Throughput Maximization of Virtual Machine Communications in Bandwidth-Constrained Data Centers

Jeff Lutz, Bin Tang, Christopher Gonzalez
Department of Computer Science
California State University Dominguez Hills, Carson, CA 90747, USA
Email: {jlutz155@gmail.com, btang@csudh.edu, cgonzalez393@toromail.csudh.edu}

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9686028

***Abstract*—In this paper we study a new algorithmic problem that maximizes the throughput of virtual machine (VM) communication in bandwidth-constrained data centers. Given a set of VM pairs with different bandwidth demands that are already placed inside cloud data centers, we study how to allocate the network bandwidth to the VM pairs to accommodate maximum number of VM communication while considering that cloud data centers have limited bandwidths. We refer to this throughput maximization problem as VMB. Due to the massive growth of cloud communication traffic in recent years and that service providers attempt to accommodate as many VM applications as possible in order to maximize their profits, VMB is an important problem to study. First we prove that VMB is NP-hard. Then we propose a suite of algorithms to solve VMB. In particular, we propose an approximation algorithm that achieves approximation ratio of $1/(2 \cdot \lceil \frac{B}{b} \rceil \cdot |E|^{1/(\lceil \frac{B}{b} \rceil + 1)} + 1)$, where $|E|$ is the number of edges in the data center network, B is the average bandwidth capacity on edges, and b is the average bandwidth demand of each request. We show through simulations that our algorithms are effective in accommodating large number of VM communications under different network parameters. In particular, our approximation algorithm accommodates more than 60% of total VM communications, and up to 38% more VM pairs compared to existing research.**

Keywords – Throughput Maximization, Virtual Machine Communication, Cloud Data Centers, Bandwidth Constraints, Approximation Algorithms

I. Introduction

Recent years have witnessed a dramatic growth of the cloud data centers. Consisting of hundreds of thousands of server machines, cloud data centers host rapidly-growing Internet applications including video streaming and social media [24], IoT-based applications such as ambient assisted living [12], and Massive Open Online Courses (MOOCs) [11]. A recent report of Cisco Global Cloud Index [2] shows that cloud data center workloads and compute instances nearly tripled (2.7-fold) for the last five years, which led to a massive growth in the volume of data and traffic in cloud data centers.

Meanwhile, as a mature virtualization technology, virtual machines (VMs) have become one of the key building blocks of modern cloud data centers. Despite that VMs are isolated units of CPU cycles, memory, and bandwidth, there are often cases where VM applications need to communicate across their isolation barriers. For example, a web service running in one VM may need to communicate with a database server running in another VM to satisfy the clients' transaction requests. A Google search engine VM might query a database VM to return search results to the cloud users. Such isolation unfortunately results in significant overheads (CPU, memory, and bandwidth) when different VM applications need to communicate with each other to achieve application objectives [8]. How to facilitate efficient inter-VM communications in cloud data centers becomes a challenging problem.

This problem is further exacerbated in *bandwidth-constrained data centers* [6], wherein bandwidth provision does not keep the pace with the growth of bandwidth demands of VM applications. Despite the continuous improvement of bandwidth provisions in cloud data centers, bandwidths are still scarce network resources for the following reasons.

First, cloud providers usually oversubscribe their data center resources (i.e., CPU, memory, storage, and bandwidth) to leverage its underutilized capacity in order to maximize their profits [7]. For example, some production datacenter networks such as Facebook data centers are oversubscribed as high as 40:1, causing the intra-datacenter traffic to contend for core bandwidth [20]. Second, the aforesaid VM applications are all data- and bandwidth-intensive, and many of them are complex combinations of multiple services that require predictable performances. With an explosive growth of such VM applications and their ensued network traffic, the demands for network resources such as bandwidths and switches' processing capabilities are rapidly growing. Consequently, such resources could be exhausted and become a performance bottleneck in a cloud environment. Although storage and processing power are commodities, bandwidth is not [3]. For example, while the cost of storage and computing has come down drastically over the years (with petabytes of storage and racks of servers of hundreds of cores), the bandwidth provision still lags behind and the bandwidth is still relatively expensive [6].

The high bandwidth demands from cloud users, combined with oversubscription of cloud data centers, lead to cloud bandwidth overloading. This consequently stresses the cloud network infrastructures and restrains VM communications in cloud environment. A cloud service provider thus must manage such bandwidth overloading while at the same time, aim to

978-1-7281-8104-2/21/$31.00 ©2021 IEEE

maximize its profit as well as minimize service level agreement (SLA) violations. When cloud infrastructures are stressed and cloud resources are insufficient to accommodate all the VM applications, service providers will accommodate as many VM applications as possible in order to maximize their profits.

We target this problem in this paper while focusing on VM communications *within* a cloud data center. In the data centers, the east-west traffic accounts for 75.4 percent of traffic and the internal traffic has increased much faster than Internet-facing traffic [2]. Furthermore, instead of focusing on the general all-pair VM communication paradigm that is commonly adopted by almost all of the research [22], [13], we focus on *pairwise* VM communication in this paper. Pairwise communication, wherein VMs communicate in pairs, is a prevalent communication paradigm in cloud computing platforms. For examples, in both cloud chatbots (e.g., Slack [4] and Amazon Lex [1]) and cloud messaging apps (e.g., WhatsApp and Facebook Messenger), one to one communication is still the most dominant one. Besides, such pairwise communicating VMs could have a diverse bandwidth demands ranging from low-bandwidth texts and voices to high-bandwidth live-streaming videos. Hence, it is important to provide bandwidth guarantees to such diverse pairwise applications in order to preserve the predictability of their response time.

In this paper we identify, formulate, and solve a new algorithmic problem called VMB. Given a bandwidth-constrained data center, and VM communicating pairs with varying bandwidth demands, the goal of VMB is to maximize the number of VM pair communications (from the perspective of the cloud service providers). We formulate the VMB as a graph-theoretical problem and prove its NP-hardness. We then propose a series of efficient bandwidth allocation algorithms that decide not only which VM pairs are to be accommodated but also the path along which each pair communicates. In particular, we prove that one of the algorithms is an approximation algorithm with approximation ratio of $1/(2 \cdot \lceil \frac{B}{b} \rceil \cdot |E|^{1/(\lceil \frac{B}{b} \rceil + 1)} + 1)$, where $|E|$ is the number of edges in the data center network, B is the average bandwidth capacity on edges, and b is the average bandwidth demand of each request. We show through simulations that our approximation algorithm accommodates more than 60% of total VM communications, and up to 38% more VM pairs compared to existing research under different network parameters. To the extent of our knowledge, maximizing the throughput of pairwise VM communication in a bandwidth-constrained data center has not been adequately tackled before. Our work is the first to deliver a constant-ratio approximation algorithm for this hard problem.

II. Related Work

How to accommodate VM communication in bandwidth-constrained data center has been an active research topic. Bodk et al. [6] observed that there exists an inherent tradeoff between achieving high fault-tolerance and reducing bandwidth usage. They created an optimization framework that achieves both high fault-tolerance and efficient bandwidth usage in a bandwidth-constrained data centers. In particular, they proposed minimum k-way cut [15] to partition the logical machines into a given number of clusters in order to minimize bandwidth consumption.

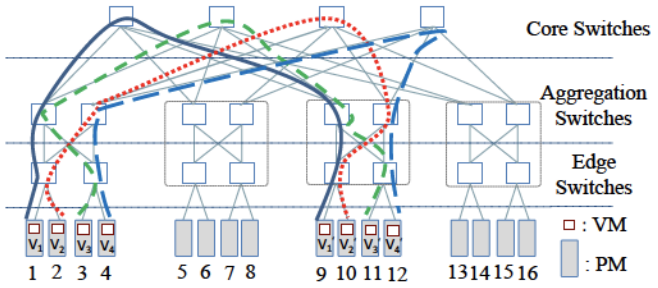


Fig. 1. A fat tree with $k = 4$ and 16 PMs. There are four communicating VM pairs: $(v_1, v_1^{'})$, ..., $(v_4, v_4^{'})$, each VM is at a different PM.

Lee et al. [19] presented a solution that provides bandwidth guarantees to cloud applications based on network abstraction and a workload placement algorithm. Duan et al. [10] studied load balancing in a multi-tenancy oriented data center considering the bandwidth constraint of servers. Yu et al. [28] studied how to embed virtual clusters survivability in the cloud data center and jointly optimized primary and backup embeddings of the virtual clusters with bandwidth-guarantee. Liu et al. [21] provided an in-network solution to achieve bandwidth guarantees and work conservation simultaneously.

All above work focuses on how to place the VMs or work loads inside cloud data centers to achieve bandwidth guarantee or resource efficiency. Our work instead assumes the VM pairs have already been placed inside the cloud data centers. This assumption is valid for the following reasons. Once the VMs and workloads are placed achieving various efficient resource utilization, it is possible such initial placement is no longer efficient due to dynamic resource demands of workloads in production data centers [25]. We thus need to consider that the VM applications and workloads are already placed in the cloud data centers and ask how to optimize their resource utilization dynamically. Our designed algorithm can indeed be executed periodically for throughput maximization of VM communications in response to dynamic user bandwidth demands.

There are a few works that allocate communicating VMs for efficient bandwidth usage [18], [14], [23]. Kumar [18] presented hierarchical bandwidth allocation infrastructure that supports service-level bandwidth allocation following prioritized bandwidth functions. Karmakar et al. [14] designed a few bandwidth allocation policies with the objective of maximizing throughput and bandwidth utilization while minimizing the service time and turnaround time. Nagaraj et al. [23] proposed a flexible and fast bandwidth allocation control that enables operators to specify how to allocate bandwidth among contending flows to optimize for different service-level objectives. Chen et al. [9] formulated an optimization problem that allocates bandwidth to maximize the social welfare across all the applications.

None of the work specifically maximizes the throughput (i.e., number of bandwidth requests) of a cloud data centers

targeted in this paper. More importantly, most of above research designed integer linear programming (ILP) solutions that run in exponential time and polynomial-time heuristics that do not have performance guarantees. We instead take a different approach by designing an approximation algorithm that is not only time-efficient but also provides performance guarantees to the service providers.

The most related work to ours is by Wang et al. [27]. They proposed efficient bandwidth allocation schemes to achieve energy efficiency in cloud data centers. In particular, they formulated a multi-commodity minimum cost flow problem, proved its NP-hardness, and proposed a heuristic solution for bandwidth allocation. As their problem set up is similar to ours, we compare with their work.

III. Problem Formulation of VMB

Network Model. We model a data center as an undirected general graph $G(V, E)$. Here $V = V_p \cup V_s$ is a set of physical machines (PMs) V_p and a set of switches V_s. E is the set of edges connecting either one switch to another switch or a switch to a PM. The bandwidth capacity of edge $e \in E$ is denoted as B_e. There are l communicating VM pairs $P = \{(v_1, v_1'), (v_2, v_2'), ..., (v_l, v_l')\}$ that are already created and placed on the PMs. All the $2l$ VMs are randomly placed on the PMs where VM v is placed at PM $S(v) \in V_p$. VM pair (v_i, v_i'), $1 \leq i \leq l$, demands b_i amount of bandwidth in order for v_i and v_i' to communicate with each other.

Fat-tree Data Centers. We focus on k-ary fat-trees [5], where k is the number of ports of each switch. Fat-tree topologies are widely adopted in data centers to interconnect commodity Ethernet switches. However, the problem formulation of VMB and its solutions are applicable to any topologies. There are three layers of switches in a fat-tree: edge switch, aggregation switch and core switch from bottom to top. In particular, there are k PODs (i.e., Points of Delivery) in a k-ary fat-tree. Each POD contains $k/2$ aggregation switches and $k/2$ edge switches. Each edge switch directly connects to $k/2$ PMs; and each of its remaining $k/2$ ports is connected to each of the $k/2$ aggregation switches from the same POD. In general, a k-ary fat-tree supports $\frac{k^3}{4}$ PMs. Fig. 1 shows a fat-tree with $k = 4$ and 16 PMs. As an example, there are four communicating VM pairs: (v_1, v_1'), ..., (v_4, v_4'), each VM is at a different PM.

Probability of Proximity (PoP) of VM Pairs. To facilitate the analysis of our VMB algorithms, we first introduce a new concept to quantify the closeness of two VMs in any VM pair in a k-ary data center. We define *probability of proximity (PoP)* as the probability of one VM that is 0, 2, 4, and 6 hops away from the other VM in the same pair assuming all VMs are all randomly placed on the PMs.

Fig. 2 shows the PoP of any VM pair w.r.t. k in a k-ary fat-tree. For example, the PoP for two VMs in a VM pair that are under different PODs (thus are 6 hops away) is $\frac{\frac{k^3}{4} - \frac{k^2}{4}}{\frac{k^3}{4}}$, which is $1 - \frac{1}{k}$. The PoPs for 0, 2, and 4 hops of the two VMs in the same pair are $\frac{4}{k^3}$, $\frac{2}{k^2} - \frac{4}{k^3}$, and $\frac{1}{k} - \frac{2}{k^2}$,

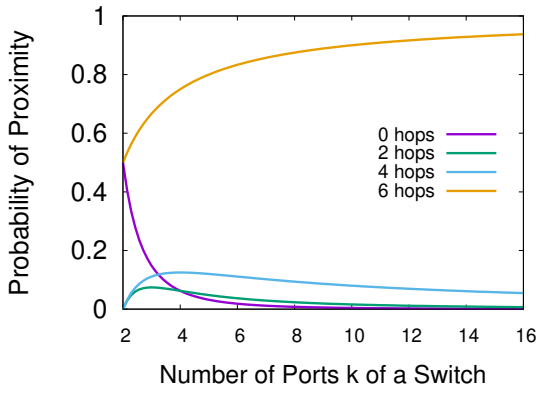


Fig. 2. The probability of proximity for VM pairs in a k-ary fat-tree.

respectively. An interesting observation from Fig. 2 is that with the increase of k, the PoP for any VM pair of being 6-hop away increases while the PoPs for other proximities decrease (with the exception of 2 and 4 hops, for which the PoPs first increase and then decrease).

Problem Formulation of VMB. We give below definition.

Definition 1: (**Feasible VM Pair Set.**) A set of VM pairs is feasible if and only if all the VM pairs in this set can successfully communicate with each other with the demanded bandwidth; that is, they are *satisfied*). Denote a feasible set of k VM pairs as $P_f = \{(v_{f1}, v_{f1}'), (v_{f2}, v_{f2}'), ..., (v_{fk}, v_{fk}')\} \subseteq P$, $1 \leq k \leq l$. Let binary variable x_{ei} indicate if the communication of (v_{fi}, v_{fi}'), $1 \leq i \leq k$, goes through edge $e \in E$: $x_{ei} = 1$ if yes and 0 otherwise. For any feasible set, it must be $\sum_{i=1}^{k}(x_{ek} \cdot b_{fi}) \leq B_e$ for any $e \in E$. □

The objective of the VMB is to find a feasible VM pair set with the largest number of VM pairs. More formally, let $\mathcal{F}$ denote the set of all feasible VM pair sets. The goal is to find a feasible VM pair set $P_f^m \in \mathcal{F}$ such as $|P_f^m| \geq |P_f|$ for all $P_f \in \mathcal{F}$ (here $|\cdot|$ means the cardinality of the set). We refer to $|P_f^m|$ as the *maximum throughput* of the VM communications.

EXAMPLE 1: In Fig. 1, there are four VM pairs: (v_1, v_1'), ..., (v_4, v_4'), each VM is at a different PM. Assume $B_e = 1$, $\forall e \in E$, and $b_i = 1$, $1 \leq i \leq l$, the maximum throughput is thus 4, as shown using the different dashed lines. In this case, all the four VM pairs can be accommodated without violating the bandwidth constraints on edges. However, if any of the VM pair has bandwidth demand of more than one, the maximum throughput will be less than four. □

Theorem 1: VMB is NP-hard.

Proof: We consider a spacial case of VMB wherein each edge has bandwidth capacity of one unit and each of the l VM pairs requests one unit of bandwidth (i.e., $b_i = B_e = 1$, $\forall e \in E$, $1 \leq i \leq l$), and denote it as VMB-1. To prove VMB is NP-hard, we show that VMB-1 is equivalent to maximum edge disjoint path problem (MEDP) [16], which is NP-hard.

MEDP is formally defined as below. Given an undirected graph $G(V, E)$, there are a set of k connection requests, each of which is specified by a pair of terminals s_i and t_i where $s_i, t_i \in V$. Let T be the set of all terminal pairs (s_1, t_1), ..., (s_k, t_k). T is realizable in G if there exists mutually edge

disjoint paths P_1, P_2, ..., P_k such that P_i has end points s_i and t_i. The goal of MEDP is to find a maximum realizable subset of a set T of terminal pairs in a graph G.

To show that MEDP is equivalent to VMB-1, let G' be the data center graph with k VM pairs, wherein VM pair $(v_i, v_i^{'})$ corresponds to the terminal pair (s_i, t_i) in G. As each edge has bandwidth capacity of one unit and each of the l VM pairs requests one unit of bandwidth, a feasible VM pair set of k VM pairs in VMB-1 and their routes must give k edge disjoint paths in MEDP, and vice versa. Therefore, there exists a VM throughput of k in G' if and only if there are k realizable terminal pairs in G. ∎

IV. Algorithmic Solutions for VMB

In this section we propose three algorithms viz. an approximation algorithm, a greedy algorithm, and a blocking island-based bandwidth allocation algorithm [27] to solve the VMB.

Approximation Algorithm. We first present our approximation algorithm. Algo. 1 below works as follows. It first assigns each edge in G a weight of 1. It then iteratively finds a shortest path of minimum cost (among all the paths) connecting a VM pair while satisfying the capacity of each edge. That is, if the accommodated VM pair is (v_k, v_k'), then each edge on this path must have remaining bandwidth of at least b_k. Then, it multiplies the weights of all the edges on this selected path by $\alpha = |E|^{\frac{1}{\lceil B/b \rceil + 1}}$, where B is the average bandwidth capacity of edges and b is the average bandwidth demand of VM pairs. Using Fibonacci heap, the shortest path computation takes $O(|E| + |V|\log|V|)$. There are at most l rounds, each round one VM pair is accommodated. In each round it finds among at most l VM pairs one minimum weighted route that can be satisfied. Therefore the time complexity of Algo. 1 is $O(l^2 \times (|E| + |V|\log|V|))$.

Algorithm 1: An Approximation Algorithm for VMB.
Input: A data center graph $G(V, E)$, l VM pairs P with demands b_i, $1 \le i \le l$, bandwidth capacity B_e of edge $e \in E$;
Output: a feasible VM pair set $\mathcal{F}$;
0. $\mathcal{F} = \emptyset$, $\alpha = |E|^{\frac{1}{B/b+1}}$;
1. For each $e \in E$, set its initial weight to 1;
2. **while** (there are still VM pairs that can be accommodated)
3. Find minimum weighted path P_i where adding P_i does not violate any edge's bandwidth capacity, and P_i connects VM pair (v_i, v_i') not yet connected;
4. $\mathcal{F} = \mathcal{F} \cup \{i\}$;
5. Use path P_i to route the message from $S(v_i)$ to $S(v_i')$;
6. Update available bandwidth of all edges in P_i;
7. Multiply the length of all edges in P_i by α;
8. **end while;**
9. **RETURN** $\mathcal{F}$.

The rationale of α is that when an edge is selected to accommodate the bandwidth request of some VM pair, we will increase its weight to discourage this edge from be used again immediately to route other VM pairs. This pricing method enables Algo. 1 to accommodate as many VM pairs as possible to achieve some constant-factor approximation ratio, as shown below.

Theorem 2: In a VMB instance, if all its edges have the same bandwidth capacity of B and all its VM pairs have the same bandwidth demand of b, then Algo. 1 achieves $1/(2 \cdot \lceil \frac{B}{b} \rceil \cdot |E|^{1/(\lceil \frac{B}{b} \rceil + 1)} + 1)$ approximation ratio. That is, the total number of satisfiable VM pairs by Algo. 1 is at least $1/(2 \cdot \lceil \frac{B}{b} \rceil \cdot |E|^{1/(\lceil \frac{B}{b} \rceil + 1)} + 1)$ times of the maximum number of satisfiable VM pairs in the optimal solution.

Proof: We first consider a special case of $\lceil \frac{B}{b} \rceil = 2$. Let F^* be the set of routing requests satisfied in the optimal solution and F be the set of requests satisfied by Algo. 1. A path P_i selected in Algo. 1 is *short* if its length is less than α^2.

Let F_s denote the set of short paths selected by Algo. 1. Let $\bar{F}$ be the length function at the first iteration in Algo. 1 where a long path is selected. For a path P_i^* in the optimal solution F^*, it is short if $\bar{F}(P_i^*) < \alpha^2$. As there are no short paths left when the length function reaches $\bar{F}$, it must be the case that path P_i^* has length at least α^2. We thus have the first observation OB (a): For a request $i \in F^*$ that is not satisfied by Algo. 1 (i.e., $i \in F^* - F$), $\bar{F}(P_i^*) \ge \alpha^2$.

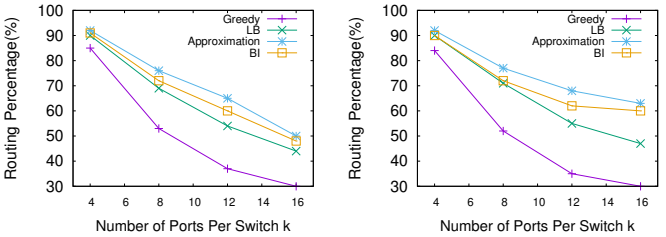
In the iteration where short paths are no longer available, the total length of the edges in the graph is $\sum_e \bar{F}(e)$. The sum of the edges in the graph starts out with $|E|$ (length 1 for each edge as indicated in Algo. 1). Adding a short path to the solution F_s can increase the length by at most α^3 as the selected path has length at most α^2, and the lengths of the edges are increased by an α factor along the path. We therefore have the second observation OB (b): $\sum_e \bar{F}(e) \le \alpha^3|F_s| + |E|$.

Consider OB (a) and all the paths in $F^* - F$, we get $\sum_{i \in F^* - F} \bar{F}(P_i^*) \ge \alpha^2|F^* - F|$. On the other hand, each edge is used by at most two paths in the solution F^*, so we have $\sum_{i \in F^* - F} \bar{F}(P_i^*) \le \sum_e 2\bar{F}(e)$. Combining these with OB (b), we get $\alpha^2|F^*| \le 2(\alpha^3|F| + |E|) + \alpha^2|F|$. Finally, we divide both sides by α^2, and considering that $|F| \ge 1$ and $\alpha = |E|^{1/3}$, we get $|F^*| \le (4|E|^{1/3} + 1)|F|$.

Above proof is for $\lceil \frac{B}{b} \rceil = 2$. For any value of $\lceil \frac{B}{b} \rceil$, if we choose $\alpha = |E|^{1/(\lceil \frac{B}{b} \rceil + 1)}$ and consider paths to be short if their lengths are at most $\alpha^{\lceil \frac{B}{b} \rceil}$, we get $|F| \ge |F^*|/(2 \cdot \lceil \frac{B}{b} \rceil \cdot |E|^{1/(\lceil \frac{B}{b} \rceil + 1)} + 1)$. ∎

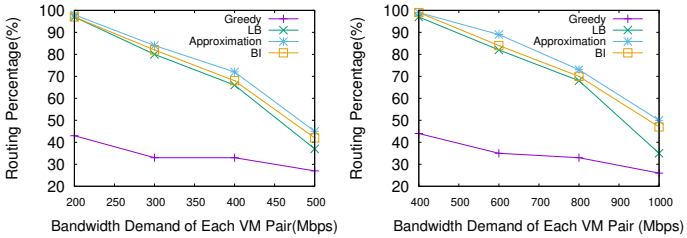
Above proof technique is inspired by Kleinberg et al. [17] solving the disjoint paths problem using pricing method, and is used in our previous work [26] solving a routing request maximization problem in static ad hoc networks.

Greedy Algorithm. We also present an efficient greedy algorithm to serve as the benchmark for the performance comparison. Algo. 2 works similar as Algo. 1 except that it does not consider the pricing method adopted in Algo. 1. In each round, it selects one VM pair that has not been satisfied whose connecting shortest path is the minimum; and meanwhile, each edge on this shortest path should have enough available bandwidth for the bandwidth demand of this VM pair. This continues until no more VM pairs can be



(a) $B_e = 10$Gbps, $b_i = 500$Mbps. (b) $B_e = 10$Gbps, b_i is random.

Fig. 3. Varying number of ports k of switches.



(a) $k = 12$, constant bandwidth. (b) $k = 12$, random bandwidth.

Fig. 4. Varying bandwidth demands b_i of VM pairs.

accommodated due to insufficient bandwidth in the data center. Its time complexity is again $O(l^2 \times (|E| + |V|\log|V|))$.

Algorithm 2: A Greedy Algorithm for VMB.
Input: A data center graph $G(V, E)$, l VM pairs P with demands b_i, bandwidth capacity B_e of edge $e \in E$;
Output: a feasible VM pair set $\mathcal{F}$;
0. $\mathcal{F} = \emptyset$;
1. For all $e \in E$, set its weight to 1;
2. **while** (there are still VM pairs that can be accommodated)
3. Find minimum weighted path P_i where adding P_i does not violate any edge's bandwidth capacity, and P_i connects some VM pair (v_i, v_i') not yet connected;
4. $\mathcal{F} = \mathcal{F} \cup \{i\}$;
5. Use path P_i to route the message from $S(v_i)$ to $S(v_i')$;
6. Update available bandwidth of all edges in P_i;
7. **end while;**
8. **RETURN** $\mathcal{F}$.

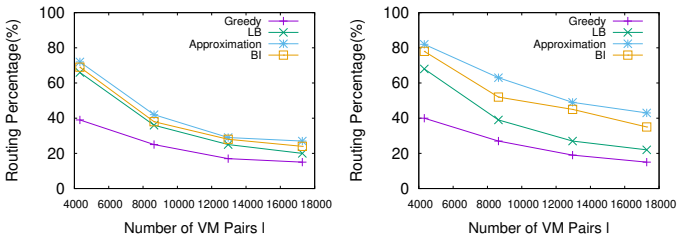
Blocking Island Bandwidth Allocation Algorithm [27]. Wang et al. [27] proposed a bandwidth heuristic allocation scheme that achieves proportional energy efficiency in data centers. The key idea of their algorithm is to abstract the original network graph into a tree containing available bandwidth information and then apply the blocking island-based method to decide which traffic demand should be allocated with its demanded bandwidth. To achieve a higher success ratio of bandwidth allocation and higher computation efficiency, it selects the unallocated traffic demands as below. First, it sorts all the VM pairs by maximum minimum-available bandwidth edge in descending order. If there are ties, it chooses the one with shorter path first. If there are still ties, it chooses the one with highest bandwidth demand first. Then, it places each VM pair along its shortest path, breaking ties randomly.

V. Performance Evaluation

Simulation Setting. In this section, we compare the performances of our algorithms viz. approximation algorithm Algo. 1 (referred to as **Approximation**) and greedy algorithm Algo. 2 (referred to as **Greedy**) with blocking island-based algorithm [27] (referred to as **BI**). We also present a variation of the Approximation that focuses on the load-balancing of the edges, and refer to it as **LB**. In particular, when there are multiple edges available, LB selects the one whose remaining bandwidth is the maximum. We create data centers of different sizes, wherein each edge has a bandwidth capacity. The source and destination VMs of each VM pair are randomly placed on the PMs and each VM pair has a bandwidth demand for its communication.

Unless otherwise mentioned, the bandwidth capacity on each edge of the cloud data center is set as 10Gbps. In all the simulation plots, each data point is an average of 10 runs and the error bars indicate 95% of confidence interval. For fair comparison, we run different algorithms on the same cloud data center instance with the same initial placement of VM communication pairs. Finally, as the number of accommodated VM pairs depends on the total number of VM pairs l, to be consistent in all the cases, instead of presenting number of satisfied VM pairs as the throughput, we normalized it as the ratio between the number of satisfied VM pairs and l, and refer to it as *routing percentage*.

Effects of Number of Ports k of Switches. We first investigate the effects of k on the throughput of VM communications, shown in Fig. 3. We increase k from 4, 8, 12, to 16, with number of PMs varying from 16 to 1024. We set he number of VM pairs $l = \frac{10 \cdot k^3}{4}$; i.e., 10 VM pairs are randomly placed on each PM. We have several observations. First, with the increase of k, the throughput of all the algorithms decreases. This is consistent with our previous analytical analysis shown in Fig. 2 that when k is large, it's more likely two VMs in the same pair are six hops away from each other, costing more bandwidth compared to when k is small. Second, we observe that Approximation outperforms the BI, which outperforms the LB and Greedy in terms of throughput of VM communications. In particular, the Approximation produces up to 38% more throughput of the VM communications than the BI.



(a) $k = 12$, $b_i = 400$Mbps. (b) $k = 12$, b_i is random.

Fig. 5. Varying number of VM pairs l.

Effects of Bandwidth Demands b_i. We then investigate the effects of bandwidth demands b_i of VM pairs on the VM throughput, shown in Fig. 4. We set k as 12, which means there are $l = 10 \cdot k^3/4 = 4320$ VM pairs. Fig. 4(a) assumes that each VM pair has the same uniform bandwidth demands as specified and in Fig. 4(b), the bandwidth demands of VM pairs are random numbers between 0 and the specified bandwidth. Again we observe the same trend that Approximation performs the best among the four algorithms. In particular, the Approximation produces up to 35% more throughput of the VM communications than BI does.

Effects of Number of VM pairs l. Finally we vary the number of VM pairs l in Fig. 5 and investigate the performance of the four algorithms in data center of $k = 12$. In particular, we change l from $10 \cdot k^3/4 = 4320$ to $40 \cdot k^3/4 = 17280$. Fig. 5(a) fixes each VM pair's bandwidth demand as 400Mbps while in Fig. 5(b), the bandwidth demand of each VM pair is a random number in [0, 800Mbps]. In these stressful scenarios, again we observe Approximation performs the best by yielding more than 60% of VM communications, showing that it is an effective bandwidth allocation scheme.

VI. Conclusion and Future Work

We proposed a new algorithmic problem to maximize the throughput of VM communication in bandwidth-constrained data centers. Given a set of VM pairs inside cloud data centers, each with a particular bandwidth demand, it studies how to allocate the cloud network bandwidth to the VM pairs to accommodate maximum number of VM communication (i.e., maximizes the throughput) while considering that cloud data centers have limited bandwidths. We proved its NP-hardness and proposed a suite of algorithms. One of our algorithms is an approximation algorithm that achieves approximation ratio of $1/(2 \cdot \lceil \frac{B}{b} \rceil \cdot |E|^{1/(\lceil \frac{B}{b} \rceil + 1)} + 1)$, where $|E|$ is the number of edges in the data center network, B is the average bandwidth capacity on edges, and b is the average bandwidth demand of each request. We compared it with existing approach and showed that our algorithm accommodate up to 38% more VM pairs compared to existing approach under different network parameters. In our current setup, we assume that each VM pair has the same priority thus our goal is to maximize the number of VM pairs to be satisfied. As a future work, we will consider that different VM pairs not only have different bandwidth demands but also different priorities (i.e., weights). How to maximize the total weight of all the accommodated bandwidth requests become a new challenging problem.

Acknowledgment

This work was supported by NSF Grant CNS-1911191.

References

[1] Amazon lex. https://aws.amazon.com/lex/.

[2] Cisco global cloud index: Forecast and methodology, 2016 to 2021 white paper. https://www.cisco.com/c/en/us/solutions/service-provider/global-cloud-index-gci/white-paper-listing.html.

[3] Data center bandwidth and measurements. https://www.colocationamerica.com/data-center-connectivity/bandwidth.

[4] Slack, the collaboration software that moves work forward. http://slack.com.

[5] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. *SIGCOMM Comput. Commun. Rev.*, 38(4):63–74, 2008.

[6] P. Bodík, I. Menache, M. Chowdhury, P. Mani, D. A. Maltz, and I. Stoica. Surviving failures in bandwidth-constrained datacenters. In *Proc. of ACM SIGCOMM 2012*.

[7] D. Breitgand and A. Epstein. Improving consolidation of virtual machines with risk-aware bandwidth oversubscription in compute clouds. In *Proc. of INFOCOM 2012*.

[8] A. Burtsev, K. Srinivasan, Prashanth Radhakrishnan, Kaladhar Voruganti, and Garth R. Goodson. Fido: Fast inter-virtual-machine communication for enterprise appliances. In *USENIX Annual Technical Conference*, 2009.

[9] Li Chen, Yuan Feng, Baochun Li, and Bo Li. Efficient performance-centric bandwidth allocation with fairness tradeoff. *IEEE Transactions on Parallel and Distributed Systems*, 29(8):1693–1706, 2018.

[10] J. Duan and Y. Yang. A load balancing and multi-tenancy oriented data center virtualization framework. *IEEE Transactions on Parallel and Distributed Systems*, 28(8):2131–2144, 2017.

[11] J. A. Gonzalez-Martinez, M. L. Bote-Lorenzo, E. Gomez-Sanchez, and R. Cano-Parra. Cloud computing and education: A state-of-the-art survey. *Computers & Education*, 80:132 – 151, 2015.

[12] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak. The internet of things for health care: A comprehensive survey. *IEEE Access*, 3:678–708, 2015.

[13] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang. Joint vm placement and routing for data center traffic engineering. In *Proc. of IEEE INFOCOM 2012*.

[14] K. Karmakar, R.K. Das, and S. Khatua. Bandwidth allocation for communicating virtual machines in cloud data centers. *J Supercomput*, 76:7268–7289, 2020.

[15] K. Kawarabayashi and M. Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In *Proc. of IEEE FOCS 2011*.

[16] J. Kleinberg. Approximation algorithms for disjoint paths problems, 1996.

[17] J. Kleinberg and E. Tardos. *Algorithm Design*. Pearson Education, Inc, USA, 2005.

[18] A. Kumar, S. Jain, U. Naik, A. Raghuraman, N. Kasinadhuni, E.C. Zermeno, C.S. Gunn, J. Ai, B. Carlin, and M. Amarandei-Stavila. Bwe: flexible, hierarchical bandwidth allocation for wan distributed computing. *ACM SIGCOMM Comput Commun Rev*, 45(4):1–14, 2015.

[19] J. Lee, Y. Turner, M. Lee, L. Popa, S. Banerjee, J. Kang, and P. Sharma. Application-driven bandwidth guarantees in datacenters. *SIGCOMM Comput. Commun. Rev.*, 44(4):467478, August 2014.

[20] J. Lee, Y. Turner, M. Lee, L. Popa, S. Banerjee, J.M. Kang, and P. Sharma. Application-driven bandwidth guarantees in datacenters. In *SIGCOMM '14*.

[21] Z. Liu, K. Chen, H. Wu, S. Hu, Y. Hut, Y. Wang, and G. Zhang. Enabling work-conserving bandwidth guarantees for multi-tenant datacenters via dynamic tenant-queue binding. In *Proc. IEEE INFOCOM 2018*.

[22] X. Meng, V. Pappas, and L. Zhang. Improving the scalability of data center networks with traffic-aware virtual machine placement. In *Proc. of IEEE INFOCOM 2010*.

[23] K. Nagaraj, D. Bharadia, H. Mao, S. Chinchali, M. Alizadeh, and S. Katti. Numfabric: fast and flexible bandwidth allocation in datacenters. In *Proc. of ACM SIGCOMM 2016*.

[24] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian. Mobile cloud computing: A survey, state of art and future directions. *Mob. Netw. Appl.*, 19(2):133–143, 2014.

[25] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren. Inside the social networks (datacenter) network. In *Proc. of SIGCOMM 2015*.

[26] Z. Sumpter, L. Burson, B. Tang, and X. Chen. Maximizing number of satisfiable routing requests in static ad hoc networks. In *Proc. of the IEEE Global Communications Conference (GLOBECOM 2013)*.

[27] T. Wang, B. Qin, and Z. Su. Towards bandwidth guaranteed energy efficient data center networking. *J Cloud Comp*, 4(9), 2015.

[28] R. Yu, G. Xue, X. Zhang, and D. Li. Survivable and bandwidth-guaranteed embedding of virtual clusters in cloud data centers. In *Proc. of IEEE INFOCOM 2017*.