

iDaaS: Inter-Datacenter Network as a Service

Wenxin Li, Deke Guo, Keqiu Li, Heng Qi, and Jianhui Zhang

Abstract—Current-generation Internet-scale applications, such as video streaming, incur a large amount of wide area traffic. Transmitted on the unreliable Internet with no bandwidth guarantees, such traffic is suffering unpredictable network performance, which is however unappealing to the application providers. Fortunately, Internet giants like Google and Microsoft are increasingly deploying their private wide area networks (WANs) to connect their datacenters on the globe. Such private WANs are highly 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 application providers, make bandwidth reservations for bandwidth guarantees from those Internet giants to support their wide area traffic. We study a bandwidth trading market of multiple iDaaS providers and application providers, and concentrate on the essential *bandwidth pricing* problem. The challenging issue involves in pricing in such a market is that the benefits of both iDaaS providers and application providers are close-knit connected. To address this issue, we model the interaction between iDaaS providers and application providers as a Stackelberg game, and analyze the existence and uniqueness of the equilibrium. We further present an efficient bandwidth price computation algorithm by blending the advantage of a geometrical Nash bargaining solution and the demand segmentation method. Based on the computed price, we present two bandwidth reservation algorithms, where each iDaaS provider's bandwidth are reserved in a weighted fair manner and a max-min fair manner, respectively. Finally, we conduct comprehensive trace-driven experiments. The evaluation results show the efficiency of our proposed algorithms in the bandwidth market.

1 INTRODUCTION

Large-scale Internet applications, such as video streaming and social network, 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 exist between different business regions due to the routine background computation and periodic data back up tasks. As revealed in [2], the wide area traffic accounts for up to 45% of the total traffic of a typical business provider. 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].

Traditionally, most application providers acquire bandwidth from Internet Service Providers (ISPs) for their wide area traffic. While this will result in unpredictable and unreliable network performance for the application providers since the network bandwidth is shared 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, Google B4, a private WAN connecting Google's datacenters across the planet, is high 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].

In light of these reasons, we design a new type of service, inter-datacenter network as a service (iDaaS), for companies

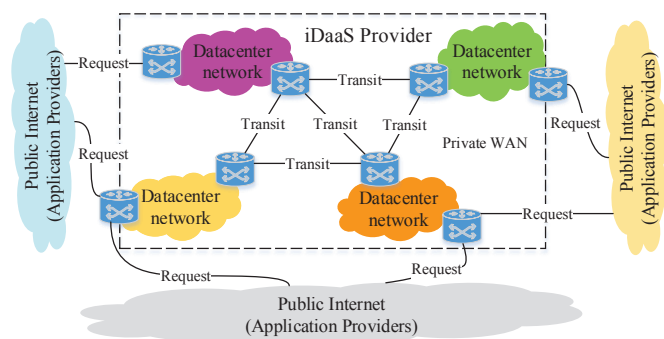


Fig. 1. An illustrative example of the new service iDaaS, where there are multiple application providers and a single iDaaS provider that hosts a private WAN connecting a large number of datacenters on the globe.

like Google, Microsoft that host such private WANs [6]. In this service, 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 requested bandwidth to it, such that the application provider's wide area traffic can be transmitted on the inter-datacenter links with bandwidth guaranteed.

When application providers are on board to use this new type of services, we believe that there will be a bandwidth market between application providers and iDaaS providers. The commodities to be traded in such a market consist of bandwidth reservation, so that the application performance can be guaranteed. Consequently, an emerging *bandwidth pricing* problem dominates the utility of such a market and hence should have priority to be tackled. However, current

- W. Li, K. Li, H. Qi, and Jianhui Zhang are with the School of Computer Science and Technology, Dalian University of Technology, No 2, Linggong Road, Dalian 116023, China. E-mail: {liwenxin@mail.dlut.edu.cn, {keqiu, hengqi}@dlut.edu.cn, zhangjh@mail.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.

pricing method, i.e., the simple pay-as-you-go model based on the number of bytes transferred [7], is insufficient as a model to price bandwidth guarantees.

Pricing in such a bandwidth trading market is a significantly challenge task. On the one hand, each iDaaS provider normally sets its per unit bandwidth price based on the total reserved bandwidth (workload) [8]. As iDaaS providers compete with each other for bandwidth demand with the aim of maximizing the revenue, the per unit bandwidth price of each iDaaS provider can be dynamically changed in the market. 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. While 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 providers and users.

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

In order to compute the bandwidth price and bandwidth reservation in an efficient and practical way, we propose a bandwidth price computation algorithm by blending the advantage of the geometrical Nash bargaining solution [10] and the demand segmentation method. We particularly present two bandwidth reservation algorithms, which reserve each iDaaS provider's bandwidth in a weighted fair manner and max-min fair manner, respectively. Finally, we use comprehensive trace-driven simulations to demonstrate the efficiency of our algorithms in the market of multiple iDaaS providers and multiple 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 hosts large private wide area network to connect their geographically distributed datacenters on the globe. 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 providers and users as a two-stage Stackelberg game. It contains a cooperative game among iDaaS providers and a noncooperative game between application providers. We perform a theoretical analysis with respect to the Nash equilibrium of the noncooperative game.
- We design an efficient and practical bandwidth price

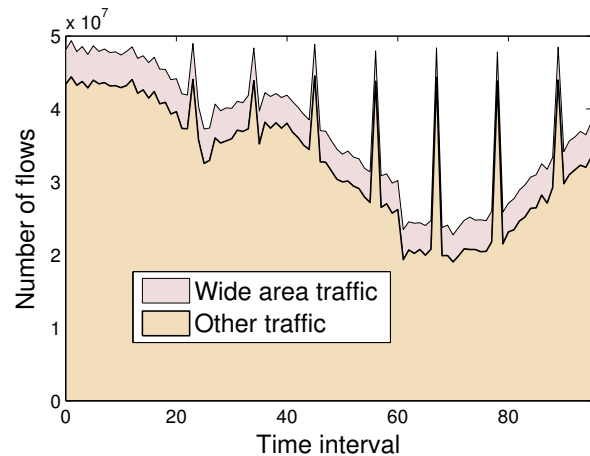


Fig. 2. The number of flows for the extracted wide area traffic from 7 most frequently used ports in traffic between Yahoo! datacenters.

computation 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 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 users. In Section 4, we present our implementation algorithms. In Section 5, we present the performance evaluation. Related work are presented in Section 6. Finally, the conclusions are discussed in Section 7.

2 BACKGROUND AND SYSTEM MODEL

2.1 Background

Current Internet suffers from reliability and network performance issues [11], [12]. In spite of 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 for them. Benson et.al recently reported that such wide area traffic accounts for 40-90% of the total traffic in a typical business [13]. 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 a roundly understanding of the wide area traffic, we detailed analyze some network flow datasets provided by Yahoo! [14]. The datasets not only contain traffic between Yahoo! servers and client, but also contain traffic between different Yahoo! datacenters. Each record in the NetFlow data includes 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! users, we extracted the wide area traffic based on the study of [2]. This study reports that the wide area traffic accounts for up to 45% of Yahoo! total traffic. In this paper, we only extracted the wide area traffic passing through ports 80, 25, 1971, 14011, 5017, 14020, and 14030, which are most frequently used ports in traffic between Yahoo! datacenters as revealed in [2]. Fig. 2 shows the number of flows for the extracted wide area traffic in a 96-period of time (each interval is 15 minutes). We can easily check that the extracted wide area traffic accounts for up to 10% 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 network deployed by some Internet giants among their geographically distributed datacenters, i.e., Google B4 [4], are highly reliable and can provide guaranteed network performance. This implies that application providers can now rent bandwidth from those Internet giants for their large amount of wide area traffic. In this way, datacenters only need to access these traffic, push them to the private wide area network, and finally forward them to the destinations. Each datacenter is exactly similar to a router. In fact, J.Roberts reports that a datacenter can actually do routing in the Internet with the abundant resources (computation, storage, and network) [5]. Following these guidelines, we 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 these efforts may force many companies to provide such service. In the following, we focus on a model of multiple iDaaS providers and multiple application providers.

2.2 System Model

In this section, we describe the system model and the main system parameters. We consider a market of multiple iDaaS providers and multiple application providers in Figure 3. Each iDaaS provider hosts a large amount of WAN transit bandwidth, and sells them to the application providers at a certain price. Each application provider sends its bandwidth demand, and buys it from multiple iDaaS providers. The controller in this market, is mainly used to collect the global information and to make optimal decisions for both iDaaS providers and application providers, and can be deployed by some agents or brokers, i.e., Zimory, an emerging intermediators connecting buyers and sellers of resources [15].

2.2.1 iDaaS Providers and Application Providers

We consider that there are N iDaaS providers. The set of providers is denoted by $\mathcal{N}=\{1, 2, \dots, N\}$. Each iDaaS provider i has v_i amount of bandwidth, which is available for reservation. Each iDaaS provider sets a per unit bandwidth price that they are willing to charge from application providers, while such per unit bandwidth price is normally a function of the total reserved bandwidth on the iDaaS provider [8]. To better indicate this, we let $P_i(x_i)$ denote

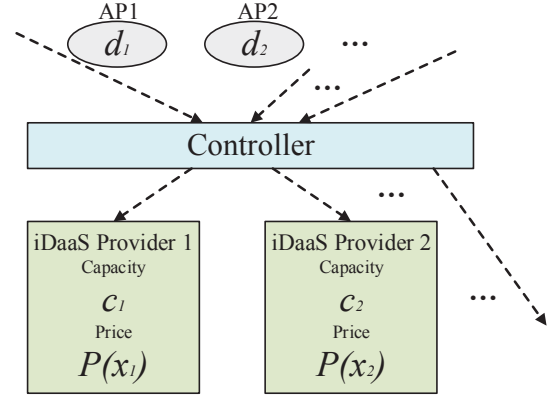


Fig. 3. A system of multiple iDaaS providers and multiple application providers.

the per unit bandwidth price of iDaaS provider i , where x_i represents the total reserved bandwidth on the i -th provider.

Similarly, we consider that there are M application providers. The set of application providers is denoted by $\mathcal{M}=\{1, 2, \dots, M\}$. Each application provider has a bandwidth demand d_j , and hopes to get the guaranteed bandwidth from 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 $\mathbf{x}_j = \{x_{1,j}, \dots, x_{N,j}\}$. The set of bandwidth reservation strategies $\mathbf{x} = \{\mathbf{x}_j : 1, \dots, M\}$ forms a bandwidth reservation policy.

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 $P_i(\cdot)$. It is important to keep in mind that such bandwidth pricing strategy must be close to real-life situations: the more users buy some goods, the lower the unit price.

Considering such guideline, our intuition is to use a smooth but fast-decreasing function to measure the per unit bandwidth price. Such price function should have gradual change rate both at the beginning (highest price) and around stable value (lowest price). This is exactly a Logistic-like pricing function that is widely used in the Internet peering pricing [16]. Applying the Logistic-like pricing, $P_i(x_i)$ can be modeled as follow:

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

where $L_i > 0, A_i > 0, 0 < B_i < 1, C_i > 0$ are parameters specified by iDaaS provider i .

Key mechanisms and benefits of the Logistic-like pricing is shown in Fig. 4. We can find that such a Logistic-like pricing $p_i(x_i)$ can be controlled to slowly decrease at the beginning, fast decrease in the middle, and finally infinitely

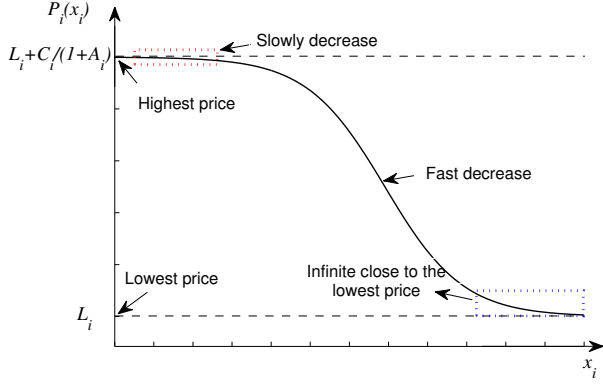


Fig. 4. Key mechanisms and benefits of the Logistic-like pricing.

close to a stable value. Clearly, the price 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.

Different iDaaS provider may have different parameters configuration, such that the price can be different from each other. For easy presentation, 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 $x_i \in [0, c_i]$, which is defined in Eq. (1). The collection of bandwidth pricing strategies $\{P_i(\cdot) : i=1, \dots, N\}$ forms a *bandwidth pricing policy*.

3 PROVIDERS AND USERS INTERACTION: A STACKELBERG GAME APPROACH

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

Based on the model described in Section 2, we consider that both iDaaS providers and application providers are rational and selfish. As iDaaS providers have the right to decide the bandwidth price that they are willing to charge from application providers, so as to maximize its own utility in terms of its own revenue. Application provider, which is competitive with other application provider, decides only the payment it is willing to make, under the announced price 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], where, 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

Now we give a concrete expression of utility functions, based on which, the existence and uniqueness of the Nash Equilibrium is analyzed.

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

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

Since application provider competes with each other, the change in one application provider's utility is likely to cause the changes in other application providers' utilities. Based on above guideline, we let the utility of application provider j be denoted by $U_j(x_{i,j}, x_i)$, which is a function of two arguments, namely $x_{i,j}$ and x_i . 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*. To put these formally, we define the utility of application provider j to be the negative value of the sum of its payment to each iDaaS provider.

$$U_j(x_{i,j}, x_i) = - \sum_{i \in \mathcal{N}} P_i(x_i) x_{i,j}. \quad (3)$$

Since each application provider $j \in \mathcal{M}$ has a strong desire of getting the guaranteed bandwidth, which means that its bandwidth demand should be fully satisfied. Hence, each application provider involves in solving the following utility maximization problem,

$$\max_{x_{i,j}} U_j(x_{i,j}, x_i) \quad s.t. \quad \sum_{i \in \mathcal{N}} x_{i,j} = d_j. \quad (4)$$

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

3.2 Maximizing iDaaS provider's utility

The leader of the Stackelberg game, iDaaS provider, optimizes its bandwidth pricing strategy in order to maximize its revenue 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, N iDaaS providers can be viewed as players who are 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 product of utility gains of all players are maximized.

$$\begin{aligned} \max_{x_i} \quad & \prod_{i \in \mathcal{N}} Q_i(x_i) \\ s.t. \quad & x_i \leq v_i, \forall i \in \mathcal{N}, \\ & \sum_{i \in \mathcal{N}} x_i = D, \end{aligned} \quad (5)$$

where the first constraint means that the total reserved bandwidth on each iDaaS provider cannot exceed its bandwidth capacity, and the second constraint means that the total reserved bandwidth should equal to the total demand of application providers. Once Eq. (5) is solved, then 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 bandwidth price and the amount of bandwidth that iDaaS providers are willing to share, each application provider makes its bandwidth reservation strategy according to its own utility maximization in Eq. (4). Clearly, one application provider's utility is likely to affect other one's, and since application providers are selfish, we are actually faced with a noncooperative game [17]. Thus, we are interested in the Nash 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 x^* is the Nash equilibrium (NE) if, for all $j \in \mathcal{M}$, the following conditions holds:

$$U_j(x_{i,j}^*, x_i^*) = \max_{x_{i,j}} U_j(x_{i,j}, x_{i-j} + x_{i,j}), \quad (6)$$

where x_{i-j} is the other application providers' bandwidth reservation strategies except j .

In general, a Nash equilibrium exists in a competitive noncooperative game when each player has a concave function as the utility. For completeness, we analyze the NE existence and uniqueness of 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

$$\frac{1}{B_i} \left(2 + \frac{4}{A_i e^{B_i x_i} - 1} \right) < x_{i,j}. \quad (7)$$

Proof: Please refer to the Appendix A. □

In the following, we present algorithms to compute the optimal bandwidth pricing policy and the bandwidth reservation policy, such that both iDaaS providers and application providers are enjoying the maximum utility.

4 IMPLEMENTATION ALGORITHM

In this section, we propose an efficient price computation algorithm by blending the advantage of the geometrical representation of Nash bargaining games [10] and the demand segmentation method. Then, we present two bandwidth reservation algorithms based on the pre-computed bandwidth price and the amount of bandwidth that iDaaS providers are willing to share.

4.1 Price computation

As described in the above Section, given the total bandwidth demand of application providers, iDaaS providers are actually playing a Nash Bargaining game where players cooperate with each other to achieve a win-win solution, such that each player gains the maximum utility. Note that the problem in Eq. (5) is usually approximatively solved by numerical methods like gradient projection method [18], which requires numerous iterations. Since our objective is

an efficient price computation algorithm that can be implemented in real-world controller hosted by agents or brokers and executed in a lightweight fashion, we propose an efficient bandwidth price computation algorithm based on the geometrical representation of Nash bargaining games, in which 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

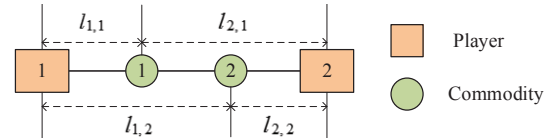


Fig. 5. An illustrative example of 2-player geometrical game.

A geometrical Nash bargaining game can be described in some low-dimensional space in the region of Euclidean Geometry. For example, Fig. 5 shows a 2-dimensional spatial game, which represents a 2-player bargaining game. Utilities of the two players for commodity 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.,

$$l_{i,k} = \frac{1/R_{i,k}}{\sum_i (1/R_{i,k})}, \forall i, \forall k \quad (8)$$

where $R_{i,k}$ is 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. 5, 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

$$\phi_{i,k} = l_{i,k} \cdot R_{i,k}. \quad (9)$$

Fig. 6 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 an equilibrium in the geometrical game, the sum of utility distance product should be equally partitioned among all players. In Fig. 6, player 1 and 2 are lying at the two end points of the lever, where the utility distance products of commodities are regarded as force moments.

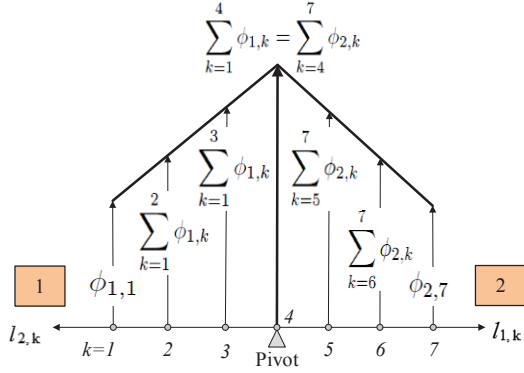


Fig. 6. Finding the pivot point in a 2-player game, which is similar to a lever system.

We can easily find that the pivot point of the lever in this example should be lying in 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 2-dimensional geometrical game is determined by balancing the moments between two players:

$$\mu = \frac{1}{2} \sum_{k \in \mathcal{K}} \phi_{i,k}. \quad (10)$$

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. (5) can not be readily 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 demand issued by all the application providers, which means that there is only one commodity. Simply solving Eq. (5) 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 an infinity number. All the sub-commodities constitute a set $\mathcal{K} = \{1, \dots, K\}$, 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. (5) 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 different player has different utility for different sub-commodity, we let player i 's utility for sub-commodity k be defined as follow:

$$Q_i(r_k) = P(r_k)r_k. \quad (11)$$

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

$$l_{i,k} = \frac{1/Q_i(r_k)}{\sum_i (1/Q_i(r_k))}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}. \quad (12)$$

Algorithm 1 Bandwidth price computation

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: **end for**
 - 9: Each iDaaS provider i computes its bandwidth price according to Eq. (1)

We can easily 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

$$\phi_{i,k} = l_{i,k} \cdot Q_i(r_k), \forall i \in \mathcal{N}, \forall k \in \mathcal{K}. \quad (13)$$

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.

$$\mu_i = \frac{1}{N} \sum_{k \in \mathcal{K}} \phi_{i,k}, \forall i \in \mathcal{N} \quad (14)$$

The following theorem 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: Please refer to the Appendix B. \square

Based on the guideline of Theorem 2, we can now design our bandwidth price computation algorithm which are simple and lightweight. The key idea is that the total bandwidth demand are allocated to iDaaS providers based on their capacities and the average demand, and if an iDaaS provider's capacity is less than the average demand, then the unallocated demand should be equally allocated to other iDaaS providers. Therefore, our bandwidth price 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 allocate to it. Once the demand allocation 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 each iDaaS provider's bandwidth price, which is useful for application provider's bandwidth reservation. The bandwidth price computation algorithm based on the geometrical bargaining game is summarized in Algorithm 1.

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: $x_{i,j}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M}$

- 1: **for** each $i \in \mathcal{N}$ **do**
 - 2: **for** each $j \in \mathcal{M}$ **do**
 - 3: $x_{i,j} \leftarrow \frac{d_j}{\sum_j d_j} \cdot x_i$
 - 4: **end for**
 - 5: **end for**
 - 6: **return** $x_{i,j}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M}$
-

4.2 Bandwidth reservation

Given the pre-computed bandwidth price and the amount of bandwidth that each iDaaS providers are willing to share, we now 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. One application provider's utility can significantly affect other one's and each application provider wishes to reserve more bandwidth from iDaaS providers with low per unit bandwidth price, and hence the amount of bandwidth of each iDaaS providers should be allocated in some way. In this paper, we consider two strategies for reserving bandwidth on each iDaaS providers: weighted fairness and max-min fairness, and thus present two algorithms.

The first bandwidth reservation algorithm is based on an idea of weighted fair bandwidth reservation, which is shown in Algorithm 2. The key of this algorithm is that each application provider reserves bandwidth from each iDaaS provider based 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 a max-min fair manner in reserving each iDaaS provider's bandwidth, which is summarized in Algorithm 3, with the computed bandwidth price and amount of bandwidth that each iDaaS provider is willing to share. This algorithm starts with sorting all the iDaaS providers in the increasing order of its bandwidth price (Step 1), and sorting all 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 mean bandwidth of each iDaaS provider can be updated. We can easy check that each iDaaS provider's bandwidth are actually reserved in a max-min fair manner. This means that if an application provider's left bandwidth demand is less than the mean bandwidth on the iDaaS provider, then the unreserved bandwidth will be fairly reserved by other

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: **end if**
 - 10: $x_{i,j} \leftarrow \min\{d_j, \frac{x_i}{\mathcal{M}_{left}}\}$
 - 11: $d_j \leftarrow d_j - x_{i,j}$
 - 12: $x_i \leftarrow x_i - x_{i,j}$
 - 13: **end for**
 - 14: **end for**
 - 15: **return** $x_{i,j}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M}$
-

application providers.

5 PERFORMANCE EVALUATION

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

5.1 Experiments settings

We simulate that there are 100 ($N = 100$) iDaaS providers. Each iDaaS provider hosts an amount of bandwidth, and sells them to the 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. Such unit can be $1kbps$ in a commonly WAN bandwidth setting ($[1, 10]Gbps$)[3], and can even become $10 \sim 100kbps$ when iDaaS provider is further developed as a representative ISP that has $100Gbps$ amount of bandwidth [19]. Note that the total bandwidth of all iDaaS providers can accommodate the total bandwidth demand of application providers in scenario S_1 , while scenario S_2 cannot. Without loss of generality, the minimum per unit bandwidth price 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. These datasets, collected from Yahoo! border routers every 15-minute during one day, contain not only traffic between Yahoo! servers and client, but also

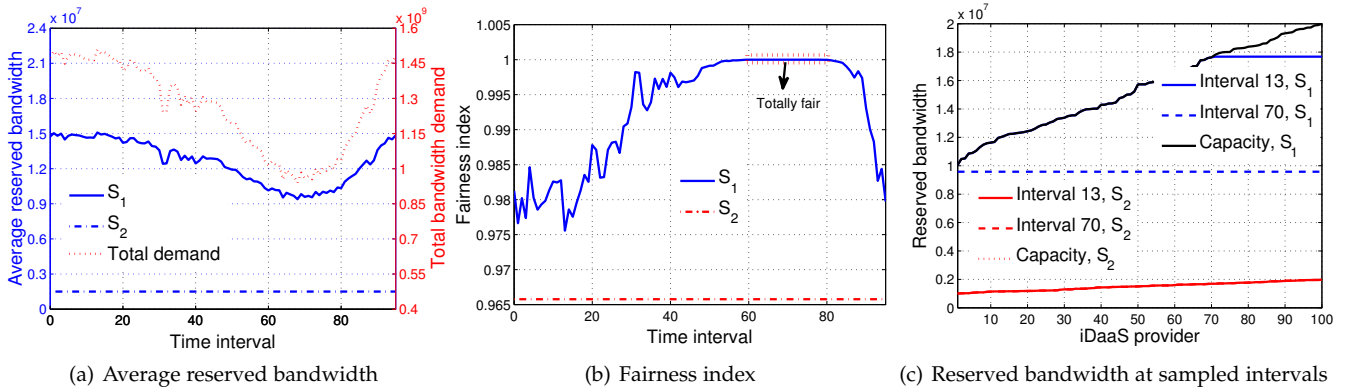


Fig. 7. Reserved bandwidth on iDaaS providers for a 96-interval period of time, in terms of (a) average reserved bandwidth, (b) fairness index associated with the reserved bandwidth x_i , (c) the reserved bandwidth of each iDaaS provider at sampled intervals 13 and 70.

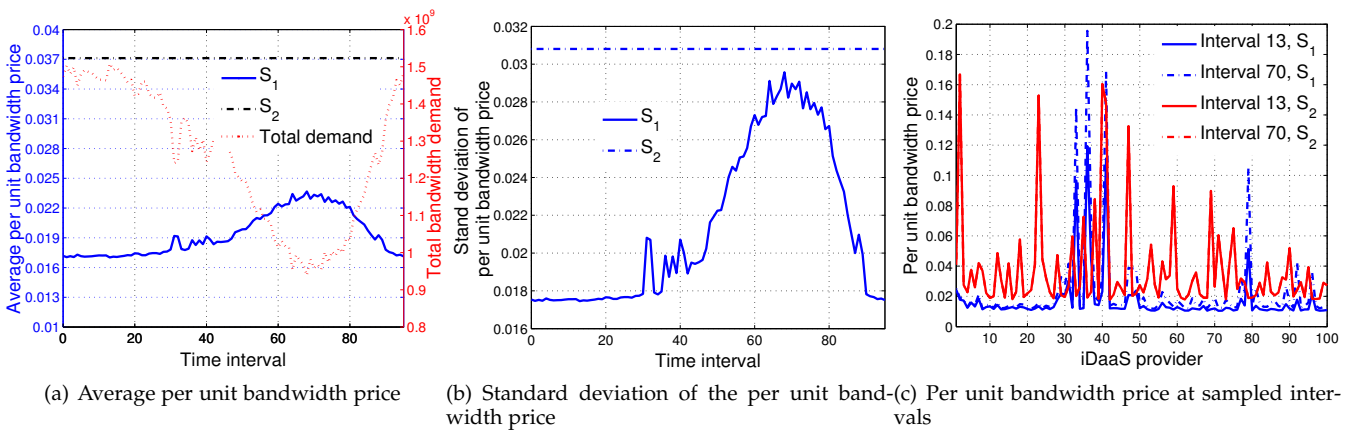


Fig. 8. Per unit bandwidth price of iDaaS providers for a 96-interval period of time, in terms of (a) average per unit bandwidth price, (b) standard deviation of the per unit bandwidth price across all iDaaS providers, (c) the per unit bandwidth price of each iDaaS provider at sampled intervals 13 and 70.

contain traffic between different Yahoo! datacenters. Each record in the NetFlow data includes 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. In our experiments, we extract the traffic between different datacenters from some most frequently used ports like 80, 25, 1971, 14011, 5017, 14020, and 14030, which are described in the above Section 2. The extracted inter-datacenter traffic is used to represent the wide area traffic. Although the wide area traffic we extracted actually are issued by only one application provider (Yahoo!), we believe that they can faithfully reflect traffic demand distribution, and it is appropriate to use them for the purpose of benchmarking the performance of our proposed algorithms.

Therefore, we consider that the extracted wide area traffic is partitioned among multiple application providers. 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. All of our experiments are conducted in 96 15-

minute intervals since the datasets are collected during one day.

5.2 Evaluation results and analysis

For the following experiments, we consider two scenarios (S_1 , S_2). In the two scenarios, 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. 7 first shows the reserved bandwidth on iDaaS providers for a 96-interval period of time. The average reserved bandwidth in both scenario S_1 and S_2 is shown in Fig. 7(a). Clearly, the average reserved bandwidth on iDaaS providers closely follows 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 .

A more 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 [20], which is defined as $F = \frac{(\sum_{i=1}^N x_i)^2}{N \cdot \sum_{i=1}^N x_i^2}$. Fig. 7(b) shows

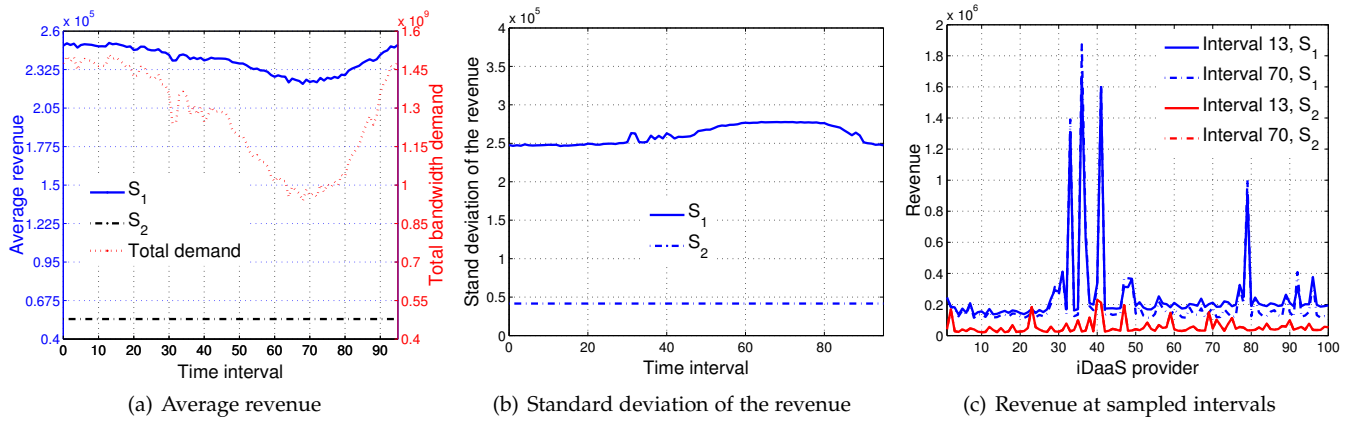


Fig. 9. Revenue of iDaaS providers for a 96-interval period of time, in terms of (a) average revenue, (b) standard deviation of the revenue across all iDaaS providers, (c) the revenue each iDaaS provider at sampled intervals 13 and 70.

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 are totally reserved in each interval for scenario S_2 , while it is not for S_1 .

To understand the reserved bandwidth on a microscopic level, we also plot the reserved bandwidth on each iDaaS provider under two sampled intervals 13 and 70 in Fig. 7(c). Interval 13 corresponds to a high demand time, while interval 70 corresponds to a low demand time. We can easily find that the reserved bandwidth on each iDaaS provider is not exceeding its capacity for each sampled interval. In scenario S_1 , most iDaaS provider's bandwidth are totally reserved except some with high capacity at interval 13, while each iDaaS provider maintains an equal reserved bandwidth at interval 70. Moreover, each iDaaS provider's bandwidth are totally reserved in scenario S_2 for both sampled intervals. These results exactly verify the key idea of our algorithm 1.

5.2.2 Per unit bandwidth price of iDaaS providers

Fig. 8 detailed describes the per unit bandwidth price of iDaaS providers in a 96-interval period of time under both scenario S_1 and S_2 . Fig. 8(a) first plots the average per unit bandwidth price across all iDaaS providers. It is clear that the average per unit bandwidth price in scenario S_1 is lower than that in S_2 , and is completely opposite to the total bandwidth demand. This implies that the more bandwidth demand, the lower the per unit bandwidth price. Note that the average per unit bandwidth price in S_1 is always maintained at a stable value. This is because that each iDaaS provider's bandwidth in scenario S_2 are totally reserved in all intervals.

To understand the difference of iDaaS providers' prices, we also plot the standard deviation of the per unit bandwidth price across all iDaaS providers in Fig. 8(b). We observe that S_2 always maintains a higher standard deviation of the per unit bandwidth price than S_1 . The main reason is that reserved bandwidth on each iDaaS provider in S_2 are exactly its bandwidth capacity, while iDaaS provider's bandwidth capacity is completely different from each other.

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 , it can easily be reasoned about that the standard deviation of per unit bandwidth price in scenario S_1 is completely opposite to total demand.

We also plot per unit bandwidth price for each iDaaS provider for both scenario S_1 and S_2 under two sampled intervals 13 and 70. Due to the limited bandwidth capacity in S_2 , the per unit bandwidth 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 per unit bandwidth price at interval 13 than that at interval 70. This further indicates that the more bandwidth one buys, the lower the per unit bandwidth price.

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. 9. We first show average revenue across all iDaaS providers in Fig. 9(a). Clearly, average revenue in S_1 is always higher than that in S_2 , and closely follows the total bandwidth demand.

Fig. 9(b) plots the standard deviation of revenue across all iDaaS providers. We observe that S_2 always maintain a lower standard deviation of the revenue than S_1 . This may be the case that a lower reserved bandwidth indicated a higher per unit bandwidth price 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 have a roundly understanding of iDaaS providers' revenue, 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 . 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

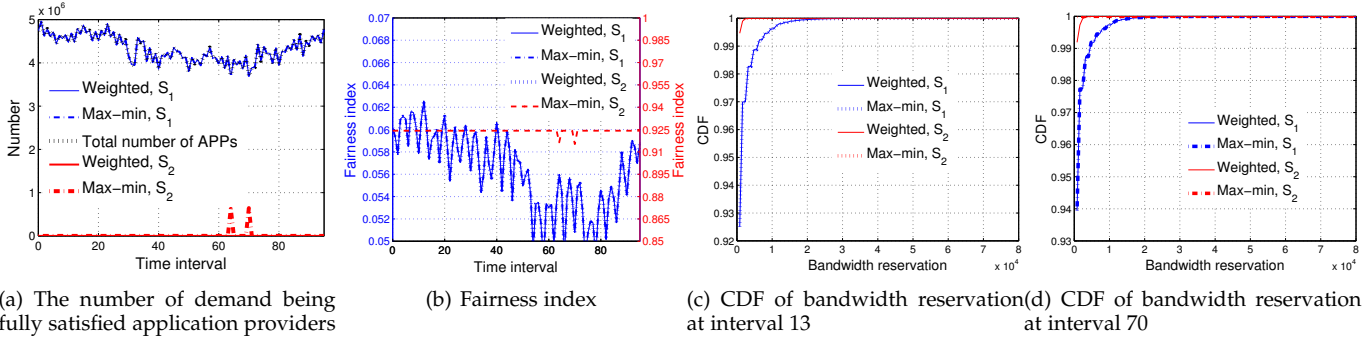


Fig. 10. Bandwidth reservation of application providers for a 96-interval period of time with (a) the number of demand being fully satisfied application providers, (b) Fairness index of bandwidth reservation across all application providers, and CDF of bandwidth reservation for application providers at interval 13 (c) and interval 70 (d).

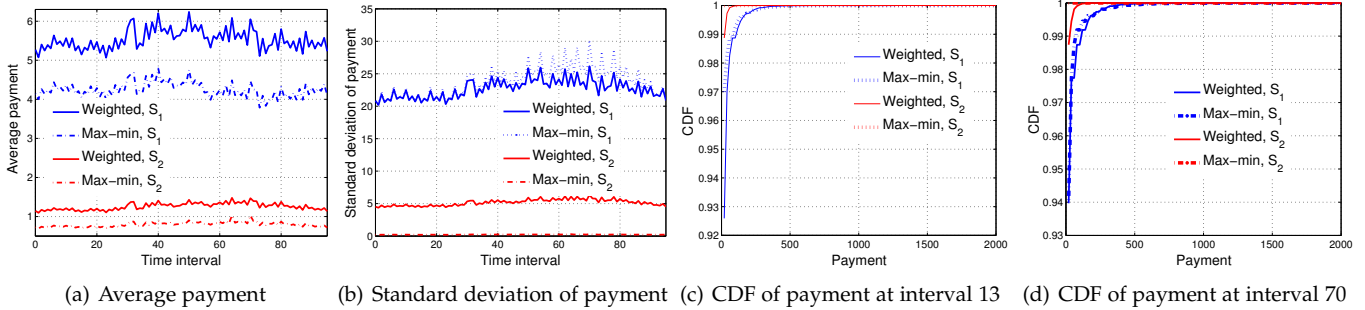


Fig. 11. Bandwidth reservation of application providers for a 96-interval period of time with (a) average payment, (b) Standard deviation of payment across all application providers, and CDF of application provider's payments at interval 13 (c) and interval 70 (d).

demand application providers issue, the higher revenue iDaaS providers gain.

5.2.4 Bandwidth reservation of application providers

We now study the behavior of the application provider in the bandwidth trading market. Specially, we evaluate the performance of both weighted fair and max-min fair bandwidth reservation algorithms. Fig. 10 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. 10(a). We can easily find that both weighted fair and max-min fair reservation algorithms can satisfy all application providers' demand in scenario S_1 , while they cannot in scenario S_2 . This is because that the total bandwidth that iDaaS providers are willing to allocate are 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 weighted fair. This is because that each iDaaS provider's bandwidth are reserved by application providers based on the corresponding demand associated weight, such that the bandwidth reserved by each application provider can be lower than their demand when the sum offered bandwidth is less than the total demand.

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. 10(b). Similar to the fairness index described above, we 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 easily find that both weighted fair and max-min fair in S_1 maintain completely same fairness index. The root cause is that both of them fully satisfy all application providers' demand. We also find that weighted fair in scenario S_2 achieve completely same fairness as the former two. This is because that the bandwidth reserved by each application provider is reduced in a same ratio based on the relative demand when applying the weighted fair bandwidth reservation algorithm under a low bandwidth supply condition. In comparison, the max-min fair in scenario S_2 always achieves higher fairness index than the above three.

We further plot the CDF of bandwidth reservation under the sampled two intervals 13 and 70 in Fig. 10(c) and Fig. 10(d), respectively. Clearly, both weighted fair and max-min fair in scenario S_1 achieve identical CDF curve associated with the bandwidth reservation of per application provider under both interval 13 and 70. We further observe that reservation algorithms in scenario S_2 at both interval 13 and 70 achieve higher CDF curve than that in scenario S_1 . This result again implies that the number of demand being fully satisfied application provider is more in scenario S_1 . In addition, since max-min fair achieves higher fairness index than weighted fair, max-min fair maintains a higher CDF curve than weighted fair in scenario S_2 under both interval 13 and interval 70.

5.2.5 Payment of application providers

A more important performance metric in application provider's behavior is the payment. Fig. 11 detailed de-

scribed the payment of application providers in a 96-interval period of time, where Fig. 11(a) first plots the average payment across all application providers in each time interval. It is clear that bandwidth reservation algorithms in scenario S_1 always achieve higher payment than that in S_2 . This is mainly because that application providers in S_1 gets more bandwidth than that in S_2 . We can further observe that max-min fair maintains a lower payment than weighted fair in both scenario S_1 and S_2 . This can be easily reasoned about by that max-min fair algorithm reserves bandwidth from iDaaS providers in the increasing order of the relative per unit bandwidth price.

To understand the difference between each application provider's payments, we plot the standard deviation of application providers' payments in Fig. 11(b). Clearly, S_2 always maintain a lower standard deviation of payment for both max-min fair and weighted fair reservation algorithms than S_1 . We further observe that max-min fair reservation algorithm achieves higher standard deviation of payment in S_1 , while it is opposite in scenario S_2 . The root cause is that the size of bandwidth that each application provider reserve is scaling down in equal ratio based on the relative demand.

Finally, we plot the CDF of the payment under two sampled intervals 13 and 70 to study the distribution of application provider payment, which are shown in Fig. 11(c) and Fig. 11(d), respectively. Clearly, both max-min and weighted reservation algorithms in scenario S_2 achieve higher CDF curve of the payment than that in S_1 under both interval 13 and 70. In addition, payment of a same portion of application providers for weighted fair reservation algorithm is higher than that for max-min fair reservation algorithm in S_2 since a little higher CDF curve is maintained by weighted fair algorithm. We further observe that max-min fair reservation algorithm maintains a little higher CDF curve of payment than weighted fair reservation algorithm in scenario S_1 under both interval 13 and 70. These results are mainly caused by that max-min fair reservation strategy starts with filling each application provider's demand in the increasing order of the relative per unit bandwidth price, which is different to the weighted fair reservation strategy.

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 provider 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 [21], flows [22], or applications [23]. 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 et.al focus on pricing cloud bandwidth reservations such that the social welfare is maximized, even with the presence of demand uncertainty [24]. Their another work [25] 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 that host large-scale private WANs between their geographically distributed datacenters. We demonstrate the feasibility and reveal the potential benefits of such new service. Specifically, we consider a bandwidth trading market of multiple iDaaS providers and multiple application providers, where application providers make bandwidth reservation for bandwidth guarantees from iDaaS providers to support their wide area traffic. We focus on the essential bandwidth pricing problem. Furthermore, we apply a two-stage Stackelberg game to model the interaction between iDaaS providers and application providers, where both iDaaS providers themselves are playing a Nash bargaining game while application providers plays a noncooperative game. Following comprehensively analysis, 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 real-world traces to evaluate the proposed algorithms. The evaluation results demonstrate that our algorithms are capable of benefiting both iDaaS providers and application providers.

APPENDIX A PROOF OF THEOREM 1

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

$$\begin{aligned} \frac{\partial G_j(x_{i,j}, x_i)}{\partial x_{i,j}} &= -x_{i,j} \frac{\partial^2 P_i(x_i)}{\partial^2 x_{i,j}} - 2 \frac{\partial P_i(x_i)}{\partial x_{i,j}} \\ &= \frac{A_i B_i C_i e^{B_i x_i} (A_i e^{B_i x_i} (2 - B_i x_{i,j}) + 2 + B_i x_{i,j})}{(1 + A_i e^{B_i x_i})^3} \end{aligned} \quad (15)$$

Based on the guidelines of [26], 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

$$\begin{aligned} & A_i e^{B_i x_i} (2 - B_i x_{i,j}) + 2 + B_i x_{i,j} < 0 \\ \implies x_{i,j} & > \frac{2A_i e^{B_i x_i} + 2}{A_i B_i e^{B_i x_i} - B_i} = \frac{1}{B_i} \left(2 + \frac{4}{A_i e^{B_i x_i} - 1} \right) \end{aligned} \quad (16)$$

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. (7).

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

$$G_j(\hat{x}_{i,j}, \hat{x}_i) \begin{cases} = \lambda_j & \text{if } \hat{x}_{i,j} > 0, \\ < \lambda_j & \text{if } \hat{x}_{i,j} = 0, \end{cases} \quad (17)$$

and

$$G_j(x_{i,j}, x_i) \begin{cases} = \hat{\lambda}_j & \text{if } x_{i,j} > 0, \\ < \hat{\lambda}_j & \text{if } x_{i,j} = 0. \end{cases} \quad (18)$$

where λ_j and $\hat{\lambda}_j$ are the corresponding Lagrange multipliers. Now we need to prove that $\mathbf{x} = \hat{\mathbf{x}}$, 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:

$$\{\hat{\lambda}_j \geq \lambda_j, \hat{x}_i \geq x_i\} \implies \hat{x}_{i,j} \leq x_{i,j}, \quad (19)$$

$$\{\hat{\lambda}_j \leq \lambda_j, \hat{x}_i \leq x_i\} \implies \hat{x}_{i,j} \geq x_{i,j}. \quad (20)$$

We only prove Eq. (19), since Eq. (20) is symmetric. Note that Eq. (19) 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 ,

$$\begin{aligned} G_j(\hat{x}_{i,j}, \hat{x}_i) &= \hat{\lambda}_j \geq \lambda_j \\ &\geq G_j(x_{i,j}, x_i) \geq G_j(x_{i,j}, \hat{x}_i). \end{aligned} \quad (21)$$

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. (20) that for each j in \mathcal{M}_1 ,

$$\begin{aligned} \sum_{i \in \mathcal{N}_1} \hat{x}_{i,j} &= d_j - \sum_{i \in \mathcal{N}_2} \hat{x}_{i,j} \\ &\leq d_j - \sum_{i \in \mathcal{N}_2} x_{i,j} = \sum_{i \in \mathcal{N}_1} x_{i,j}. \end{aligned} \quad (22)$$

Note that Eq. (19) 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

$$\begin{aligned} \sum_{i \in \mathcal{N}_1} \hat{x}_i &= \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{N}_1} \hat{x}_{i,j} \\ &\leq \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{N}_1} x_{i,j} = \sum_{i \in \mathcal{N}_1} x_i. \end{aligned} \quad (23)$$

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

$$\hat{x}_i = x_i, \text{ for every } i \in \mathcal{N}. \quad (24)$$

We now proceed to show that $\hat{\lambda}_j = \lambda_j$ for each application provider j . To this end, note that Eq. (19) may be strengthened as follow

$$\{\hat{\lambda}_j > \lambda_j, \hat{x}_i = x_i\} \text{ implies that either } \hat{x}_{i,j} < x_{i,j}, \text{ or } \hat{x}_{i,j} = x_{i,j} = 0. \quad (25)$$

Indeed, if $\hat{x}_{i,j} = 0$, then the implication is trivial. Otherwise, if $\hat{x}_{i,j} > 0$, it follows similarly to Eq. (21) that $G_j(\hat{x}_{i,j}, \hat{x}_i) > G_j(x_{i,j}, \hat{x}_i)$, so that $\hat{x}_{i,j} < x_{i,j}$ as required.

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. (21) implies that

$$\sum_{i \in \mathcal{N}} x_{i,j} > \sum_{i \in \mathcal{N}} \hat{x}_{i,j} = d_j, \quad (26)$$

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 . A symmetric argument may be used to show that $\hat{\lambda}_j < \lambda_j$ can not hold as well. Thus, $\hat{\lambda}_j = \lambda_j$ for every application provider $j \in \mathcal{M}$.

Combined with Eq. (24), this implies that by Eq. (19) and Eq. (20) that $\hat{x}_{i,j} = x_{i,j}$ for each i, j , and the uniqueness of the NE is thus proved.

APPENDIX B PROOF OF THEOREM 2

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 obtained by finding its limit value associated K .

$$\begin{aligned} \mu_i &= \frac{1}{N} \lim_{K \rightarrow \infty} \sum_{k=1}^K \phi_{i,k} \\ &= \frac{1}{N} \lim_{K \rightarrow \infty} \sum_{k=1}^K \frac{1}{\sum_{i=1}^N \frac{1}{r_k (L_i + \frac{C_i}{1 + A_i e^{B_i r_k}})}} \\ &= \frac{1}{N} \lim_{K \rightarrow \infty} \sum_{k=1}^K \frac{1}{\sum_{i=1}^N \frac{1}{\frac{D}{K} (L_i + \frac{C_i}{1 + A_i e^{B_i \frac{D}{K}})}}} \\ &= \frac{1}{N} \lim_{K \rightarrow \infty} \frac{1}{\sum_{i=1}^N \frac{1}{K \frac{D}{K} (L_i + \frac{C_i}{1 + A_i e^{B_i \frac{D}{K}})}}} \\ &= \frac{1}{N} \frac{1}{\sum_{i=1}^N \frac{1}{D(L_i + \frac{C_i}{1 + A_i})}}. \end{aligned} \quad (27)$$

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 $0 <$

$\bar{h}_i < 1$ 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 $\bar{h}_i K$. The cumulative utility-distance product on iDaaS provider i 's side can be calculated as

$$\begin{aligned} \lim_{K \rightarrow \infty} \sum_{k=1}^{\bar{h}_i K} \phi_{i,k} &= \lim_{K \rightarrow \infty} \sum_{k=1}^{\bar{h}_i K} \frac{1}{\sum_{i=1}^N \frac{1}{\frac{D}{K}(L_i + \frac{C_i}{1+A_i e^{B_i \frac{D}{K}})}}} \\ &= \lim_{K \rightarrow \infty} \bar{h}_i K \frac{1}{\sum_{i=1}^N \frac{1}{\frac{D}{K}(L_i + \frac{C_i}{1+A_i e^{B_i \frac{D}{K}})}}} \quad (28) \\ &= \bar{h}_i \frac{1}{\sum_{i=1}^N \frac{1}{D(L_i + \frac{C_i}{1+A_i})}} \end{aligned}$$

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

$$\begin{aligned} \bar{h}_i \frac{1}{\sum_{i=1}^N \frac{1}{D(L_i + \frac{C_i}{1+A_i})}} &= \frac{1}{N} \frac{1}{\sum_{i=1}^N \frac{1}{D(L_i + \frac{C_i}{1+A_i})}} \quad (29) \\ \implies \bar{h}_i &= \frac{1}{N}. \end{aligned}$$

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

$$x_i = \bar{h}_i K r_k = \frac{1}{N} K \frac{D}{K} = \frac{D}{N}. \quad (30)$$

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.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61225010); NSFC under Grant nos. of 61173160, 61173161, and 61173162; Project funded by China Postdoctoral Science Foundation (Grant No. 2013M530916), the Fundamental Research Funds for the Central Universities (Grant No. DUT13ZD101 and DUT13JS04).

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 *Proceedings of 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 *Proceedings of IEEE INFOCOM*, Shanghai, China, 2011.
- [3] "Forrester research," <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 *Proceedings of ACM SIGCOMM*, Hong Kong, 2013.
- [5] J. Roberts, "The cloud is the future Internet: how can we engineer a cloud," *IEEE INFOCOM Keynote*, 2013.
- [6] P. Gill, M. F. Arlitt, Z. Li, and A. Mahanti, "The flattening internet topology: Natural evolution, unsightly barnacles or contrived

- collapse?" in *Proceedings of Springer PAM*, Cleveland, OH, USA, 2008.
- [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," *Communications of the 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 Transaction on Automatic Control*, vol. 47, no. 1, pp. 92–96, 2002.
- [9] D. Fudenberg and J. Tirole, *Game Theory*. The MIT Press, 1991.
- [10] Y. Feng, B. Li, and B. Li, "Bargaining towards maximized resource utilization in video streaming datacenters," in *Proceedings of IEEE INFOCOM*, 2012, pp. 1134–1142.
- [11] 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 *Proceedings of USENIX OSDI*, San Francisco, California, USA, 2004.
- [12] S. Sundaresan, W. d. Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, "Broadband internet performance: a view from the gateway," in *Proceedings of ACM SIGCOMM*, Toronto, ON, Canada, 2011.
- [13] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of ACM IMC*, Melbourne, Australia, 2010.
- [14] "Yahoo! research webscope program," <http://labs.yahoo.com/organization/academic-relations>.
- [15] "Zimory cloud computing," <http://www.zimory.com/>.
- [16] W. B. Norton, "Drpeering.net," <http://drpeering.net>.
- [17] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM Series in Classics in Applied Mathematics, 1999.
- [18] J. Guo, F. Liu, D. Zeng, J. Lui, and H. Jin, "A cooperative game based allocation for sharing data center networks," in *Proceedings of IEEE INFOCOM*, Turin, Italy, 2013.
- [19] L. Chiaraviglio, M. Mellia, and N. Fabio, "Minimizing isp network energy cost: formulation and solutions," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 463–476, Arpil 2012.
- [20] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.
- [21] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, and I. Stoica, "Faircloud: sharing the network in cloud computing," in *Proceedings of ACM SIGCOMM*, Helsinki, Finland, 2012.
- [22] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, "Sharing the data center network," in *Proceedings of USENIX NSDI*, Boston, America, 2011.
- [23] L. Chen, Y. Feng, B. Li, and B. Li, "Towards performance-centric fairness in datacenter networks," in *Proceedings of IEEE INFOCOM*, Toronto, Canada, 2014.
- [24] D. Niu, C. Feng, and B. Li, "Pricing cloud bandwidth reservations under demand uncertainty," in *Proceedings of ACM SIGMETRIC*, London, England, 2012.
- [25] —, "A theory of cloud bandwidth pricing for video-on-demand providers," in *Proceedings of IEEE INFOCOM*, Orlando, Florida, 2012.
- [26] J. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.