreover, the inter-datacenter optimization [15], [16] and the SDN technique [4] can ensure that the incoming wide area traffic can efficiently be isolated, and will not hinder the normal transmission of the already existed traffic. Hence, we are motivated to consider that private wide area networks owned by some Internet giants can be offered as a service to application providers which have large amount of wide area traffic to be transmitted. Looking ahead, we believe that such efforts may force more companies to provide such kind of services. In the following, we focus on characterize the interactions among multiple iDaaS providers and application providers under this type of services.

2.2 System Model

In this new type of service, iDaaS provider will open tunnels and allocate the required bandwidth for application providers, by using the SDN technique in the private WAN, i.e., *B4* [4]. As such, each iDaaS provider hosts a large amount of WAN transit bandwidth, and sells them to the application providers at a certain price, with the aim of maximizing its own revenue. The application provider sends its bandwidth demand, and buys the WAN bandwidth from multiple iDaaS providers with the aim of getting fully served in a low price. To make appropriate decisions for both iDaaS providers and application providers, we believe that the global network controlling is need. Hence, we are motivated to resort to the SDN controller and design a bandwidth trading market consisting of multiple application providers, iDaaS providers and a controller. Important notations used throughout this paper are listed in Table 1.

2.2.1 The Controller, iDaaS Providers and Application Providers

The controller. It is important to keep in mind that application providers mainly resort to the private WANs for their wide traffic in the iDaaS service model. In other words, application providers actually do not need to move their applications to the public cloud hosted on the data centers. In such a case, application providers can have choices to

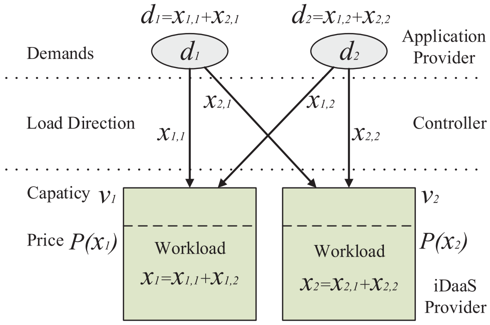


Fig. 5. A system of two iDaaS providers and two application providers.

select iDaaS providers for transferring data among the private WANs. To guide such selection, we believe that an economically free agent or broker is needed in the market of multiple iDaaS providers and multiple application providers. The broker mainly acts as a role of global regulation. More precisely, it receives the bandwidth demand of application providers in a time period. Based on the demands, it then computes and maintains stable pricing strategies of the iDaaS providers. Finally, it reserves bandwidth for the application providers. For the newly coming bandwidth demands, the broker computes the new price for such demands, but keeps the current price for existing demands. Actually, to perform such price computation, the broker can deploy some inter-connectors between iDaaS providers and application providers, i.e., Zimory, an emerging intermediator connecting buyers and sellers of resources [17]. Optionally, the broker can also deploy some SDN controllers to get a central control of the network [35], and gather information (i.e., bandwidth capacity, bandwidth demand) of both application providers and iDaaS providers. In this paper, we generally refer the inter-connectors as the controller, which is shown in Fig. 5.

iDaaS providers. We consider that there are N iDaaS providers, which are denoted as $\mathcal{N} = \{1, 2, \ldots, N\}$. Each iDaaS provider i offers v_i amount of bandwidth, which is available for reservation. Each iDaaS provider sets a bandwidth price per unit it will charge from application providers. The bandwidth price per unit is normally a function of the total requirement of reserved bandwidth on the iDaaS provider [8]. To better indicate this, we let $P_i(x_i)$ denote the bandwidth price per unit of iDaaS provider i, where x_i represents the total reserved bandwidth on the ith iDaaS provider.

Application providers. Similarly, we consider M application providers, which are denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$. Each application provider has a bandwidth demand d_j, which can be guaranteed by multiple iDaaS providers. Let $x_{i,j}$ represent the amount of bandwidth that application provider j reserved from iDaaS provider i. Recall that the total reserved bandwidth of iDaaS provider i can now be expressed as $x_i = \sum_{j\in\mathcal{M}} x_{i,j}$. In the following, we define the bandwidth reservation strategy and policy for application providers.

Definition 1. *A* bandwidth reservation strategy *relative to application provider j is $\boldsymbol{x_j} = \{x_{1,j}, \ldots, x_{N,j}\}$. The collection of bandwidth reservation strategies $\boldsymbol{x} = \{\boldsymbol{x_j} : 1, \ldots, M\}$ forms a* bandwidth reservation policy.

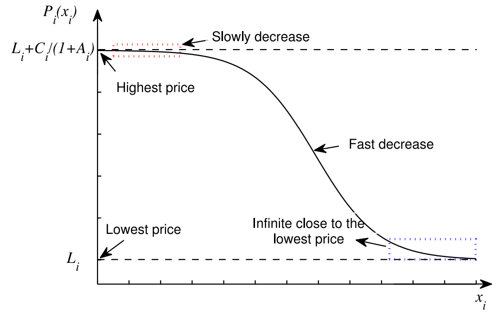


Fig. 6. Key mechanisms and benefits of the Logistic-like pricing.

2.2.2 Pricing Bandwidth Reservation

Each iDaaS provider i charges application providers some fee for accommodating a certain amount of bandwidth according to some bandwidth pricing strategy $P_i(\cdot)$. So, we are now in a position to design an efficient and practical bandwidth pricing function that is suitable for real life scenarios.

It is well-known that in today's Internet transit market, there are many pricing models applying the blended rate to enable the discount for user's traffic, i.e., tiered pricing [18]. Motivated by such pricing models in the Internet, our intuition for pricing the bandwidth reservation are two folds: 1) the more one application provider buys the bandwidth from an iDaaS provider, the lower the per unit price; 2) the per unit bandwidth price should be limited between a minimum value and a maximum value, such that both iDaaS providers and application providers can enjoy some benefit. Hence, we innovate by using a smooth but fast-decreasing function as the per unit bandwidth price, with gradual change rate at the beginning (the highest price) and around the stable value (the lowest price). This is exactly the situation in the Logistic-like function [19], where we can model the per unit bandwidth pricing function $P_i(x_i)$ as:

$$P_i(x_i) = L_i + \frac{C_i}{1 + A_i e^{B_i x_i}}, \tag{1}$$

where $L_i > 0$, $A_i > 0$, $0 < B_i < 1$, $C_i > 0$ are parameters specified by iDaaS provider i. Note that by setting different values for these parameters, different goals can be achieved by the iDaaS providers. For example, iDaaS providers may increase the value of the lowest price (L_i) due to the relatively high maintaining or operating cost for their private WANs. iDaaS provider may also increase the value for both A_i and B_i, or decrease the value for C_i to enable a bigger discount for the application providers. So, based on such pricing model, some other factors like the maintenance cost could further be considered by configuring the corresponding parameters.

Fig. 6 shows an illustrative example of a Logistic-like pricing function. We can find that such $P_i(x_i)$ can be controlled to slowly decrease at the beginning, fast decrease in the middle, and finally infinitely close to a stable value. Clearly, the price $P_i(x_i)$ is maintained between the lowest price (L_i) and the highest price ($L_i + \frac{C_i}{1+A_i}$). Such properties imply that the more application providers buy the bandwidth, the lower the per unit price, but not be free. In the following, we formally define the bandwidth pricing strategy and policy as follows:

Definition 2. A bandwidth pricing strategy $P_i(\cdot)$ *relative to provider i is a decreasing function* $P_i(x_i)$ *of* $x_i \in [0, v_i]$, *which is defined in Eq. (1). The collection of bandwidth pricing strategies* $\{P_i(\cdot) : i = 1, \ldots, N\}$ *forms a* bandwidth pricing policy.

Discussions. In addition to the bandwidth pricing and reservation, we discuss a critical issue on how to access for this new type of service (iDaaS). As we know, each datacenter has a border router to connect the Internet Service Provider to reach its clients, or connect to other datacenters [2]. Actually, these border routers are exactly similar to the Network Access Points (NAPs) [20]. Moreover, some providers also deploys metro-fiber to enable high access speed [21]. Therefore, application providers can use either private metro-fiber or the public Internet to access to those border routers, and accordingly transfer their data on the private WANs.

3 Interactions between iDaaS Providers and Application Providers: A Stackelberg Game Model

In this section, we define the utility function for both iDaaS providers and application providers. We then formulate the interactions between iDaaS providers and application providers as a Stackelberg game and analyze the existence and uniqueness of the equilibrium.

Based on the model described in Section 2, we consider that both iDaaS providers and application providers are selfish. Each iDaaS provider has the right to decide its bandwidth price, so as to maximize its own utility in terms of its own revenue. Each application provider, compete with other application providers to decide only the payment they are willing to make, with the aim of getting fully served without making too much payment. Therefore, it is a typical two-stage leader-follower game, which can be analyzed under the Stackelberg game framework [9]. Each iDaaS provider, the leader of the game, optimizes its strategy, based on the knowledge of the total bandwidth demand of followers (application providers).

3.1 Utility Functions

The utility function of an iDaaS provider is defined to be the sum of the revenue it collects from the application providers, and is calculated as follow

$$Q_i(x_i) = P_i(x_i) \sum_{j \in \mathcal{M}} x_{i,j}. \tag{2}$$

Since application provider competes with each other, the utility change of one application provider is likely to cause the utility changes of other application providers. For this reason, let $U_j(x_{i,j}, x_i)$ denote the utility of application provider j, where $x_{i,j}$ and x_i are two arguments. In this paper, we consider that a selfish application provider always *1) expects to get fully served in terms of its demand*, and *2) tries to reduce the bandwidth reservation price it has to pay*. More precisely, we define the utility of application provider j to be the negative value of the sum of its payment to each iDaaS provider, $U_j(x_{i,j}, x_i) = -\sum_{i \in \mathcal{N}} P_i(x_i) x_{i,j}$. Note that each application provider $j \in \mathcal{M}$ has a strong desire of getting the guaranteed bandwidth, i.e., its bandwidth demand should be fully satisfied. Hence, each application provider involves in solving the following utility maximization problem,

$$\max_{x_{i,j}} \quad U_j(x_{i,j}, x_i) \quad s.t. \quad \sum_{i \in \mathcal{N}} x_{i,j} = d_j. \tag{3}$$

In this way, each application provider's requirement of *1)* and *2)* can be met.

3.2 Maximizing iDaaS Providers' Utilities

Each leader of the Stackelberg game, iDaaS provider, optimizes its bandwidth pricing strategy in order to maximize its revenues according to Eq. (2), being aware of the total bandwidth demand of application providers. In this paper, we consider a Nash bargaining game, which can be described as follow. The total bandwidth demand $D = \sum_j d_j$ can be viewed as the commodity, while N iDaaS providers are players competing for the bandwidth demand. Each player enters the game with an utility function, which is described by Eq. (2). All the players cooperate in this game to achieve a win-win solution, in which the products of utility gains of all players are maximized

$$\max_{x_i} \quad \prod_{i \in \mathcal{N}} Q_i(x_i) \; s.t. \quad x_i \le v_i, \forall i \in \mathcal{N}; \text{and} \; \sum_{i \in \mathcal{N}} x_i = D. \tag{4}$$

Note that the first constraint means that the total reserved bandwidth on each iDaaS provider cannot exceed its bandwidth capacity. The second constraint means that the total reserved bandwidth should equal to the total demand of application providers. Once Eq. (4) is solved, the bandwidth price of each iDaaS provider can be computed, and thus can be announced in the bandwidth market.

3.3 Competition among Application Providers

Given the price and amount of bandwidth that iDaaS providers are willing to share, each application provider will seeks the optimal bandwidth reservation strategy to maximize its own utility. As indicated in Eq. (3), to maximize the utility, each application provider actually seeks iDaaS providers with low bandwidth price to reserve as much bandwidth as possible, as long as it does not exceed the bandwidth demand. Therefore, multiple application providers are competing for the bandwidth on iDaaS providers with low price. This implies that the utility of one application provider is likely to affect that of other. So, we are actually faced with a noncooperative game [22] since application providers are selfish. In this paper, we are interested in the Nash bargaining solution of the game. In other words, we seek an optimal bandwidth reservation policy, such that no application provider can improve its own utility by unilaterally changing its own reservation strategy. In the following, we formally define such an optimal bandwidth reservation policy.

Definition 3. *A bandwidth reservation policy* $\boldsymbol{x}^*$ *is the Nash equilibrium if, for all* $j \in \mathcal{M}$, *the following conditions holds:*

$$U_j(x^*_{i,j}, x^*_i) = \max_{x_{i,j}} \quad U_j(x_{i,j}, x_{i_{-j}} + x_{i,j}), \tag{5}$$

where $x_{i_{-j}}$ *is the other application providers' bandwidth reservation strategies except the jth application provider.*

The following theorem shows that there is an unique NE in the game between application providers in the following theorem.

Theorem 1 (Existence and uniqueness). *A unique Nash equilibrium point exists, if $A_i > 1, \forall i$ and for all $i \in \mathcal{N}$ and $j \in \mathcal{M}$, the following equation is satisfied*

$$\frac{1}{B_i}\left(2 + \frac{4}{A_i e^{B_i x_i} - 1}\right) < x_{i,j}. \tag{6}$$

Proof. Let $G_j(x_{i,j}, x_i)$ be the first-order derivative of $U_j(x_{i,j}, x_i)$ with respect to $x_{i,j}$. We can take the first order derivative of $G_j(x_{i,j}, x_i)$ with respect to $x_{i,j}$ as follow

$$\begin{aligned}\frac{\partial G_j(x_{i,j}, x_i)}{\partial x_{i,j}} &= -x_{i,j}\frac{\partial^2 P_i(x_i)}{\partial^2 x_{i,j}} - 2\frac{\partial P_i(x_i)}{\partial x_{i,j}} \\ &= \frac{A_i B_i C_i e^{B_i x_i}(A_i e^{B_i x_i}(2 - B_i x_{i,j}) + 2 + B_i x_{i,j})}{(1 + A_i e^{B_i x_i})^3}.\end{aligned} \tag{7}$$

Based on the guidelines of [23], a Nash equilibrium exists when $U_j(x_{i,j}, x_i)$ is continuous in x_i and concave in $x_{i,j}$. Obviously, $R_j(x_{i,j}, x_i)$ is continuous in x_i. We only need to prove that $U_j(x_{i,j}, x_i)$ is concave in $x_{i,j}$. This means that $\frac{\partial G_j(x_{i,j}, x_i)}{\partial x_{i,j}} < 0$. Since A_i, B_i, C_i are positive values and $A_i > 1$. Then, to maintain the concavity of $U_j(x_{i,j}, x_i)$, the following equation should be satisfied

$$\begin{aligned}&A_i e^{B_i x_i}(2 - B_i x_{i,j}) + 2 + B_i x_{i,j}) < 0 \\ &\Longrightarrow x_{i,j} > \frac{2A_i e^{B_i x_i} + 2}{A_i B_i e^{B_i x_i} - B_i} = \frac{1}{B_i}\left(2 + \frac{4}{A_i e^{B_i x_i} - 1}\right).\end{aligned} \tag{8}$$

So, $U_j(x_{i,j}, x_i)$ is concave in $x_{i,j}$ and the existence of the Nash equilibrium is thus proved.

Having settled the question of existence of the NE, we now establish the uniqueness of the NE. To this end, we first need to understand the monotonicity of $G_j(x_{i,j}, x_i)$. Clearly, $G_j(x_{i,j}, x_i)$ is decreasing in $x_{i,j}$ since its second-order derivative is less than 0, which has been proved above. Similar to the proof of $G_j(x_{i,j}, x_i)$ monotonicity in $x_{i,j}$, we find that $G_j(x_{i,j}, x_i)$ is also decreasing in x_i by Eq. (6).

Now, let $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ be two NEs. By applying Kuhn-Tracker conditions, we have

$$G_j(\hat{x}_{i,j}, \hat{x}_i)\begin{cases} = \lambda_j & \text{if } \hat{x}_{i,j} > 0, \\ < \lambda_j & \text{if } \hat{x}_{i,j} = 0, \end{cases} \tag{9}$$

and

$$G_j(x_{i,j}, x_i)\begin{cases} = \hat{\lambda}_j & \text{if } x_{i,j} > 0, \\ < \hat{\lambda}_j & \text{if } x_{i,j} = 0, \end{cases} \tag{10}$$

where λ_j and $\hat{\lambda}_j$ are the corresponding Lagrange multipliers. Now we need to prove that $\boldsymbol{x} = \hat{\boldsymbol{x}}$, i.e., for every $i, j, x_{i,j} = \hat{x}_{i,j}$.

To this end, we first prove that for each i and j, the following relations hold:

$$\{\hat{\lambda}_j \geq \lambda_j, \hat{x}_i \geq x_i\} \Rightarrow \hat{x}_{i,j} \leq x_{i,j}, \tag{11}$$

$$\{\hat{\lambda}_j \leq \lambda_j, \hat{x}_i \leq x_i\} \Rightarrow \hat{x}_{i,j} \geq x_{i,j}. \tag{12}$$

We only prove Eq. (11), since Eq. (12) is symmetric. Note that Eq. (11) holds trivially if $\hat{x}_{i,j} = 0$. Otherwise, we have the following equation by applying that $G_j(x_{i,j}, x_i)$ is decreasing in both $x_{i,j}$ and x_i,

$$G_j(\hat{x}_{i,j}, \hat{x}_i) = \hat{\lambda}_j \geq \lambda_j \geq G_j(x_{i,j}, x_i) \geq G_j(x_{i,j}, \hat{x}_i). \tag{13}$$

Since $G_j(x_{i,j}, x_i)$ is decreasing in $x_{i,j}$, we have $\hat{x}_{i,j} \leq x_{i,j}$. For symmetric reason, we can also obtain $\hat{x}_{i,j} \geq x_{i,j}$.

Now, we let $\mathcal{N}_1 = \{i : \hat{x}_i > x_i\}$ and $\mathcal{M}_1 = \{j : \hat{\lambda}_j < \lambda_j\}$, such that $\mathcal{N}_2 = \mathcal{N} - \mathcal{N}_1 = \{i : \hat{x}_i \leq x_i\}$. Assume that $\mathcal{N}_1$ is nonempty. Recalling that $\sum_i \hat{x}_{i,j} = \sum_i x_{i,j} = d_j$, it follows by Eq. (12) that for each j in $\mathcal{M}_1$,

$$\sum_{i \in \mathcal{N}_1} \hat{x}_{i,j} = d_j - \sum_{i \in \mathcal{N}_2} \hat{x}_{i,j} \leq d_j - \sum_{i \in \mathcal{N}_2} x_{i,j} = \sum_{i \in \mathcal{N}_1} x_{i,j}. \tag{14}$$

Note that Eq. (11) implies that $\hat{x}_{i,j} \leq x_{i,j}$ for each $i \in \mathcal{N}_1$ and $j \notin \mathcal{M}_1$. So, we can now get that

$$\sum_{i \in \mathcal{N}_1} \hat{x}_i = \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{N}_1} \hat{x}_{i,j} \leq \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{N}_1} x_{i,j} = \sum_{i \in \mathcal{N}_1} x_i. \tag{15}$$

This inequality significantly contradicts the definition of $\mathcal{N}_1$. This means that $\mathcal{N}_1$ is an empty set. Similarly, it also can be concluded that the set $\{i : \hat{x}_i < x_i\}$ is empty. This implies that

$$\hat{x}_i = x_i, \text{for every } i \in \mathcal{N}. \tag{16}$$

We now show that $\hat{\lambda}_j = \lambda_j$ for each application provider j. To achieve this gogal, Eq. (11) can be strengthened as follow

$$\begin{aligned}\{\hat{\lambda}_j > \lambda_j, \hat{x}_i = x_i\} &\text{ implies that either} \\ \hat{x}_{i,j} < x_{i,j}, &\text{ or } \hat{x}_{i,j} = x_{i,j} = 0.\end{aligned} \tag{17}$$

Note that the implication is trivial if $\hat{x}_{i,j} = 0$. If $\hat{x}_{i,j} > 0$, it follows similarly to Eq. (13) that $G_j(\hat{x}_{i,j}, \hat{x}_i) > G_j(x_{i,j}, \hat{x}_i)$. Hence, $\hat{x}_{i,j} < x_{i,j}$.

Suppose that $\hat{\lambda}_j > \lambda_j$ for some $j \in \mathcal{M}$. Since $\sum_{i \in \mathcal{N}} \hat{x}_{i,j} = d_j > 0$, then $\hat{x}_{i,j} > 0$ for at least one i and Eq. (13) implies that $\sum_{i \in \mathcal{N}} x_{i,j} > \sum_{i \in \mathcal{N}} \hat{x}_{i,j} = d_j$, which contradicts the demand constraint for application provider j. Therefore, we conclude that $\hat{\lambda}_j > \lambda_j$ does not hold for any application provider j. Similarly, $\hat{\lambda}_j < \lambda_j$ cannot hold as well. Thus, $\hat{\lambda}_j = \lambda_j$ for every application provider $j \in \mathcal{M}$.

Combined with Eq. (16), we can conclude that $\hat{x}_{i,j} = x_{i,j}$ for each i, j, by Eq. (11) and Eq. (12). Hence, the uniqueness of the NE is thus proved. □

4 Implementation Algorithms

In this section, we first present an efficient pricing algorithm, and then present two bandwidth reservation algorithms.

4.1 Price Computation

As aforementioned, given the total bandwidth demand of application providers, iDaaS providers are actually playing a Nash Bargaining game where players (iDaaS providers)

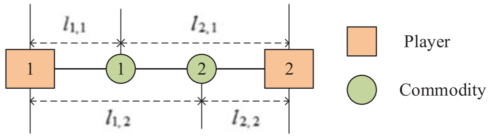


Fig. 7. An illustrative example of 2-player geometrical game.

cooperate with each other to achieve a win-win solution, such that each player gains the maximum utility. Note that the problem in Eq. (4) is usually approximately solved by numerical methods like gradient projection method [24]. Such methods, however, require numerous iterations and are not practical to be implemented in real-world controller hosted by agents or brokers. For this reason, we are motivated to design an efficient and lightweight bandwidth pricing algorithm based on the geometrical representation of Nash bargaining games. Here, the concept of utility-distance product is introduced to unify iDaaS providers' utilities for a certain amount of demand. The computation overhead of the algorithm is significantly reduced.

4.1.1 A Primer on the Geometrical Game

A geometrical Nash bargaining game can be described in some low-dimensional space in the region of Euclidean Geometry. For example, Fig. 7 shows a two-dimensional spatial game, which represents a 2-player bargaining game. Utilities of the two players for commodities 1 and 2 are clearly denoted by the distance $l_{i,1}$ and $l_{i,2}$, where $i \in \{1, 2\}$ represents the player. In this example, commodities are presented as points based on their spatial proximities, which lie within the boundary enclosed by all players. The relative distance of player i for commodity k is defined as a function of the inverse of player i's utility compared to the sum of the inverse of all players' utilities, i.e.,

$$l_{i,k} = \frac{1/R_{i,k}}{\sum_i (1/R_{i,k})}, \forall i,\ \forall k, \tag{18}$$

where $R_{i,k}$ denotes player i's utility for commodity k in the geometrical game. We can find that the higher a player's utility for one commodity, the closer it is to the commodity. The distance of each commodity to all players are normalized, and they add up to a unitary value, i.e,, $\sum_i l_{i,k} = 1, \forall k$. As demonstrated in Fig. 7, player 1 prefers to commodity 1, while player 2 prefers to commodity 2.

In the bargaining process, each player selects commodities based on their relative distances to him. Accordingly, commodities with higher utility will be selected with a higher priority. Usually, the utility-distance product is similar to the moment of force in a lever system. The utility-distance product of a player to a commodity is defined as

$$\phi_{i,k} = l_{i,k} \cdot R_{i,k}. \tag{19}$$

Fig. 8 shows an example of finding the pivot point in a 2-player game. In this example, each player sorts all the commodities based on their distances to it. In a typical lever system, weights are aligned along the lever, such that the collective moment generated by weights on the left hand side equals to that on the right hand side. Similar, in order to achieve equilibrium in the geometrical game, the sum of utility distance product should be equally partitioned among all players. In Fig. 8, players 1 and 2 are lying at the two end points of the lever, where the utility distance products of commodities are regarded as the force moments.

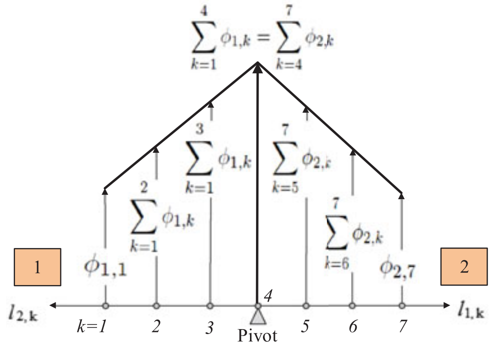


Fig. 8. Finding the pivot point in a 2-player game, which is similar to a lever system.

We can easily find that pivot point of the lever in this example should be lying on point 4, where the collective moment $\sum_{k=1}^{4} \phi_{1,k} = \sum_{k=4}^{7} \phi_{2,k}$. As a matter of fact, the pivot point in the two-dimensional geometrical game is determined by balancing the moments between two players:

$$\mu = \frac{1}{2} \sum_{k \in \mathcal{K}} \phi_{i,k}. \tag{20}$$

After determining the pivot point, the bargaining solution is to assign commodities lying on the left side to player 1, and commodities lying on the right side to player 2.

4.1.2 Algorithm Design

Our problem in Eq. (4) cannot be directly solved by applying such a geometrical game due to the following fact. The commodity in the Nash bargaining game between iDaaS providers is the total bandwidth demand issued by all the application providers, which means that there is only one commodity. Simply solving Eq. (4) by a geometrical game is likely to result in a case where there is only one player gets the commodity and accommodates all the bandwidth demand from application providers. Clearly, this result gravely contradicts a win-win Nash bargaining solution, since each iDaaS provider can actually get a portion of the total demand.

To address this challenge, we apply the demand segmentation method. We split the commodity into K sub-commodities, where K is a number of infinity. All the sub-commodities constitute a set $\mathcal{K} = \{1, \ldots, K\}$, where each sub-commodity k occupies a portion of the total demand. Here, we consider each sub-commodity has a same demand, i.e., $r_k = \frac{D}{K}, \forall k$. Therefore, we now can solve Eq. (4) by applying a multi-player geometrical game, which is described as follow: N iDaaS providers are viewed as players, who are competing for K sub-commodities. Since players differ in their utilities for different sub-commodity, player i's utility for sub-commodity k is defined as follow:

$$Q_i(r_k) = P(r_k) r_k. \tag{21}$$

Extending the definitions in the above 2-players game, we get the distance and utility-distance product in our multi-player geometrical game. The distance between player i and sub-commodity k is defined as follow

$$l_{i,k} = \frac{1/Q_i(r_k)}{\sum_i (1/Q_i(r_k))}, \forall i \in \mathcal{N},\ \forall k \in \mathcal{K}. \tag{22}$$

We can easy check that the sum distance of all the players to each sub-commodity k is unitary 1, i.e., $\sum_{i\in\mathcal{N}} l_{i,k} = 1, \forall k$. The utility-distance product of player i to commodity k is defined as

$$\phi_{i,k} = l_{i,k} \cdot Q_i(r_k), \forall i \in \mathcal{N},\ \forall k \in \mathcal{K}. \tag{23}$$

In the ideal condition when all the players constitute a multi-dimensional lever system and all sub-commodities lie along the lever, the determination of the pivot location can be based on balancing the utility-distance product with respect to all players:

$$\mu_i = \frac{1}{N} \sum_{k\in\mathcal{K}} \phi_{i,k}, \forall i \in \mathcal{N}. \tag{24}$$

Theorem 2 shows the amount of bandwidth demand issued to each iDaaS provider in the multi-player geometrical game.

Theorem 2. *In the multi-player geometrical representation of our Nash bargaining game, the amount of demand that each iDaaS provider i gets should be $\min\{v_i, \frac{D}{N}\}$.*

Proof. Recall that we split the total bandwidth demand into K pieces, where K is a number of infinity. Therefore, when determining the pivot location by equally dividing the sum utility-distance product towards all players, the mean utility-distance product relative to player i can actually be obtained by finding its limit value associated K

$$\begin{aligned}\mu_i &= \frac{1}{N}\lim_{K\to\infty}\sum_{k=1}^{K}\phi_{i,k} = \frac{1}{N}\lim_{K\to\infty}\sum_{k=1}^{K}\frac{1}{\sum_{i=1}^{N}\frac{1}{r_k\left(L_i+\frac{C_i}{1+A_i e^{B_i r_k}}\right)}}\\ &= \frac{1}{N}\lim_{K\to\infty}\sum_{k=1}^{K}\frac{1}{\sum_{i=1}^{N}\frac{1}{\frac{D}{K}(L_i+\frac{C_i}{1+A_i e^{B_i\frac{D}{K}}})}} = \frac{1}{N}\frac{1}{\sum_{i=1}^{N}\frac{1}{D(L_i+\frac{C_i}{1+A_i})}}.\end{aligned} \tag{25}$$

In the ideal condition, each iDaaS provider can actually get a portion of the total demand. This means that each iDaaS provider can get some sub-commodities in $\mathcal{K}$. Let $0 < \hbar_i < 1$ denote the portion that iDaaS provider i obtains from sub-commodities set $\mathcal{K}$, such that the number of sub-commodities that occupied by iDaaS provider i is $\hbar_i K$. The cumulative utility-distance product on iDaaS provider i's side can be calculated as

$$\begin{aligned}\lim_{K\to\infty}\sum_{k=1}^{\hbar_i K}\phi_{i,k} &= \lim_{K\to\infty}\sum_{k=1}^{\hbar_i K}\frac{1}{\sum_{i=1}^{N}\frac{1}{\frac{D}{K}(L_i+\frac{C_i}{1+A_i e^{B_i\frac{D}{K}}})}}\\ &= \hbar_i\frac{1}{\sum_{i=1}^{N}\frac{1}{D(L_i+\frac{C_i}{1+A_i})}}.\end{aligned} \tag{26}$$

Since the cumulative utility-distance product of one iDaaS provider is no larger than its mean utility-distance product, such that by applying the Nash bargaining solution in the multi-player geometrical game, we have

$$\hbar_i\frac{1}{\sum_{i=1}^{N}\frac{1}{D(L_i+\frac{C_i}{1+A_i})}} = \frac{1}{N}\frac{1}{\sum_{i=1}^{N}\frac{1}{D(L_i+\frac{C_i}{1+A_i})}} \Longrightarrow \hbar_i = \frac{1}{N}. \tag{27}$$

Finally, by adding up all the demands that each iDaaS provider occupies, we have

$$x_i = \hbar_i K r_k = \frac{1}{N}K\frac{D}{K} = \frac{D}{N}. \tag{28}$$

Since the total demand received by an iDaaS provider must be constrained by its bandwidth capacity v_i. Hence, x_i should be the minimum value between the bandwidth capacity v_i and $\frac{D}{N}$. Proved. □

Based on the guideline of Theorem 2, we can now design our bandwidth pricing algorithm which is simple and lightweight. The key idea is that the total bandwidth demands are allocated to iDaaS providers, based on their capacities and the average demand. If an iDaaS provider's capacity is less than the average demand, those unallocated demand should be equally allocated to other iDaaS providers. Therefore, our bandwidth pricing algorithm starts with sorting all the iDaaS providers in the increasing order of its capacity v_i. For each iDaaS provider i, the algorithm computes the demand that can be allocated to it. Once the demand allocating process is accomplished for one iDaaS provider, the average demand is updated. After the demand allocation is finished for all iDaaS providers, the final step is to compute the bandwidth price for each iDaaS provider, which is useful for application provider's bandwidth reservation. The bandwidth pricing algorithm based on the geometrical bargaining game is summarized in Algorithm 1.

Algorithm 1. Bandwidth Pricing Algorithm

Input:
Total bandwidth demand: D;
Bandwidth capacity: $v_i, \forall i \in \mathcal{N}$;
parameters: A_i, B_i, C_i, L_i, $\forall i \in \mathcal{N}$;
Output:
Price and reserved bandwidth $\{(P(x_i), x_i) : \forall i \in \mathcal{N}\}$
1: Sort all iDaaS providers in $\mathcal{N}$ in the increasing order of its bandwidth capacity v_i;
2: Initialize $D_{left} = D, N_{left} = N$;
3: **for** each iDaaS provider $i \in \mathcal{N}$ **do**
4: $mean \leftarrow \frac{D_{left}}{N_{left}}$;
5: $x_i \leftarrow \min\{v_i, mean\}$;
6: $D_{left} \leftarrow D_{left} - x_i$;
7: $N_{left} \leftarrow N_{left} - 1$;
8: Each iDaaS provider i computes its bandwidth price according to Eq. (1);

We can easily check that Algorithm 1 follows the guideline of Theorem 2. This implies that our algorithm can ensure a win-win solution for the Nash Bargaining game among iDaaS providers, which shows the effectiveness of our bandwidth pricing algorithm. Recall that our objective is to design a lightweight bandwidth pricing algorithm, such that it can simply be implemented in the controller. We find that the complexity of Algorithm 1 is only $O(N)$, where N is the number of iDaaS providers. This immediately shows the efficiency of our algorithm.

4.2 Bandwidth Reservation

Given the pre-computed bandwidth price and the amount of bandwidth that each iDaaS providers are willing to share, we can compute the bandwidth reservation strategy for each application provider. As described above, each application provider targets at maximizing its own utility while has a strong desire of getting fully served. The utility of one application provider can significantly affect that of others and each application provider wishes to reserve more bandwidth from iDaaS providers with low bandwidth price per unit. Thus, the amount of bandwidth of each iDaaS providers should be carefully allocated. In this paper, we consider two bandwidth reservation algorithms, the weighted fairness and max-min fairness.

Algorithm 2. Weighted Fair Bandwidth Reservation

Input:
Bandwidth demand: $d_j, \forall j \in \mathcal{M}$;
The amount of bandwidth that each iDaaS provider are willing to share: $x_i, \forall i \in \mathcal{M}$;
Output:
Bandwidth reservation variable: $x_{i,j}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M}$;
for each $i \in \mathcal{N}$ **do**
  for each $j \in \mathcal{M}$ **do**
    $x_{i,j} \leftarrow \frac{d_j}{\sum_j d_j} \cdot x_i$;
return $x_{i,j}, \forall i \in \mathcal{N},\ \forall j \in \mathcal{M}$;

The first bandwidth reservation algorithm is based on the idea of weighted fair bandwidth reservation, which is shown in Algorithm 2. The insight behind this algorithm is that each application provider reserves bandwidth from each iDaaS provider based on its relevant weight. The weight of application provider j is defined to be its demand compared to all application providers, i.e., $\frac{d_j}{\sum_j d_j}$, such that the sum weight is unitary 1. The rationale for weight is that the more an application provider demands, the more bandwidth it can reserve, and finally achieve the weighted fairness among application providers.

The second algorithm follows the max-min fair manner in reserving each iDaaS provider's bandwidth, which is summarized in Algorithm 3, with the computed bandwidth price and the amount of bandwidth that each iDaaS provider is willing to share. Algorithm 3 starts with sorting all of the iDaaS providers in the increasing order of its bandwidth price (Step 1), and then sorting all of the application providers in the increasing order of its bandwidth demand (Step 2). In Step 4-13, each application provider reserves $\min\{d_j, \frac{x_i}{\mathcal{M}_{left}}\}$ amount of bandwidth from each iDaaS provider until its bandwidth demand is fully satisfied. Once an application provider has finished its bandwidth reservation process, then the application provider should be removed, and thus the average bandwidth on each iDaaS provider that each application provider can reserved will be updated. We can easily check that each iDaaS provider's bandwidth is actually reserved in the max-min fair manner. That is, if the unsatisfied bandwidth demand of an application provider is less than the average bandwidth on an iDaaS provider, then the unreserved bandwidth of that iDaaS provider will be fairly reserved by other application providers.

Algorithm 3. Max-Min Fair Bandwidth Reservation

Input:
Bandwidth demand: $d_j, \forall j \in \mathcal{M}$;
Pre-computed parameters: $x_i, P_i(x_i), \forall i \in \mathcal{M}$;
Output:
Bandwidth reservation variable: $x_{i,j}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M}$;
1: Sort all iDaaS providers in $\mathcal{N}$ in the increasing order of the bandwidth price $P_i(x_i)$;
2: Sort all application providers in $\mathcal{M}$ in the increasing order of the bandwidth demand d_j;
3: Initialize $\mathcal{M}_{left} = \mathcal{M}$;
4: **for** each $j \in \mathcal{M}$ **do**
5:   **for** each $i \in \mathcal{N}$ **do**
6:     **if** $d_j \leq 0$ **then**
7:       $\mathcal{M}_{left} \leftarrow \mathcal{M}_{left} - 1$;
8:       **Break**;
9:     $x_{i,j} \leftarrow \min\{d_j, \frac{x_i}{\mathcal{M}_{left}}\}$;
10:     $d_j \leftarrow d_j - x_{i,j}$;
11:     $x_i \leftarrow x_i - x_{i,j}$;
12: **return** $x_{i,j}, \forall i \in \mathcal{N},\ \forall j \in \mathcal{M}$;

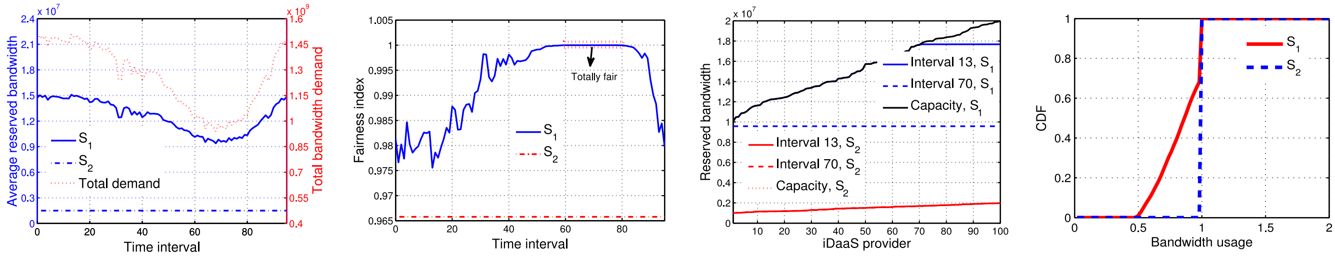
The max-min fair and the weighted fair bandwidth reservation algorithms are complementary. On the one hand, the max-min fair algorithm can guarantee more application providers to get fully served with respect to the corresponding bandwidth demand, compared to the weighted fair algorithm. In addition, application providers will pay less for the reserved bandwidth, if the max-min fair algorithm is applied. The main reason is that max-min fair algorithm reserves bandwidth in the increasing order of the associated bandwidth price. On the other hand, the weighted fair algorithm can reduce the variance of the payment among the application providers. This is because that each application provider actually reserves some bandwidth on each iDaaS provider.

5 Performance Evaluation

In this section, we use real-world traces to realistically evaluate the performance of our proposed algorithms.

5.1 Experiments Settings

In our experiment settings, there are 100 ($N = 100$) iDaaS providers. Each iDaaS provider hosts an amount of bandwidth, and sells them to potential application providers, according to its own bandwidth pricing strategy. In our experiments, we use unit to measure the bandwidth capacity of each iDaaS provider. We consider two scenarios (S_1 and S_2) where the bandwidth capacity v_i for each iDaaS providers are set as a random value within $[10^7, 2 \times 10^7]$ units and $[10^6, 2 \times 10^6]$ units, respectively. Note that such unit can be $1\,\text{kbps}$ in a commonly WAN bandwidth setting ($[1, 10]\,\text{Gbps}$)[3], and can even become $10 \sim 100\,\text{kbps}$ when iDaaS providers develop into a representative ISP that has



(a) Average reserved bandwidth
(b) Fairness index
(c) Reserved bandwidth at sampled intervals
(d) CDF of bandwidth usage across all iDaaS providers and all time intervals

Fig. 9. Reserved bandwidth on iDaaS providers for a 96-interval period of time, in terms of (a) average reserved bandwidth, (b) fairness index associated with the reserved bandwidth x_i, (c) the reserved bandwidth of each iDaaS provider at sampled intervals 13 and 70, (d) CDF of bandwidth usage across all iDaaS providers and all time intervals.

$100\,\mathrm{Gbps}$ amount of bandwidth [25]. Moreover, the total bandwidth of all iDaaS providers can accommodate the total bandwidth demand of application providers in scenario S_1, but not in scenario S_2. Without loss of generality, the minimum bandwidth price per unit L_i and the maximum per unit price associated parameter C_i are set to 0.01 and 1 for all iDaaS providers, respectively. Based on the guideline of Theorem 1 and in order to make the difference between iDaaS providers' price more clear, parameters A_i and B_i for all $i \in \mathcal{N}$ are set to be uniformly random within $[1, 100]$ and $[10^{-7}, 2\times 10^{-7}]$, respectively.

Datasets. Our experiments are conducted on Yahoo! network flow datasets [11]. These datasets, collected from Yahoo! border routers every 15-minute during one day, contain not only traffic between Yahoo! servers and client, but also contain traffic across different Yahoo! datacenters. In our experiments, we extract the traffic between different datacenters from seven frequently used D2D ports, which are described in [2]. The extracted inter-datacenter traffic is used to represent the wide area traffic. Although the wide area traffic we extracted actually is issued by only one application provider (Yahoo!), we believe that they can faithfully reflect the traffic demand distribution, and it is appropriate to use them for the purpose of benchmarking the performance of our algorithms. Each flow in the extracted traces is assumed to be issued by one application provider, and the bytes to be transferred are considered to be the bandwidth demand for that application provider. For example, if a flow needs to transfer 50 bytes from its source to destination, then we consider that the corresponding application provider demands 50 units of bandwidth.

5.2 Evaluation Results and Analysis

For the following experiments, we consider two scenarios S_1, S_2, the bandwidth capacity v_i for each iDaaS provider is set to be uniformly random within range $[10^7, 2\times 10^7]$ and $[10^6, 2\times 10^6]$, respectively.

5.2.1 *Reserved Bandwidth on iDaaS Providers*

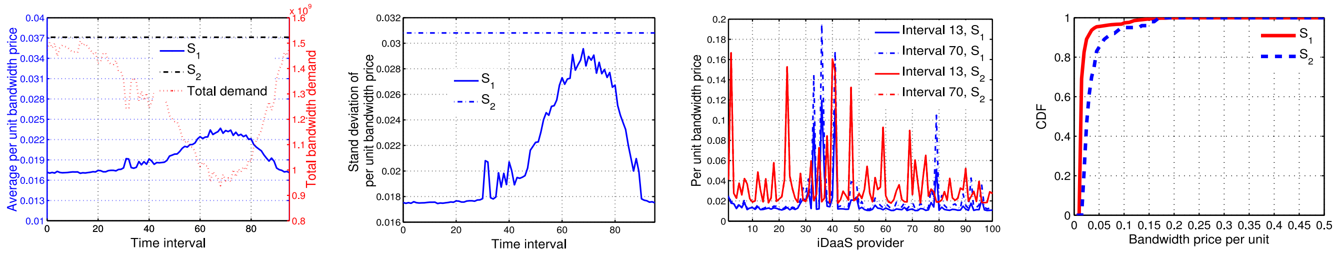
Fig. 9 first shows the reserved bandwidth on iDaaS providers for a 96-interval period of time. Fig. 9a plots the reserved bandwidth in both scenario S_1 and S_2 on average. Clearly, the average reserved bandwidth on iDaaS providers closely follows the change of the total demand in scenario S_1, while it maintains at a stable value in S_2. This is because that the sum bandwidth capacity of all iDaaS providers can accommodate the total bandwidth demand in each time interval for scenario S_1, while it cannot in S_2.

The most important performance metric in the Nash bargaining game among the iDaaS providers is fairness. As a quantitative evaluation, we use the Jain's fairness index [26], which is defined as $F = \frac{(\sum_{i=1}^{N} x_i)^2}{N\cdot\sum_{i=1}^{N} x_i^2}$. Fig. 9b shows the fairness index of the reserved bandwidth in the two scenarios S_1 and S_2. In comparison, S_1 is able to maintain a higher level of fairness in terms of the reserved bandwidth. Its fairness index can be up to 1 during the extremely low bandwidth demand period (interval 60-80). The root cause is that each iDaaS provider's bandwidth is totally reserved in each interval for scenario S_2, while it is not for S_1.

To precisely understand the reserved bandwidth, we also plot the reserved bandwidth on each iDaaS provider under two sampled intervals 13 and 70 in Fig. 9c. Interval 13 corresponds to a high demand interval, while interval 70 corresponds to a low demand interval. We can easily find that the reserved bandwidth on each iDaaS provider does not exceed its capacity for each sampled interval. In the scenario S_1, most iDaaS provider's bandwidth are totally reserved except some ones with high capacity at interval 13. Each iDaaS provider, however, maintains an equal amount of bandwidth that is reserved at interval 70. In the scenario S_2, each iDaaS provider's bandwidth is fully reserved at both sampled intervals. These results exactly verify the key idea of our algorithm 1. That is, the amount of bandwidth demand that each iDaaS provider gets should be the minimum value between the capacity and the mean demand, i.e., $\min\{v_i, \frac{D}{N}\}$. We further plot the CDF of bandwidth usage across all iDaaS providers and all time intervals in Fig. 9d. We observe that across all time intervals, the bandwidth usage of all iDaaS providers is higher than 46 percent in scenario S_1, while the bandwidth of each iDaaS providers is fully utilized in scenario S_2.

5.2.2 *Bandwidth Price per Unit of iDaaS Providers*

Fig. 10 describes the bandwidth price per unit of iDaaS providers in a 96-interval period of time under both scenarios S_1 and S_2. Fig. 10a first plots the average bandwidth price per unit across all iDaaS providers. It is clear that the average bandwidth price per unit in scenario S_1 is lower than that in S_2, and is completely opposite to the total bandwidth



(a) Average bandwidth price per unit

(b) Standard deviation of the bandwidth price per unit

(c) Bandwidth price per unit at sampled intervals

(d) CDF of bandwidth price per unit across all iDaaS providers and all time intervals

Fig. 10. Bandwidth price per unit of iDaaS providers for a 96-interval period of time, in terms of (a) average bandwidth price per unit, (b) standard deviation of the bandwidth price per unit across all iDaaS providers, (c) the bandwidth price per unit of each iDaaS provider at sampled intervals 13 and 70, (d) CDF of the bandwidth price per unit across all iDaaS providers and all time intervals.

demand. This implies that the more bandwidth demand, the lower the bandwidth price per unit. Note that the average bandwidth price per unit in S_2 is always maintained at a stable value. This is because that each iDaaS provider's bandwidth in scenario S_2 is fully reserved in all intervals.

To understand the difference of iDaaS providers' prices, we also plot the standard deviation of the bandwidth price per unit across all iDaaS providers in Fig. 10b. We observe that the standard deviation in S_2 is always higher than that in S_1. The main reason is that the reserved bandwidth on each iDaaS provider in S_2 is exactly its bandwidth capacity, while iDaaS providers usually differ in their bandwidth capacity. Recall that the lower bandwidth demand, the higher level of fairness in terms of the reserved bandwidth on iDaaS providers. Combined different parameters settings of iDaaS providers, L_i, A_i, B_i, C_i, we can infer that the standard deviation of bandwidth price per unit in scenario S_1 is completely opposite to the total demand.

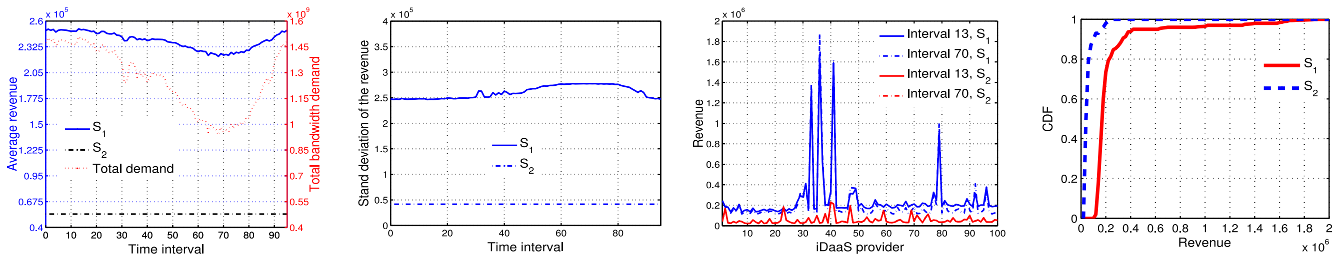
We also plot bandwidth price per unit for each iDaaS provider for both scenario S_1 and S_2 under two sampled intervals 13 and 70, as shown in Fig. 10c. Due to the limited bandwidth capacity in S_2, the bandwidth price per unit at interval 13 is identical with that at interval 70 in scenario S_2. We further observe the per unit bandwidth of most iDaaS providers in S_2 is higher than that in S_1. Moreover, in scenario S_1, most iDaaS providers maintain a lower bandwidth price per unit at interval 13 than that at interval 70. This further indicates that the more bandwidth one buys, the lower the bandwidth price per unit. We further plot the CDF of bandwidth price per unit across all iDaaS providers and all time intervals in Fig. 10d. We find that the bandwidth price in scenario S_2 is higher than that in scenario S_1 because that the blue curve (S_2) is lower than the red curve (S_1). The root cause is that the total bandwidth capacity in scenario S_1 is higher than that in scenario S_2, and thus the reserved bandwidth of each iDaaS provider in scenario S_1 can be relatively lower than that in scenario S_2.

5.2.3 Revenue of iDaaS Providers

Since each iDaaS provider seeks to maximize its own revenue, we plot the revenue of iDaaS providers across a 96-interval period of time in Fig. 11. We first show the average revenue across all iDaaS providers in Fig. 11a. Clearly, the average revenue in S_1 is always higher than that in S_2, and closely follows the total bandwidth demand.

Fig. 11b plots the standard deviation of the revenue across all iDaaS providers. We observe that S_2 always maintains a lower standard deviation of the revenue than S_1. This may be the case that a lower reserved bandwidth indicates a higher bandwidth price per unit on each iDaaS provider, and the product of these two values finally reduces the variance of the revenue. We further observe that the standard deviation of revenue at lower demand intervals is higher than that at higher demand intervals in scenario S_1.

In order to comprehensively understand iDaaS providers' revenues, we also plot the revenue of each iDaaS provider under the two sampled interval 13 and 70 for both scenario S_1 and S_2, as shown in Fig. 11c. Compared to S_2, S_1 achieves a higher revenue for most iDaaS providers. In addition, most iDaaS providers get a higher revenue at interval 13 than that at interval 70. This result means that the higher bandwidth demand application providers issue, the higher revenue the iDaaS providers gain. We further



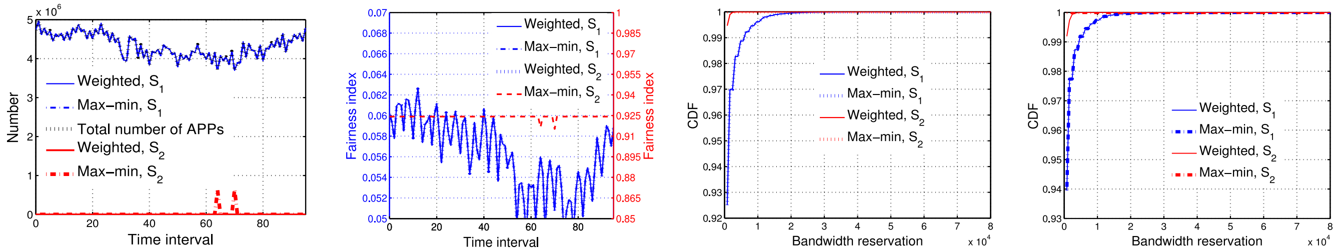
(a) Average revenue

(b) Standard deviation of the revenue

(c) Revenue at sampled intervals

(d) CDF of revenue across all iDaaS providers and all time intervals

Fig. 11. Revenue of iDaaS providers for a 96-interval period of time, in terms of (a) average revenue, (b) standard deviation of the revenue across all iDaaS providers, (c) the revenue each iDaaS provider at sampled intervals 13 and 70.



(a) The number of demand being fully satisfied application providers (b) Fairness index (c) CDF of bandwidth reservation at interval 13 (d) CDF of bandwidth reservation at interval 70

Fig. 12. Bandwidth reservation of application providers for a 96-interval period of time with (a) the number of demand being fully satisfied application providers, (b) Fairness index of bandwidth reservation across all application providers, and CDF of bandwidth reservation for application providers at interval 13(c) and interval 70(d).

plot the CDF of the revenue across all iDaaS providers and time intervals in Fig. 11d. It is clear that iDaaS providers in scenario S_1 achieves higher revenue than that in scenario S_2. More precisely, all iDaaS providers' revenue is less than $240{,}000$ in scenario S_2, while only 81.27 percent of iDaaS providers get an amount of revenue that is less than $240{,}000$.

5.2.4 Bandwidth Reservation of Application Providers

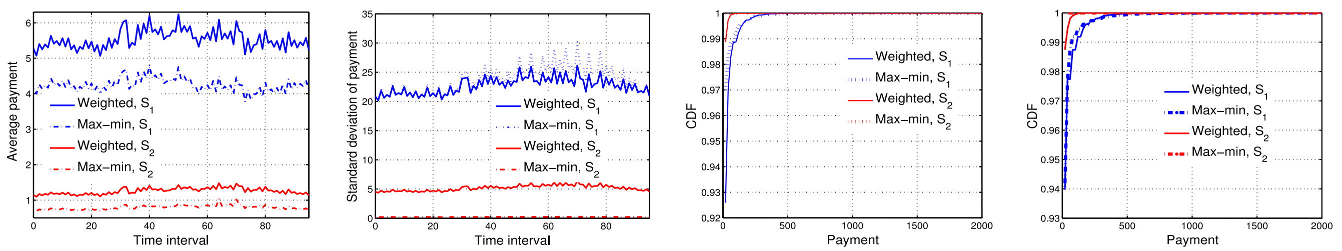
Fig. 12 first plots application providers' bandwidth reservation in a 96-interval period of time, where the number of application providers with bandwidth demand being fully satisfied, is shown in Fig. 12a. We can easily find that both the weighted fair and max-min fair reservation algorithms can satisfy all application providers' demands in scenario S_1, but not in scenario S_2. This is because that the amount of all bandwidth is lower than the total demand of application providers in scenario S_2. We further observe that the max-min fair can satisfy more demand than the weighted fair. This is because that each iDaaS provider's bandwidth is reserved by application providers based on the weight $\frac{d_j}{\sum_j d_j}$. In this way, such that the amount of reserved bandwidth by each application provider can be lower than its demand when the sum offered bandwidth by all iDaaS providers is less than the total demand of all application providers.

To study the fairness in the bandwidth reservation process for application providers, we also plot the fairness index of bandwidth reservation of application providers in Fig. 12b. Similar to the fairness index described above in Section 5.2, let $F = \frac{(\sum_{j=1}^{M} x_j)^2}{M \cdot \sum_{j=1}^{M} x_j^2}$ denote the fairness index associated with application provider's bandwidth reservation. We can find that both the weighted fair and max-min fair in S_1 exhibit the same fairness index. The root cause is that both of them fully satisfy the bandwidth demand of all application providers. We also find that the weighted fair in scenario S_2 completely achieves the same fairness as the weighted fair and max-min fair in S_1. This is because that the same ratio of the amount of reserved bandwidth to the demand is maintained for each application provider in the weighted fair manner in scenario S_2. In comparison, the max-min fair in scenario S_2 always achieves a higher fairness index than the above three.

We further plot the CDF of bandwidth reservation under the sampled two intervals 13 and 70 in Figs. 12c and 12d, respectively. Clearly, both weighted fair and max-min fair in scenario S_1 achieve an identical CDF curve at both intervals 13 and 70. We further observe that reservation algorithms in scenario S_2 at both intervals 13 and 70 achieve higher CDF curves than that in scenario S_1. This result again implies that the number of application providers with bandwidth demand being fully satisfied, is more in scenario S_1. In addition, since the max-min fair achieves higher fairness index than the weighted fair, the max-min fair maintains a higher CDF curve than the weighted fair in scenario S_2 under both interval 13 and interval 70.

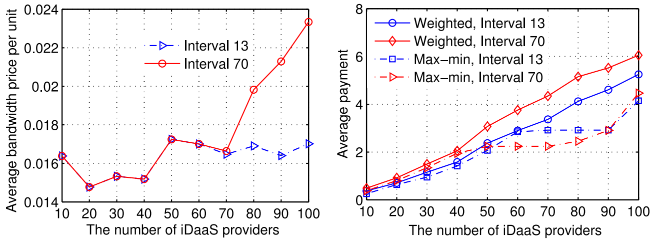
5.2.5 Payment of Application Providers

Another essential performance metric of each application provider is the payment. Fig. 13 reports the payment of application providers in a 96-interval period of time, where Fig. 13a first plots the average payment across all application providers in each time interval. It is clear that



(a) Average payment (b) Standard deviation of payment (c) CDF of payment at interval 13 (d) CDF of payment at interval 70

Fig. 13. Bandwidth reservation of application providers for a 96-interval period of time with (a) average payment, (b) Standard deviation of payment across all application providers, and CDF of application provider's payments at interval 13 (c) and interval 70 (d).



(a) Average bandwidth price per unit (b) Average payment

Fig. 14. The impact of the number of iDaaS providers on the performance of proposed algorithms, with respect to (a) the average bandwidth price per unit and (b) the average payment.

bandwidth reservation algorithms in scenario S_1 always achieve higher payment than that in S_2. The root cause is that application providers in S_1 gets more bandwidth than that in S_2. We can further observe that the max-min fair maintains a lower payment than the weighted fair in both scenario S_1 and S_2. The behind reason is that max-min fair algorithm reserves bandwidth from iDaaS providers in the increasing order of the relative bandwidth price per unit.

To understand the difference of payments among application provider, we plot the standard deviation of application providers' payments in Fig. 13b. Clearly, the standard deviation of payment in scenario S_2 is always lower than that in scenario S_1. Additionally, the max-min fair reservation algorithm achieves a higher standard deviation of payment in S_1, while it is opposite in scenario S_2. The root cause is that an equal ratio of the amount of reserved bandwidth to the demand is maintained by the weighted fair in S_2, while application providers differ in the bandwidth demand.

Finally, we plot the CDF of the payment under two sampled intervals 13 and 70 to study the distribution of application provider's payments, as shown in Figs. 13c and 13d, respectively. Clearly, both the max-min fair and weighted fair in scenario S_2 achieve higher CDF curve than that in S_1, irrespective of interval 13 or 70. In addition, the weighted fair achieves a little higher CDF curve than the max-min fair in S_2. The max-min fair maintains a little higher CDF curve of payment than the weighted fair in scenario S_1 under both interval 13 and 70. These results are mainly caused by that max-min fair starts with reserving iDaaS provider's bandwidth in the increasing order of the bandwidth price per unit, which is different to the weighted fair.

5.3 The Impact of the Number of iDaaS Providers

In order to study the performance of proposed algorithms in small-scale settings, we let the bandwidth demand of application providers remain the same, and change the number of iDaaS providers. More concretely, we use the demand of application providers on two sampled time intervals (13 and 70). Similarly, the bandwidth capacity of each iDaaS provider is set to be uniformly random within range $[10^7, 2\times 10^7]$. Fig. 14a first shows the average bandwidth price per unit at two sampled time intervals 13 and 70, with varying the number of iDaaS providers. We observe that the average bandwidth price per unit approximately increases as the number of iDaaS providers. This is because that given the same demand issued by application providers, the more the number of iDaaS providers, the lower the workload of each iDaaS provider. We further observe that the average bandwidth price per unit in interval 13 is lower than that in interval 13 when the number of iDaaS providers increases to 60. The root cause is that the total demand at time interval 13 is higher than that at interval 70. Fig. 14b further plots the average payment of application providers for both weighted fair and max-min fair algorithms at two sampled time intervals (13 and 70), with varying the number of iDaaS providers. Clearly, the average payment increases as the number of iDaaS providers. Moreover, the max-min fair algorithm always maintains a lower average payment than the weighted fair algorithm.

6 Related Work

In this section, we will present related work in cloud bandwidth reservation and bandwidth price, as these are most closely related to our work in this paper. Recently, there are many researches on reserving cloud bandwidth, which makes offering bandwidth guarantees to application provider's wide area traffic become technically feasible. To provide bandwidth guarantee for such traffic, both intra-datacenter bandwidth and the WAN bandwidth should be reserved. Actually, many proposals have been proposed on datacenter engineering to offer bandwidth guarantees for VM-pairs [27], [28], [29], [30], flows [31], or applications [32]. Moreover, advances on inter-datacenter network also make the bandwidth reservation feasible, i.e., Google B4 [4], which can open a tunnel with guaranteed WAN bandwidth for each flow. These proposals have made private WANs more attractive to application providers that have large amount of wide area traffic.

Currently, wide area traffic is mainly priced based on a usage-based pricing policy [7], which is however unable to price the bandwidth guarantees. Niu etal. focus on pricing cloud bandwidth reservations such that the social welfare is maximized, even with the presence of demand uncertainty [33]. Their another work [34] further proposes a theory of pricing cloud bandwidth for video-on-demand providers who move their video streaming services to cloud. Our work in this paper differs markedly from these works above since they mainly concentrate on the intra-datacenter bandwidth. Our focus is to price the bandwidth guarantees in the new type of service—inter-datacenter network as a service, where application providers make wide area transit bandwidth reservations from iDaaS providers to support their wide area traffic.

7 Conclusion

Motivated by the enormous and fast growing wide area traffic in large-scale Internet applications, we propose a new type of service—inter-datacenter network as a service, for Internet giants like Google and Facebook. Such service providers host large-scale private WANs between their geographically distributed datacenters. We demonstrate the feasibility and reveal the potential benefits of such kind of new service. Specifically, we design a bandwidth trading market for multiple iDaaS providers and application providers. Here, application providers reserve guaranteed from iDaaS providers to support their wide area traffic. We focus on the essential bandwidth pricing problem. Furthermore, we introduce a two-stage Stackelberg game to model the interaction between iDaaS providers and application providers. Those iDaaS providers play a Nash bargaining game while application providers play a non-cooperative game. We prove the existence and uniqueness of the Nash equilibrium for the application provider's game. We further make efforts to compute the bandwidth price by blending the advantage of a geometric Nash bargaining solution and demand segmentation method. Based on the pre-computed price, we propose two bandwidth reservation algorithms. Finally, we conduct trace-driven experiments to evaluate the proposed algorithms. The evaluation results demonstrate that our algorithms are capable of benefiting both iDaaS providers and application providers.

Acknowledgments

This work is supported by the National Key Research and Development Program of China No. 2016YFB1000205, the State Key Program of National Natural Science of China (Grant No. 61432002), the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61225010), NSFC Grant Nos. 61772112, 61672379, 61272417 and 61370199; Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20130041110019), and the Fundamental Research Funds for the Central Universities (Grant. DUT15QY20); the National Natural Science Foundation for Outstanding Excellent young scholars of China under Grant No.61422214, and National Basic Research Program (973 program) under Grant No.2014CB347800. We would also like to thank Yingwei Jin for providing experimental environment for this paper.

References

[1] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-CDN movie delivery," in *Proc. IEEE INFOCOM*, 2012, pp. 1620–1628.
[2] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu, "A first look at Inter-data center traffic characteristics via yahoo! datasets," in *Proc. IEEE INFOCOM*, Shanghai, China, 2011, pp. 1620–1628.
[3] (2011). Forrester research [Online]. Available: http://info.infineta.com/1/5622/2011-01-27/Y26
[4] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu et al., "B4: Experience with a globally-deployed software defined wan," in *Proc. ACM SIGCOMM Conf.*, Hong Kong, 2013, pp. 3–14.
[5] J. Roberts, "The cloud is the future Internet: How can we engineer a cloud," in *Proc. IEEE INFOCOM Keynote*. [Online]. Available: http://infocom.di.unimi.it/images/stories/infocom/keynote-jroberts.pdf, 2013.
[6] P. Gill, M. F. Arlitt, Z. Li, and A. Mahanti, "The flattening internet topology: Natural evolution, unsightly barnacles or contrived collapse?" in *Proc. 9th Int. Conf. Passive Active Netw. Meas.*, Cleveland, OH, USA, 2008, pp. 1–10.
[7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica et al., "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.
[8] E. Altman, T. Basar, T. Jiménez, and N. Shimkin, "Competitive routing in networks with polynomial costs," *IEEE Trans. Autom. Control*, vol. 47, no. 1, pp. 92–96, Jan. 2002.
[9] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA, USA: MIT Press, 1991.
[10] Y. Feng, B. Li, and B. Li, "Bargaining towards maximized resource utilization in video streaming datacenters," in *Proc. IEEE INFOCOM*, 2012, pp. 1134–1142.
[11] (2007). Yahoo! research webscope program [Online]. Available: http://labs.yahoo.com/organization/academic-relations
[12] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall, "Improving the reliability of internet paths with One-hop source routing," in *Proc. USENIX 6th Conf. Symp. Operating Syst. Des. Implementation*, San Francisco, CA, USA, 2004.
[13] S. Sundaresan, W. d. Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, "Broadband internet performance: A view from the gateway," in *Proc. ACM SIGCOMM Conf.*, Toronto, ON, Canada, 2011, pp. 134–145.
[14] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, Melbourne, Vic., Australia, 2010, pp. 267–280.
[15] Z. Zhou, F. Liu, Y. Xu, R. Zou, H. Xu, J. C. S. Lui, and H. Jin, "Carbon-aware load balancing for geo-distributed cloud services," in *Proc. IEEE 21st Int. Symp. Model., Anal., Simul. Comput. Telecommun. Syst.*, San Francisco, CA, USA, 2013, pp. 232–241.
[16] Z. Zhou, F. Liu, Z. Li, and H. Jin, "When smart grid meets geo-distributed cloud: An auction approach to datacenter demand response," in *Proc. IEEE INFOCOM*, Kowloon, Hong Kong, 2015, pp. 2650–2658.
[17] (2015). Zimory cloud computing [Online]. Available: http://www.zimory.com/
[18] V. Valancius, C. Lumezanu, N. Feamster, R. Johari, and V. V. Vazirani, "How many tiers? pricing in the internet transit market," in *Proc. ACM SIGCOMM Conf.*, Toronto, ON, Canada, 2011, pp. 194–205.
[19] W. B. Norton. (2010). Drpeering.net [Online]. Available: http://drpeering.net
[20] S. Shakkottai and R. Srikant,, "Economics of network pricing with multiple ISPs," in *Proc. IEEE INFOCOM*, Miami, FL, USA, 2005, pp. 184–194.
[21] (2012). Google fiber [Online]. Available: https://fiber.google.com/
[22] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. SIAM Series in Classics in Applied Mathematics. Philadelphia, PA USA: SIAM, 1999.
[23] J. Rosen, "Existence and uniqueness of equilibrium points for concave N-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.
[24] J. Guo, F. Liu, D. Zeng, J. Lui, and H. Jin, "A cooperative game based allocation for sharing data center networks," in *Proc. IEEE INFOCOM*, Turin, Italy, 2013, pp. 2139–2147.
[25] L. Chiaraviglio, M. Mellia, and N. Fabio, "Minimizing ISP network energy cost: Formulation and solutions," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 463–476, Arp. 2012.
[26] R. Jain, D.-M. Chiu, and W. R. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System*. Hudson, MA, USA: Eastern Res. Laboratory, Digital Equipment Corporation, 1984.
[27] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, and I. Stoica, "FairCloud: Sharing the network in cloud computing," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Helsinki, Finland, 2012, pp. 187–198.
[28] 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. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 1572–1580.
[29] J. Guo, F. Liu, H. Tang, Y. Lian, H. Jin, and J. C. S. Lui, "Falloc: Fair network bandwidth allocation in IaaS datacenters via a bargaining game approach," in *Proc. IEEE Int. Conf. Netw. Protocol*, Göttingen, Germany, 2013, pp. 1–10.
[30] J. Guo, F. Liu, J. C. S. Lui, and H. Jin, "Fair network bandwidth allocation in IaaS datacenters via a bargaining game approach," *IEEE Trans. Netw.*, 2015, Doi: 10.1109/TNET.2015.2389270, in press.
[31] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, "Sharing the data center network," in *Proc. USENIX Conf. Netw. Syst. Des. Implementation*, Boston, America, 2011, pp. 309–322.
[32] L. Chen, Y. Feng, B. Li, and B. Li, "Towards performance-centric fairness in datacenter networks," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 1599–1607.
[33] D. Niu, C. Feng, and B. Li, "Pricing cloud bandwidth reservations under demand uncertainty," in *Proc. 12th ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. Meas. Model. Comput. Syst.*, London, England, 2012, pp. 151–162.
[34] D. Niu, C. Feng, and B. Li, "A theory of cloud bandwidth pricing for video-on-demand providers," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 711–719.
[35] X. Dong, Z. Guo, X. Zhou, H. Qi, and K. Li, "AJSR: An efficient multiple jumps forwarding scheme in software-defined WAN," *IEEE Access*, vol. 5, pp. 3139–3148, 2017.

Wenxin Li received the BE degree from the School of Computer Science and Technology, Dalian University of Technology, China, in 2012. He is currently working toward the PhD degree in the School of Computer Science and Technology, Dalian University of Technology, China. His research interests include datacenter networks and cloud computing.

Deke Guo received the BS degree in industry engineering from Beijing University of Aeronautic and Astronautic, Beijing, China, in 2001, and the PhD degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 2008. He is an associate professor with the College of Information System and Management, National University of Defense Technology, Changsha, China. His research interests include distributed systems, software-defined networking, data center networking, wireless and mobile systems, and interconnection networks.

Keqiu Li received the bachelor's and master's degrees from the Department of Applied Mathematics, Dalian University of Technology in 1994 and 1997, respectively. He received the PhD degree from the Graduate School of Information Science, Japan Advanced Institute of Science and Technology in 2005. He also has two-year postdoctoral experience in the University of Tokyo, Japan. He is currently a professor in the School of Computer Science and Technology, Dalian University of Technology, China. He has published more than 100 technical papers, such as *IEEE TPDS*, *ACM TOIT*, and *ACM TOMCCAP*. His research interests include internet technology, data center networks, cloud computing, and wireless networks. He is an associate editor of *IEEE TPDS* and *IEEE TC*. He is a senior member of the IEEE.

Heng Qi received the bachelor's degree from Hunan University in 2004 and the master's degree from Dalian University of Technology in 2006. He was a lecture at the School of Computer Science and Technology, Dalian University of Technology, China. He received the doctorate degree from Dalian University of Technology in 2012. He served as a software engineer in GlobalLogic-3CIS from 2006 to 2008. His research interests include computer network, multimedia computing, and mobile cloud computing. He has published more than 20 technical papers in international journals and conferences, including *ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMCCAP)* and *Pattern Recognition (PR)*.

Jianhui Zhang received the BE degree from the School of Computer Science and Technology, Harbin Engineering University, Harbin, China, in 2009. He is currently working toward the PhD degree in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests include datacenter networks, network protocols, and cloud computing.