A Cooperative Game Based Allocation for Sharing Data Center Networks

Jian Guo[1] Fangming Liu*[1] Dan Zeng[1] John C.S. Lui[2] Hai Jin[1]
[1]Key Laboratory of Services Computing Technology and System, Ministry of Education,
School of Computer Science and Technology, Huazhong University of Science and Technology, China.
[2]The Chinese University of Hong Kong.

Abstract—**In current IaaS datacenters, tenants are suffering unfairness since the network bandwidth is shared in a best-effort manner. To achieve predictable network performance for rented virtual machines (VMs), cloud providers should guarantee minimum bandwidth for VMs or allocate the network bandwidth in a fairness fashion at VM-level. At the same time, the network should be efficiently utilized in order to maximize cloud providers' revenue. In this paper, we model the bandwidth sharing problem as a Nash bargaining game, and propose the allocation principles by defining a tunable base bandwidth for each VM. Specifically, we guarantee bandwidth for those VMs with lower network rates than their base bandwidth, while maintaining fairness among other VMs with higher network rates than their base bandwidth. Based on rigorous cooperative game-theoretic approaches, we design a distributed algorithm to achieve efficient and fair bandwidth allocation corresponding to the Nash bargaining solution (NBS). With simulations under typical scenarios, we show that our strategy can meet the two desirable requirements towards predictable performance for tenants as well as high utilization for providers. And by tuning the base bandwidth, our solution can enable cloud providers to flexibly balance the tradeoff between minimum guarantees and fair sharing of datacenter networks.**

I. Introduction

Cloud computing has reformed the way of running today's business in many industries. By multiplexing large-scale computation, storage and network resources, cloud providers can offer an economical choice for enterprises to create and run their respective jobs independently in data centers. However, unlike the CPU and memory resources, the scarce network bandwidth in current data centers is shared across many tenants in a best-effort manner. The bandwidth between two rented virtual machines (VMs) can be significantly affected by other traffic using the same congested link, leading to unpredictable job completion times for tenants. Since in today's IaaS data centers, such as Amazon EC2 [1], the VMs are charged according to their renting time durations, the tenants will suffer unpredictable costs if they are not aware of their achieved bandwidth. Without any performance guarantees, enterprises may be reluctant to migrate their businesses or services to the cloud, which will in turn harm the provider's revenue.

To share data center networks, we argue that the following two basic requirements should be considered from the viewpoints of both cloud providers and tenants:

First, cloud providers, with their overall system perspective, expect to achieve *high utilization* of network resources throughout the data center. The network resources should be fully allocated across VMs if there exist unsatisfied demands, rather than being statically reserved. If a VM is able to utilize the residual bandwidth left by other idle VMs, the increase in throughput will shorten the completion times of jobs that are bottlenecked by network resources. In this way, more applications or VMs can be deployed with the same infrastructure, which will further increase the providers' revenues.

Second, tenants, in their individual perspective, have a strong desire for *predictable network performance*. Based on the state of the arts, two approaches can be applied to achieve predictable performance: *minimum bandwidth guarantee* and *fair bandwidth allocation*. Bandwidth guarantee provides strong isolation for VMs or tenants, since it ensures a lower bounded bandwidth irrespective of the communication patterns of other applications. For potential network intensive applications like MapReduce, this guarantee policy makes it possible to achieve predictable job completion time by reserving bandwidth for them. Fair bandwidth allocation is to share bandwidth in a fairness fashion at VM-level irrespective of the competition at flow level. This can be achieved by extending the traditional fairness notion to a VM-level, such as proportional share, max-min fairness and etc. By fairly allocating bandwidth among VMs/tenants, cloud providers are able to ensure a certain portion of the shared bandwidth for each VM/tenant. For example, in a proportional shared data center network, each VM/tenant is assigned with a certain weight, and then the network resource is shared in proportion to these weights on congested links.

To provide flexible choices for tenants and cloud providers, we believe that both bandwidth guarantee and fair bandwidth allocation are needed for an IaaS data center. Unfortunately, the current policy, either solely relying on bandwidth guarantee or fair bandwidth share, has its shortage on sharing the data center networks. Specifically, if one chooses to reserve bandwidth for VMs in the data center, the precious network resources may be wasted when the traffic demand decreases

*The Corresponding Author is Fangming Liu (fmliu@hust.edu.cn). The research was supported in part by a grant from The National Natural Science Foundation of China (NSFC) under grant No.61103176 and No.61133006, by a grant from the Research Fund of Young Scholars for the Doctoral Program of Higher Education, Ministry of Education, China, under grant No.20110142120079, by a grant from the Independent Innovation Research Fund of Huazhong University of Science and Technology under grant No. 2011QN051, by the CHUTIAN Scholar Project of Hubei Province.

below the reserved bandwidth, which is against the requirement of high utilization. On the other hand, if one solely resorts to a fair bandwidth allocation policy among VMs, e.g., proportional share, the amount of bandwidth shared by a VM will be determined by the number of VMs using the same physical bandwidth resources. Thus, the minimum expected bandwidth of VMs cannot be guaranteed. These tradeoffs, also illustrated in previous work [2], increase the challenges in sharing data center networks.

In this paper, we make the first attempt in the literature to apply rigorous game-theoretic approaches to model and solve datacenter network sharing problem with constraints on both efficiency and fairness. Specifically, we make two main contributions: First, we develop a cooperative game based framework for allocating bandwidth to VMs, which achieves the Nash bargaining solution (NBS) for sharing data center networks. The bargaining solution guarantees minimum bandwidth for each connection between a VM-pair, while keeping fairness among all the VM-pairs. In particular, a distributed bandwidth allocation algorithm is designed to achieve the NBS solution and high utilization across the entire datacenter networks. The allocation procedure for each connection can be executed in parallel and only requires the local bandwidth information of two involved servers (source and destination).

Second, we assign a base bandwidth to each VM as a tunable design knob. Specifically, our policy guarantees the bandwidth for those VMs with bandwidth demands lower than their base bandwidth, while maintaining fairness across other VMs with bandwidth demands higher than their base bandwidth. Through extensive simulations under typical scenarios, we show that by tuning our proposed design knob, cloud providers can flexibly balance the tradeoff between the two potentially conflicting objectives of minimum bandwidth guarantee and fair bandwidth share.

II. Related Work

Recently, there have emerged a number of proposals to provide predictable network performance for VMs in data centers. Taking the perspective of performance isolation at VM level, such prior works either attempted to guarantee minimum bandwidth or to share bandwidth among VMs or network flows in a proportional way.

First, several existing studies focused on bandwidth guarantees via the idea of reserving specified bandwidth for VMs. Oktopus [3] and SeconNet [4] used static reservations throughout the data center to ensure the bandwidth provisioning of inter-VM networks. The virtual topologies presented in these studies provide strong network isolation at VM level. Since the virtual topology is not against bandwidth allocation for VMs, our design can be combined with such virtual topologies. However, it may sacrifice the high utilization of data center networks, if the statically reserved bandwidth is not fully utilized. Gatekeeper [5] assumed a full bisection bandwidth network, and provided minimum bandwidth guarantee for VMs by shaping the traffic of VMs. Yet, the residual bandwidth is shared in a best-effort manner.

Second, other existing solutions provided network isolation at VM level by sharing the network resources proportionally. NetShare [6] assigned different weights among different servers to allocate their relative bandwidth in a centralized fashion. It provides constant proportionality for tenants throughout the datacenter network. Seawall [7] provided a hypervisor based mechanism to share network proportionally. It divides the bandwidth of each congested link according to associated weights of source VMs. Seawall's congestion control inspires us with a lightweight method to realize our distributed algorithm with varying demands. Faircloud [2] offered a deep understanding of the key requirements and properties for network sharing in data centers. The presented policies lay the foundation of exploring the tradeoff space in sharing data center networks. While such competition based policies can achieve high network utilization, they cannot provide hard guarantees on network performance for cloud tenants.

With these existing efforts, we are at the point to design a flexible strategy for the cloud providers to balance the tradeoff between bandwidth guarantee and fair bandwidth allocation, as both the two desirable features are needed by tenants. In order to achieve such goals, we use a game theory based method to share the data center networks. Different from another work [8] that utilized the bargaining game to maximize multi-dimension resource utilization in video streaming data centers, our paper focuses on designing a datacenter network sharing policy with bandwidth guarantee and fairness for multi-tenant IaaS cloud.

III. Model formulation

We first describe the assumptions and model of data center networks in this section, and then formulate the requirements of our bandwidth allocation policy design.

A. Data Center Network Model

Based on recent works on sharing data center networks [2], [5], we abstract the connections of VMs into a hose model [9] shown in Fig. 1, where the network throughput of each VM can only be blocked by its access link. Since existing works such as multi-path routing (e.g., [10]) and multi-tree topologies (e.g., [11]) have considerably improved bisection bandwidth, our policy assumes a full bisection bandwidth network where the physical bandwidth of servers can be fully utilized without considering the bottleneck links inside the data center. Note that even if we cannot fully utilize the bandwidth of each server, we can adjust our model by setting proper capacity constraints for servers.

We consider an IaaS data center consisting of M servers $\mathcal{M} = \{p_1, p_2, \ldots, p_M\}$, and N VMs $\mathcal{N} = \{v_1, v_2, \ldots, v_N\}$ are located across these servers. The bandwidth demand between VMs is described by a matrix $D_N(t) = [D_{i,j}(t)]_{N\times N}$ (as SecondNet [4]), where $D_{i,j}(t)$ denotes the traffic demand from VM v_i to v_j at time t. We specify a bandwidth allocation strategy by giving a rate matrix $r_N(t) = [r_{i,j}(t)]_{N\times N}$, where $r_{i,j}(t)$ is the bandwidth allocated for VM v_i to v_j at time t.

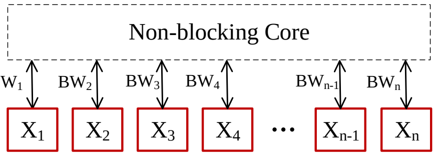

Fig. 1: Data center hose model, where N VMs are connected to a unblocked virtual switch.

In our model, the VM should announce a minimum bandwidth to be guaranteed, which we call it as the *base bandwidth*. The base bandwidth is a borderline value for choosing certain policies when allocating bandwidth. Specifically, if the bandwidth demand of a VM is lower than its base bandwidth, we allocate sufficient bandwidth to satisfy its demand. Otherwise, we set an upper-bounded bandwidth for each VM-pair to maintain fairness among VMs. In addition, the total amount of base bandwidth of VMs on a server should be less than the server's bandwidth capacity, in case of guarantee failure when all VMs have unsatisfied demand.

Accordingly, the VM v_i with its associated base bandwidth B_i can be denoted by a 6-tuple, $v_i : (r_i^I(t), r_i^E(t), D_i^I(t), D_i^E(t), B_i^I, B_i^E)$ with total ingress (egress) bandwidth demand $D_i^I(t)$ $(D_i^E(t))$ and ingress (egress) rate $r_i^I(t)$ $(r_i^E(t))$ at time t. Let $D_j^I(t)$ and $D_j^E(t)$ denote the total ingress and egress demands of VM v_j, respectively. Then, we have $D_j^I(t) = \sum_{i=1}^N D_{i,j}(t)$ and $D_j^E(t) = \sum_{i=1}^N D_{j,i}(t)$. Similarly, the total ingress and egress rates of VM v_j are $r_j^I(t) = \sum_{i=1}^N r_{i,j}(t)$ and $r_j^E(t) = \sum_{i=1}^N r_{j,i}(t)$, respectively.

We use a 3-turple $p_m : (C_m^I, C_m^E, V_{p_m})$ to represent server p_m, where C_m^I and C_m^E are the ingress and egress capacity, respectively, and $V_{p_m} \subset \mathcal{N}$ denotes the set of VMs hosted on server p_m. Without loss of generality, we assume the physical ingress and egress bandwidth of each server are the same, then we have $C_m = C_m^I = C_m^E$.

In this paper, we focus on solving the allocation problem under instantaneous demand at a specific time, hence, the time variable t in the following sections are dropped for simplicity.

B. Bandwidth Allocation Principles

Before presenting the requirements for our allocation policy, we first define two types of VMs in data centers based on whether they have fully utilized the base bandwidth: 1) *Poor* VM whose rate is lower than its base bandwidth, 2) *Rich* VM whose rate has fully occupied or exceeded its base bandwidth.

Due to the aggression of TCP flows, the stable rate of a VM will always reach to the allocated bandwidth if there is unsatisfied demand, but cannot exceed the allocated bandwidth or the bandwidth demand. Hence, given the bandwidth demand D_i and base bandwidth B_i, a VM will be "poor" if $D_i < B_i$. Otherwise, if a VM's bandwidth demand is higher than its base bandwidth $(D_i \geq B_i)$, it will be "rich" with higher rate, which is at least be guaranteed with the base bandwidth. Based on the above model and notions, the desirable requirements of our allocation policy are as follows.

1) Predictable performance: **Minimum bandwidth guarantee** is to ensure a certain bandwidth allocation for a VM irrespective of the bandwidth demands from other VMs. It provides strong network isolation for VMs and is the basic property to achieve predicable performance in data center networks. In our model, we guarantee the base bandwidth for VMs, which implies that we should satisfy the traffic demand for those poor VMs. This is naturally acceptable as the poor VMs only consume bandwidth that belongs to themselves, if cloud providers charge for such guaranteed bandwidth. However, rather than reserving the base bandwidth in a fixed manner, we allocate bandwidth according to the actual instantaneous demand of these VMs, so as to improve the entire utilization of data center networks.

Fair bandwidth allocation intends to share bandwidth by extending the notion of fairness at flow level to VM-level. Network resources in today's data center are shared in a best-effort manner among VMs or servers, which makes it hard for cloud providers to assign certain weights to different applications. By using weighted proportional share among VMs, a VM can achieve a certain share of the total amount of available bandwidth without dictating the bandwidth competitions from flow-level. For consideration of generality, we propose to use the general notion of fairness in game theory [12]. The advantage is that the game-theoretic framework can achieve very flexible solutions corresponding to either max-min fairness or proportional fairness, by customizing specific objective functions (Sec. IV).

In this paper, we choose to maintain such fairness among VM-pairs. One of the most obvious advantages is that it can provide fine-grained resource management in data centers. For example, the reduce node in a MapReduce job may shuffle data from several map nodes (in separate VMs), and the congestion of any shuffle process will delay the job completion time.

The tradeoff between minimum guarantee and network proportionality is discussed in [2]. To resolve such potentially conflicting objectives, we first guarantee the bandwidth demand from poor VMs, and then consider the rich VMs by allocating the residual bandwidth under the principle of fairness.

2) High Utilization: This implies that datacenter network resources should be fully used when there are unsatisfied demands. By utilizing the idle network resources in data centers, VMs can improve their throughput and thus shorten the job completion times for tenants. For cloud providers, they are able to improve their revenue by deploying more VMs with the same underlying physical resources. Our overall objective is to utilize precious bandwidth resources in a data center as much as possible, while maintaining our aforementioned goals for predictable performance.

IV. Nash bargaining solution for fairness and utilization

Based on the datacenter model and network requirements, we first derive a centralized solution by taking advantage of the *Nash bargaining framework* and the method of Lagrange multipliers.

A. Nash Bargaining Framework

Nash bargaining has been widely used to balance the tradeoff between fairness and efficiency for resource sharing problems. Under the context of bandwidth allocation in a data center, the game can be described as follows: N VMs can be viewed as players who are competing for the bandwidth resources of their hosting servers. Each VM is assigned an initial rate u_i^0, which is the bandwidth to be guaranteed.

Let $\mathcal{X} \subset \mathcal{R}^N$ be the vector of available bandwidth allocation space for N VMs in a data center, where $\mathcal{R}^N$ is the set of all allocation strategies. Given a vector of initial rates $u^0 = (u_1^0, \ldots, u_N^0)$, and a set of bandwidth allocation $\mathcal{U} = \{x \mid x \in \mathcal{X}, x_i \geq u_i^0, \forall i\}$ which ensures that each VM can get at least their initial bandwidth, we assume that $\mathcal{U} \subset \mathcal{R}^N$ is a nonempty convex closed and upper-bounded set. Let $\mathcal{J} = \{j \mid x \in \mathcal{U}, x_j > u_j^0\}$ denote the set of VMs that can achieve strictly higher bandwidth compared to their initial rates. The pair $(\mathcal{U}, u^0)$ is called a bargaining problem.

Definition 1. A mapping $S : (\mathcal{U}, u^0) \rightarrow \mathcal{R}^N$ is called a Nash bargaining solution if it satisfies: $S(\mathcal{U}, u^0) \in \mathcal{U}$, Pareto optimality, linearity axiom, irrelevant alternatives, and symmetry axiom [13].

Suppose $\mathcal{X}$ is a convex and compact subset of $\mathcal{R}^N$ and $\mathcal{J}$ is nonempty, we can derive the following theorem.

Theorem 1. *There exists a bargaining solution. The vector x of the solution set solves the following optimal problem (P_J):*

$$\max \prod_{j \in J} (x_j - u_j^0), \quad x \in \mathcal{X}. \tag{1}$$

In the Nash bargaining game, two or more players enter the game with an initial utility as well as an utility function. They cooperate in the game to achieve a win-win solution, in which the social utility gains (represented by the Nash product in Eq. (1)) are maximized. This corresponds to the requirements in sharing data center networks, with respect to guarantee certain bandwidth for VMs, maintain fairness among them and achieve high network utilization across the data center. By taking the logarithm of the objective, we can derive an equivalent optimization problem:

$$\max \sum_{j \in J} \ln(x_j - u_j^0), \quad x \in \mathcal{X}.$$

The above maximization problem has a unique bargaining solution equivalent to (P_J). We refer the reader to [14] for details of the Nash bargaining solution.

B. Optimization Problem based on NBS

Based on the bargaining framework, we now define the optimization problem in our model, which aims to achieve the NBS for network resource allocation. First, for a VM $(r_i^I, r_i^E, D_i^I, D_i^E, B_i^I)$, with given base bandwidth and bandwidth demand, the rate $r_{i,j}$ is what we need to obtain and allocate to the connection from v_i to v_j. The allocation is bounded by the physical bandwidth of each server, i.e., $\sum_{v_i \in V_{p_j}} r_i \leq C_j$ for both ingress and egress bandwidth capacities. To guarantee the minimum bandwidth, we use the constraint $r_{i,j} \geq L_{i,j}$ to ensure that each VM can obtain an initial minimum bandwidth, thus

$$L_{i,j} = \min\{D_{i,j}, B_{i,j}\}. \tag{2}$$

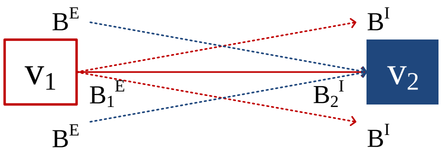

Fig. 2: Guarantee minimum bandwidth $B_{1,2}$ from VM v_1 to v_2.

$B_{i,j}$ represents the base bandwidth for the connection of v_i to v_j, which needs to be specified by using the respective base bandwidth of VMs and the traffic demands across these VMs. Specifically, we use a simple method to specify $B_{i,j}$ as follows:

$$B_{i,j} = \min\{B_i^E \frac{B_j^I}{\sum_{D_{ik} \neq 0} B_k^I}, B_j^I \frac{B_i^E}{\sum_{D_{kj} \neq 0} B_k^E}\}, \tag{3}$$

$\forall k \in \{1, 2, \ldots, N\}$, where $D_{ij} \neq 0$ implies that there exists certain traffic from v_i to v_j.

As shown in Fig. 2, each VM may be in communication with one or more VMs. We explain base bandwidth $B_{i,j}$ of a VM pair as (i) a portion of the egress base bandwidth of VM v_i, or (ii) a portion of the ingress base bandwidth of VM v_j. Since the base bandwidth is the minimum guarantee for VMs, it indicates the weights of the VMs. In other words, a connection should obtain a higher base bandwidth if it is associated with VMs of higher base bandwidth. Hence, the portion of $B_{i,j}$ to B_i^E is set as the portion of the ingress base bandwidth of VM v_j to all the destination VMs from source VM v_i. This is similar for the portion of $B_{i,j}$ to B_i^I. Then $B_{i,j}$ is determined by the smaller value in case that the total base bandwidth exceeds the physical bandwidth, causing failure on guaranteeing the base bandwidth $B_{i,j}$.

After determining the minimum bandwidth guarantee for each connection via Eq. (3), we have the joint profit optimization problem (P_r) as follows:

$$\max_{r} \quad \sum_{j} \sum_{i} \ln(r_{i,j} - L_{i,j}) \tag{4}$$

$$\text{s.t.} \quad r_{i,j} \leq U_{i,j}, \ \forall i, j \in \{1, \ldots, N\}, \tag{5}$$

$$r_{i,j} \geq L_{i,j}, \ \forall i, j \in \{1, \ldots, N\}, \tag{6}$$

$$\sum_{v_i \in V_{p_m}} r_i^I \leq C_m, \ \forall p_m \in \mathcal{M}, \tag{7}$$

$$\sum_{v_i \in V_{p_m}} r_i^E \leq C_m, \ \forall p_m \in \mathcal{M}, \tag{8}$$

where $r_i^I = \sum_{j=1}^N r_{j,i}$ and $r_i^E = \sum_{j=1}^N r_{i,j}$ are total ingress and egress rate of VM v_i, respectively.

$U_{i,j}$ is the upper bound for bandwidth allocation from v_i to v_j, which is denoted by

$$U_{i,j} = \min\{D_{i,j}, C_m, C_n\}, \text{s.t. } v_i \in p_m, v_j \in p_n. \quad (9)$$

C. Centralized Optimal Solution

We use the matrix $V = (v_{mi})_{M\times N}$ to denote the placement of VMs on each server, where v_{mi} is a binary variable defined as follows:

$$v_{mi} = \begin{cases} 1, & v_i \text{ is on } p_m; \\ 0, & \text{otherwise.} \end{cases}$$

Let $r^I = (r^I_1, r^I_2, \ldots, r^I_N)$ ($r^E = (r^E_1, r^E_2, \ldots, r^E_N)$) be the vector of ingress (egress) rates of VMs and $C = (C_1, C_2, \ldots, C_N)$ be the vector of bandwidth of the servers. To derive the formal optimal solution to problem P_r, we apply the method of Lagrange multipliers. Note that the constraints of the variable r are linear, the Kuhn-Tucher conditions are necessary and sufficient for an existing optimal solution.

Theorem 2. *There exists $\gamma^I_m \geq 0$ and $\gamma^E_m \geq 0$ $(m \in \{1, 2, \ldots, M\})$ such that*

- $\forall i, j \in \{1, \ldots, N\}$:
$$r^*_{i,j} = L_{i,j} + \frac{1}{\sum_{m=1}^{M} \gamma^E_m v_{mi} + \sum_{m=1}^{M} \gamma^I_m v_{mj}}, \quad (10)$$
$$L_{i,j} \leq r_{i,j} \leq U_{i,j},$$
- $\forall m \in \{1, 2, \ldots, M\}$:
$$\gamma^I_m \big[(V \cdot r^I)_m - C_m\big] = 0,$$
$$\gamma^E_m \big[(V \cdot r^E)_m - C_m\big] = 0,$$

where $r^*_{i,j}$ is the unique NBS for the optimization problem (P_r).

Proof: We begin with an assumption that the allocation space $X \in \mathcal{R}^{N\times N}$ is a nonempty, convex and compact set. Define

$$\theta(r) = \sum_{j=1}^{N} \sum_{i=1}^{N} \ln(r_{i,j} - L_{i,j}),$$

then $\theta(\cdot) : X \to \mathcal{R}^+$ is strictly concave [13].

Let $\alpha_{i,j} \geq 0$, $\forall i, j \in \{1, 2, \ldots, N\}$, $\beta_{i,j} \geq 0$ $\forall i, j \in \{1, 2, \ldots, N\}$, and $\gamma^I_m, \gamma^E_m \geq 0$, $\forall m \in \{1, 2, \ldots, M\}$ denote the Lagrange multipliers for the minimum bandwidth in Eq. (5), upper-bound bandwidth in Eq. (6) and server capacity in Eq. (7) and Eq. (8), respectively. Then, the Lagrangian of the problem P_r is

$$\begin{aligned} \mathcal{L}(r, \alpha, \beta, \gamma^I, \gamma^E) &= \theta(r) - \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_{i,j}(L_{i,j} - r_{i,j}) \\ &- \sum_{j=1}^{N} \sum_{i=1}^{N} \beta_{i,j}(r_{i,j} - U_{i,j}) - \sum_{m=1}^{M} \gamma^I_m \big((V \cdot r^I)_m - (C)_m\big) \\ &- \sum_{m=1}^{M} \gamma^E_m \big((V \cdot r^E)_m - (C)_m\big). \end{aligned}$$

To give the necessary and sufficient conditions for optimizing the objective, we have

$$\nabla \mathcal{L}(r^*, \alpha, \beta, \gamma^I, \gamma^E) = 0 \Longleftrightarrow$$

$$\frac{1}{r^*_{i,j} - L_{i,j}} + \alpha_{i,j} - \beta_{i,j} - \sum_{m=1}^{M} \gamma^I_m v_{mj} - \sum_{m=1}^{M} \gamma^E_m v_{mi} = 0, \quad (11)$$

$\forall i, j \in \{1, 2, \ldots, N\}$ and

$$\begin{cases} \alpha_{i,j}(L_{i,j} - r^*_{i,j}) = 0, & i, j \in \{1, 2, \ldots, N\}, \\ \beta_{i,j}(r^*_{i,j} - U_{i,j}) = 0, & i, j \in \{1, 2, \ldots, N\}, \\ \gamma^I_m \big[(V \cdot r^I)_m - (C)_m\big] = 0, & m \in \{1, 2, \ldots, M\}, \\ \gamma^E_m \big[(V \cdot r^E)_m - (C)_m\big] = 0, & m \in \{1, 2, \ldots, M\}, \end{cases} \quad (12)$$

where $r^* = (r^*_{1,1}, \ldots, r^*_{i,j}, \ldots, r^*_{N,N})$ is the optimal solution to the problem P_r.

To derive the solutions, we should consider the values of the multipliers in Eq. (12). For constraints $r_{i,j} = L_{i,j}$ and $r_{i,j} = U_{i,j}$, they represent the special case when $r_{i,j}$ reaches the boundary value. Hence, we focus on such a general situation that $L_{i,j} < r_{i,j} < U_{i,j}$, which can derive $\alpha_{i,j} = 0$ and $\beta_{i,j} = 0$. Then we can obtain $r_{i,j}$ by solving Eq. (11). ■

In summary, by solving the centralized problem, we have shown how to achieve our design goals on sharing data center networks with consideration on bandwidth guarantee, fairness as well as utilization. The original problem P_r is a convex optimization with $2(M+N)$ constraints, whose computational complexity may increase significantly as the number of VMs and servers scales up.

Fortunately, the solution in Eq. (10) indicates that each optimal rate $r_{i,j}$ can be solved by the optimal multipliers associated with two servers, i.e., the server hosting the source VM v_i and the server hosting the destination VM v_j. Hence, the key to maximize the objective is to obtain the optimal Lagrange multiplier of each server, which is independent from other servers. This motivates us to obtain the rate of each VM-pair in a distributed manner, rather than being restricted to a centralized approach.

V. Distributed Cooperative game based algorithm

In this section, we present a distributed algorithm based on the cooperative game framework. Specially, we apply the gradient projection methods in constrained optimization to guide the design of the distributed algorithm.

A. Dual-Based Decomposition

The centralized primal problem can be solved by the dual-based decomposition. Specifically, we consider the primal problem which has the same optimal solution as problem P_r.

$$\begin{aligned}\min_{r} \quad & P(r) = -\sum_{j=1}^{N}\sum_{i=1}^{N}\ln(r_{i,j}-L_{i,j}) && (13)\\ \text{s.t.} \quad & r_{i,j} \le U_{i,j},\ \forall i,j \in \{1,\dots,N\}, && (14)\\ & r_{i,j} \ge L_{i,j},\ \forall i,j \in \{1,\dots,N\}, && (15)\\ & \sum_{v_i \in V_{p_m}} r_i^I \le C_m,\ \forall p_m \in \mathcal{M}, && (16)\\ & \sum_{v_i \in V_{p_m}} r_i^E \le C_m,\ \forall p_m \in \mathcal{M}. && (17)\end{aligned}$$

We mainly discuss the situation that $L_{i,j} < r_{i,j} < U_{i,j}$. Let X be the allocation space of r defined in Sec. IV-A. The Lagrangian associated with the primal problem in Eq. (13) is defined as $\mathcal{L}(\cdot) : X \times \mathcal{R}^M \times \mathcal{R}^M \to \mathcal{R}$, where

$$\begin{aligned}&\mathcal{L}(r,\gamma^I,\gamma^E) = -\sum_{j=1}^{N}\sum_{i=1}^{N}\ln(r_{i,j}-L_{i,j})\\ &+\sum_{m=1}^{M}\gamma_m^I\big((V\cdot r^I)_m - C_m\big)+\sum_{m=1}^{M}\gamma_m^E\big((V\cdot r^E)_m - C_m\big).\end{aligned}$$

Note that γ^I and γ^E are the dual variables associated with the problem. The Lagrange dual function $d(\cdot) : \mathcal{R}^M \times \mathcal{R}^M \to \mathcal{R}$ corresponding to $\mathcal{L}(r,\gamma^I,\gamma^E)$ is expressed as

$$d(\gamma^I,\gamma^E) = \inf_{r\in\mathcal{R}^{N\times N}} \mathcal{L}(r,\gamma^I,\gamma^E).$$

Since the primal problem has a unique optimal solution, the dual function yields lower bounds on each optimal $r_{i,j}^*$ which solves Eq. (13). For any $\gamma^I, \gamma^E \in \mathcal{R}^M$, we have $d(\gamma^I,\gamma^E) \le \mathcal{L}^*$, where $\mathcal{L}^*$ is the infimum of $\mathcal{L}(r,\gamma^I,\gamma^E)$. The infimum of Lagrangian occurs where the gradient is equal to 0, thus $r_{i,j}^* = L_{i,j} + 1/(\sum_{m=1}^{M}\gamma_m^E v_{mi} + \sum_{m=1}^{M}\gamma_m^I v_{mj})$ according to Theorem 2.

It is obvious that there exists such r that $\forall m \in \{1,2,\dots,M\}$, we have $(V\cdot r)_m < C_m$ and $L_{i,j} < r_{i,j} < U_{i,j}$. It implies that there exists r in the relative interior of the intersection of the domain of all constraint functions. And since X is convex and $P(r)$ is convex over X, the Slater's condition holds, which is a sufficient condition for strong duality [14]. Hence, there is no duality gap and there exists γ^I and γ^E satisfying $d(\gamma) = \mathcal{L}^*$.

In summary, we get the dual problem corresponding to the primal problem with no duality gap. The dual problem (P_d) is described as follows:

$$\max_{\gamma^I,\gamma^E\in\mathcal{R}^M} d(\gamma^I,\gamma^E) = \mathcal{L}(r^*,\gamma^I,\gamma^E), \tag{18}$$

The optimal solution of the dual problem can also be derived through the set of Lagrange multipliers, and it can be characterized by the following gradient projection method.

B. Gradient Projection Method

To solve the primal problem P_r, we first obtain the optimal solution to the dual problem P_d. Specifically, by using a suitable step-size, we design an algorithm that converges to the optimal γ^I (γ^E) in the gradient projection method. Since the strong duality holds as discussed above, we can achieve the optimal Nash bargaining solution with Eq. (10).

Let the set $\overline{\Gamma}$ denote the optimal solution to the dual problem and the set $\overline{\mathcal{R}}$ be the solution to the primal one. We define the following recursion:

$$\gamma_m^{(k+1)} = \max(0, \gamma_m^{(k)} + \xi\frac{\partial d}{\partial \gamma_m}), \forall m \in \{1,2,\dots,M\}, \tag{19}$$

where ξ is the step-size. We first discuss the sequence for γ^I, while regarding γ^E as a constant.

Theorem 3. *For the recursive sequence $\{\gamma^{I(k)}\}$, if $\gamma^{I(0)} \in \mathcal{R}^{+M}$ and $\xi \in (0, \frac{2}{K}]$, then $\{\gamma^{I(k)}\}$ converges, thus*

$$\lim_{k\to\infty}\gamma^{I(k)} = \gamma^{I*} \in \overline{\Gamma}, \tag{20}$$

where K is the Lipschitz constant [15] of the dual function in Eq. (18), such that

$$K = \sqrt{M}\sum_{j=1}^{N}\sum_{i=1}^{N}(U_{i,j}-L_{i,j})^2. \tag{21}$$

Proof: The first step is to characterize that the dual function $d(\gamma)$ is K-Lipschitz continuous. We define a function $h_{i,j}(x) = L_{i,j} + \frac{1}{x}$, $\forall i,j \in \{1,2,\dots,N\}$, where $x \ge \frac{1}{U_{i,j}-L_{i,j}}$.

By changing the variable of $d(\gamma^I,\gamma^E)$ and replacing $r_{i,j}^*$ in Eq. (18) with

$$r_{i,j}^*(\gamma^I,\gamma^E) = h_{i,j}(\sum_{m=1}^{M}\gamma_m^E v_{mi} + \sum_{m=1}^{M}\gamma_m^I v_{mj}), \tag{22}$$

we get the partial derivatives of $d(\gamma^I,\gamma^E)$ as

$$\frac{\partial d(\gamma^I,\gamma^E)}{\partial \gamma_m^I} = \sum_{j=1}^{N} v_{mj}\sum_{i=1}^{N} r_{i,j}^* - C_m. \tag{23}$$

It is explicit that the Lipschitz constant of the real function $h_{i,j}$ is $(U_{i,j}-L_{i,j})^2$. If we define $f_m(\gamma) = \sum_{j=1}^{N} v_{mj}\sum_i h_{i,j}(\gamma)$, then we can get the following relationship, $\forall\gamma$ and $\gamma' \in \mathcal{R}^M$:

$$\begin{aligned}&|f_m(\gamma) - f_m(\gamma')|\\ \le\ & \sum_{j=1,v_{mj}=1}^{N}|\sum_{i=1}^{N}h_{i,j}(\gamma) - \sum_{i=}^{N}h_{i,j}(\gamma')|\\ \le\ & \Big[\sum_{j=1,v_{mj}=1}^{N}\sum_{i=1}^{N}(U_{i,j}-L_{i,j})^2\Big]\|\gamma-\gamma'\|_1,\end{aligned}$$

where $\|\cdot\|_1$ is the L_1 norm in vector space. Summing up both sides of the equation for $m \in \{1,2,\dots,M\}$ yields

$$\begin{aligned}&\|f(\gamma) - f(\gamma')\|_1\\ \le\ & \Big[\sum_{m=1}^{M}\sum_{j=1,v_{mj}=1}^{N}\sum_{i=1}^{N}(U_{i,j}-L_{i,j})^2\Big]\|\gamma-\gamma'\|_1\\ =\ & \sum_{j=1}^{N}\sum_{i=1}^{N}(U_{i,j}-L_{i,j})^2\|\gamma-\gamma'\|_1.\end{aligned}$$

For $\forall \gamma \in \mathcal{R}^M$, we have $\|\gamma\| \leq \|\gamma\|_1 \leq \sqrt{M}\|\gamma\|$ in metric space, where $\|\cdot\|$ or $\|\cdot\|_2$ is the Euclidean norm. Hence we obtain the following inequality

$$\begin{aligned} & \|f(\gamma) - f(\gamma')\| \\ \leq \quad & \sqrt{M}\Big(\sum_{j=1}^{N}\sum_{i=1}^{N}(U_{i,j} - L_{i,j})^2\Big)\|\gamma - \gamma'\|. \end{aligned}$$

The Lipschitz constant can be derived from the above relationship. It has been proved in [13] that when $d(\cdot)$ is K-Lipschitz continuous, $(\gamma^I)^{(k)}$ converges in $\overline{\Gamma}$ as $k \to \infty$ if we choose such a step size that $\xi \in (0, \frac{2}{K}]$. ■

For the sequence $\{\gamma^{E(k)}\}$, we have the conclusion that the partial derivative is derived as follows

$$\frac{\partial d(\gamma^I, \gamma^E)}{\partial \gamma_m^E} = \sum_{j=1}^{N} v_{mj} \sum_{i=1}^{N} r_{i,j}^* - C_m. \tag{24}$$

In Theorem 2, we have obtained the explicit form of optimal rate $r_{i,j}^*$. The sequences generated by Eq. (19) converge to the optimal solution to the dual problem (P_d) in Eq. (18) according to Theorem 3. Since there is no duality gap in the dual decomposition, the rate vector r associated with γ^I and γ^E converges to the Nash bargaining solution, thus $\forall i, j \in \{1, 2, \ldots, N\}$

$$\lim_{k\to\infty} r(\gamma^{I(k)}, \gamma^{E(k)}) = r^* \in \overline{R}. \tag{25}$$

C. Distributed Cooperative Algorithm

Based on the theoretical guidelines above, we are able to design a distributed allocation algorithm in Algorithm 1.

From the sequence in Eq. (19), the dual variables γ^I and γ^E decrease within each update. The update rate is determined by the step-size and the gradient of the dual function. Specifically, the step-size ξ is bounded by the global constant $\frac{2}{K}$ in Theorem 3, and the absolute value gradient is equal to the unused bandwidth of each server.

Let $r_{p_m}^E$ and $r_{p_m}^I$ represent the allocated egress and ingress bandwidth of server p_m, respectively. Without loss of generality, suppose v_i is hosted on a certain server p_m and v_j is hosted on a certain server p_n, we use the notation γ^E and γ^I to represent γ_n^I and γ_m^E in Algorithm 1, respectively. Then the algorithm is a parallel iteration of $r_{i,j}^*$ with the dual variables γ^I and γ^E. For example, if there exists unsatisfied demand for $r_{i,j}$ and residual bandwidth on server p_m, the dual variable γ_m^E will decrease and the server allocates its residual bandwidth to $r_{i,j}$. Otherwise, if the total allocated rates exceed the bandwidth capacity of server p_m, the dual variable γ_m^E will increase so as to reduce the excessive bandwidth of p_m.

Implementation Issues. As the algorithm to update each $r_{i,j}$ can be performed in parallel on each server, the algorithm can be implemented in the hypervisor by enforcing a limited rate for each VM [5]; and each server only manages those connections with source VMs on this server. Since the allocated bandwidth is always lower than (or equal to) bandwidth demand, the allocated bandwidth can be fully utilized by VMs. The rate of convergence is determined by the step-size ξ (characterized by K) and the initial price vector γ^0. Since K is a global constant, we need to broadcast it to all the servers before performing the iteration. We will examine the rate of convergence in Sec. VI.

Algorithm 1: Distributed Algorithm for Bandwidth Allocation

```
Input:
    The step-size characterized by $K$: $\xi \in (0, \frac{2}{K}]$;
    Server bandwidth capacity: $C_m, \forall p_m \in \mathcal{M}$;
    Bandwidth demand matrix of VMs: $[D_{i,j}]_{N\times N}$,
    $\forall i \in \{1, 2, \ldots, N\}$;
    VM placement across servers: $[v_{m,i}]_{M\times N}$,
    $\forall m \in \{1, 2, \ldots, M\}, \forall i \in \{1, 2, \ldots, N\}$;
    The base bandwidth of VMs $< B_i^I, B_i^E >$, $\forall v_i \in \mathcal{N}$;
    The rounds of iteration: $S$;
Output:
    Rate allocation matrix, $[r_{i,j}]_{N\times N}$, $\forall i \in \{1, 2, \ldots, N\}$
for all $r_{i,j}$ do
    Initialize $L_{i,j}$ and $U_{i,j}$ by Eq. (2) and (9), respectively;
    $r_{i,j}$ = $L_{i,j}$;
end for
while steps < S do
    for all server $p_m$ do
        Update $r_{p_m}^E = \sum_i v_{mi} r_i^E$, $r_{p_m}^I = \sum_i v_{mi} r_i^I$;
        $\gamma_m^E = \max(0, \gamma_m^E - \xi(C_m - r_{p_m}^E))$ (Eq. 19);
        $\gamma_m^I = \max(0, \gamma_m^I - \xi(C_m - r_{p_m}^I))$ (Eq. 19);
    end for
    for all $r_{i,j}$ do
        if $\frac{1}{\gamma^E + \gamma^I} \leq U_{i,,j} - L_{i,j}$ then
            $r_{i,j} = U_{i,j}$ (Eq. 10);
        else
            $r_{i,j} = L_{i,j} + \frac{1}{\gamma^E + \gamma^I}$
        end if
    end for
    steps++;
end while
```

VI. Performance Evaluation

A. Simulation Setup

We simulate an IaaS data center with homogeneous servers, each equipped with equal ingress and egress network bandwidth capacity of 1 Gbps. Jobs in the data center are processed in VMs which have already been placed on these servers. In our simulations, we assume that the data center has full bisection bandwidth network (following [2]) as the consideration in Sec. III-A, where the server has the ability to shape the bandwidth of its hosting VMs.

We focus on verifying whether our allocation algorithm can satisfy the proposed requirements. Specifically, 1) we examine the bandwidth allocation to different VMs and quantify the convergence of our algorithm by specifying certain demand matrix. Then we observe how different types of VMs compete for bandwidth resources with different bandwidth demands; 2)

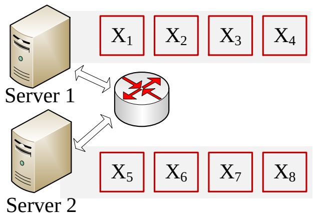

Fig. 3: Sharing network with traffic from server 2 to server 1.

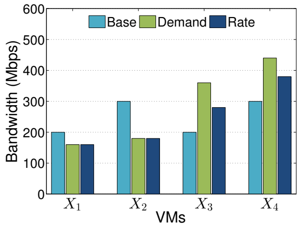

Fig. 4: Bandwidth allocation to VMs on server 1 with different demands and base bandwidths.

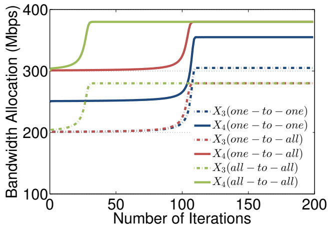

Fig. 5: Bandwidth allocation to VMs on server 1 with increasing demand of X_1 for all-to-all communication patterns.

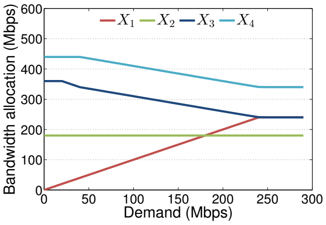

Fig. 6: Rates of VM on server 1 with increasing number of iterations.

by changing the base bandwidth of VMs, we test our algorithm in several typical scenarios and observe how it balances the tradeoff between fair share and minimum guarantee.

B. Simulation Results

For the following experiments, we consider a scenario illustrated in Fig. 3, where VMs on server 2 send (or receive) traffic to (or from) VMs on server 1. Since our algorithm uses different strategies for different types of VMs, i.e., bandwidth guarantee for poor VMs and fair bandwidth share for rich VMs, the simulation should involve both poor and rich VMs. We qualify the algorithm within three typical communication patterns: 1) one-to-one (i.e., X_i to X_{i+4}, $i = 1, \ldots, 4$), 2) one-to-all (i.e., X_1 to $X_5 \sim X_8$) and 3) all-to-all (i.e., X_i to $X_5 \sim X_8$, $i = 1, \ldots, 4$), each consists of 4 poor VMs and 4 rich VMs.

Bandwidth allocation for VMs. Due to similarity of bandwidth allocation under three scenarios, we only plot the bandwidth of the VMs on server 1 with all-to-all communication pattern. As we can see in Fig. 4, our algorithm guarantees the bandwidth demand of VM X_1 and X_2, whose demands are less than their base bandwidth, i.e., poor VMs. For VM X_3 and VM X_4, they are allocated with bandwidth higher than the minimum guarantee, by using the residual bandwidth capacity after the guaranteed bandwidth allocations on the server. This verifies that our NBS based solution achieves high utilization when there are unsatisfied demands. In addition, since the connections associated with VM X_3 and X_4 share the same source and destination servers, and they also share the same changing rates in the iterations, they get equal excessive bandwidth compared to their own base bandwidth, which implies that the fairness fashion in NBS is to fairly share the residual capacity.

Rate of convergence. Fig. 5 shows the rates of the VMs changing within the iterations of the algorithm under the above three scenarios. The rates of VMs converge under the control of our allocation algorithm, which verifies the correctness of the algorithm. For each convergence process, the rate changes slowly in the beginning and then quickly converges to a stable value. This is because in our model, the small constant step size (about 10^{-3} in these scenarios) may restrict the increase of rate r, i.e., $\frac{1}{\gamma^E+\gamma^I}$, which can be 1 Mbps at most when the dual variables are larger than 1. Hence, it is necessary to choose a proper initial value for the dual variables so as to shorten the iterative process.

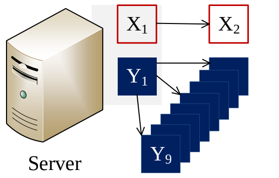

Fig. 7: X_1 communicates with one VM, while Y_1 communicates with an increasing number of VMs.

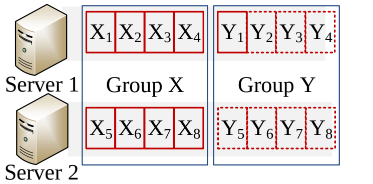

Fig. 8: Sharing network with traffic from server 2 to server 1 by increasedly placing the VMs of group Y on server 1.

Varying the bandwidth demand of VMs. To study the interference between VMs caused by the changes in demand, we run a similar experiment with all-to-all communication pattern by varying the demand of one VM. The bandwidth allocation to VMs on server 2 is shown in Fig. 6. When the bandwidth demand of VM X_1 is less than its base bandwidth (from 0 Mbps to 200 Mbps), the demand of the poor VMs (VM X_1 and X_2) is hardly guaranteed irrespective of such variation. If the bandwidth demand of VM X_1 is above its base bandwidth, it suffers competition from other VMs but can still get a fair share of the residual bandwidth capacity. The amount of share arrives at a maximum value of 240 Mbps, when its bandwidth demand grows higher than 240 Mbps. Such observation indicates that the algorithm can maintain fairness among the rich VMs, by allocating the residual bandwidth in a fairness fashion based on the Nash bargaining solution. In addition, the total amount of the bandwidth allocations to the VMs can always cap the server's bandwidth capacity, which illustrates that the algorithm can fully utilize the congested bandwidth.

C. Exploring the Tradeoff

Since the base bandwidth represents the amount of bandwidth that should be firmly guaranteed, and the total base bandwidth of VMs located on one server determines the residual bandwidth that can be fairly shared, the base bandwidth can be viewed as a tunable design knob of balancing the tradeoff between minimum bandwidth guarantee and fair share. We carry out the following experiments to verify whether the algorithm can provide flexibility on achieving these two requirements, by varying the base bandwidth of VMs.

In the first case, we focus on a single congested link. We use a scenario where two VMs, X_1 and Y_1, are co-located on the same server as shown in Fig. 7. VM X_1 communicates with

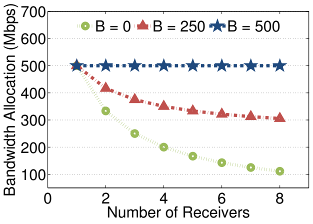

Fig. 9: Network allocation to VM X_1 when increasing receivers of Y_1 with different base bandwidth.

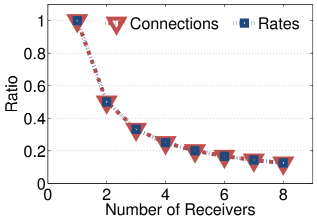

Fig. 10: Network allocation to tenant X when increasing the ratio of sender to receivers in Y_1 with different base bandwidth.

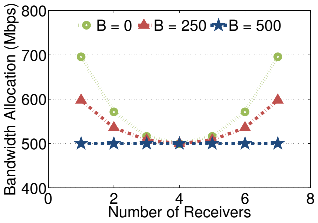

Fig. 11: Ratio of connections of tenant X to Y and bandwidth allocation for X to Y when increasing receivers of Y_1.

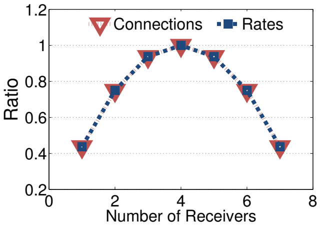

Fig. 12: Ratio of connections of tenant Y to X and bandwidth allocation for Y to X when increasing the ratio of sender to receivers.

one VM, while Y_1 communicates with an increasing number of VMs each time. All the VMs are assigned with the same base bandwidth B, so as to exclude the influence caused by the difference of base bandwidth. Since the VMs connected to Y_1 may become the bottleneck when sending (receiving) data, we place each of them on an individual host.

Fig. 9 plots the bandwidth allocation to VM X_1 with $B = 0, 250, 500$ Mbps. When $B = 500$ Mbps, the policy guarantees the bandwidth of X_1 regardless of communications of Y_1. When $B = 250$ Mbps, the bandwidth allocated for VM X_1 drops as Y_1 communicates with more VMs, but maintains at least 300 Mbps. The amount of bandwidth shared by VM X_1 indicates that the policy consists of both a guaranteed portion and a proportionally shared portion. By setting $B = 0$, we can obtain the ratio of bandwidth shared by X_1 to Y_1 in Fig. 10 and observe that the bandwidth shared by a VM is in proportion to the number of its connected VMs. Since our policy maintains fairness among VM-pairs, the result shows that the policy only considers fair share when the base bandwidth is 0.

In the next simulation, we use a scenario illustrated in Fig. 8, where two groups (each with 8 VMs) are placed on two servers. We change the placement of VMs in group Y by increasing the number of VMs on server 1, from 1 to 7. We plot the same performance metrics as the previous simulation in Fig. 11 and Fig. 12, by varying the total amount of base bandwidth B of VMs in group X and Y, respectively. When $B = 500$ Mbps, as shown in Fig. 11, there is no residual bandwidth that can be shared and the policy only involves the idea of bandwidth guarantee intuitively. But as B decreases, the policy tends to prefer fair bandwidth sharing. When it comes to 0, all the bandwidth are shared in a fairness fashion in the bargaining game, where the shared bandwidth is in proportion to the number of connections within its VM group.

In summary, we validate the tradeoff between minimum bandwidth guarantee and fair share by turning the knob, i.e., the base bandwidth. With its flexibility on balancing such a tradeoff, our proposed algorithm enables flexible design choices for cloud providers to fine-tune their data center management tools for meeting tenants' diverse needs.

VII. Conclusion

This paper takes a first step towards using game-theoretic approaches to share multi-tenant data center networks, by applying rigorous cooperative game framework and developing a distributed algorithm that achieves the Nash bargaining solution. Our allocation policy can simultaneously achieve high network utilization for cloud providers and predictable performance for tenants, by guaranteeing bandwidth for VMs with bandwidth demand lower than their base bandwidth, as well as maintaining fairness among VMs with bandwidth demand higher than their base bandwidth. Extensive simulations under typical scenarios show that by tuning the base bandwidth as a design knob, our policy enables large flexibility for cloud providers to balance the tradeoff between bandwidth guarantee and fair bandwidth share.

References

[1] Amazon elastic compute cloud. [Online]. Available: http://aws.amazon.com

[2] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, and I. Stoica, "Faircloud: Sharing the network in cloud computing," in *Proc. of ACM SIGCOMM*, 2012.

[3] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *Proc. of ACM SIGCOMM*, 2011.

[4] C. Guo, G. Lu, H. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "Secondnet: A data center network virtualization architecture with bandwidth guarantees," in *Proc. of ACM CoNEXT*, 2010.

[5] H. Rodrigues, J. Santos, Y. Turner, P. Soares, and D. Guedes, "Gatekeeper: Supporting bandwidth guarantees for multi-tenant datacenter networks," in *Proc. of 3rd Workshop on I/O Virtualization*. USENIX, 2011.

[6] T. Lam and G. Varghese, "Netshare: Virtualizing bandwidth within the cloud," UCSD, Tech. Rep., 2009.

[7] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, "Sharing the data center network," in *Proc. of USENIX NSDI*, 2011.

[8] Y. Feng, B. Li, and B. Li, "Bargaining towards maximized resource utilization in video streaming datacenters," in *Proc. of INFOCOM*, 2012.

[9] P. Mishra, K. Ramakrishnan, and J. van der Merwe, "A flexible model for resource management in virtual private networks," in *Proc. of ACM SIGCOMM*, 1999.

[10] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath tcp," in *Proc. of ACM SIGCOMM*, 2011.

[11] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, "Vl2: a scalable and flexible data center network," in *Proc. of ACM SIGCOMM*, 2009.

[12] Z. Fang and B. Bensaou, "Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks," in *Proc. of IEEE INFOCOM*, 2004.

[13] H. Yaïche, R. Mazumdar, and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 8, no. 5, pp. 667–678, 2000.

[14] A. Muthoo, *Bargaining Theory With Applications*. Cambridge University Press, 1999.

[15] D. Bertsekas, *Nonlinear programming*. Athena Scientific, 1995.