Utility-Driven Bandwidth Allocation in Data Center Networks

Hongbo Wang[1], *Yangyang Li*[2], *Shiduan Cheng*[1]
[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China*
[2]*Innovation Center, China Academy of Electronics and Information Technology, China*
hbwang@bupt.edu.cn, yli@csdslab.net, chsd@bupt.edu.cn

Abstract

It has been realized that network performance is one of the most important metrics for evaluating the performance of applications deployed in cloud data centers. Numerous bandwidth allocation schemes have been proposed to maximize the utilization of data center networks as well as to improve the performance of applications. However, those works only focus on allocating link bandwidth shared among elastic applications; the path delay experienced by inelastic traffic such as delay-sensitive traffic has been neglected. In this paper, we investigate the problem of maximizing application utility in data center networks considering both the throughput and the delay influence. The utility of an application is the benefit brought by bandwidth increases minus the expenditure charged by congestion delay growth. We then formulate the utility-driven bandwidth allocation problem as a convex optimization with the objective of maximizing overall utility across all applications. Standard interior point algorithm is applied to derive the optimal solution. We show the outstanding performance of our solution through extensive simulations with several realistic data center network (DCN) topologies.

Keywords: Cloud computing, Utility-driven, Bandwidth allocation, Data center network.

1 Introduction

With the proliferation of cloud computing, many enterprises have adopted data centers as the standard computing platform to run their applications. For example, on-demand streaming provider Netflix [1] has moved their services to Amazon EC2 [2] to take the advantage of computation and storage resources which are available on demand. Data centers provide many mechanisms [3-5] to schedule the computation, memory and disk resources to achieve cost efficiencies and on-demand scaling. However, existing schemes fall short to provide predictable network performance of applications.

Measurement and analysis [6-7] indicate that network performance has become a key concern for tenants evaluating the performance of their applications. The authors of Ref. [8] conducted several experiments in both public data centers and production data centers to evaluate the network performance for tenants in shared environment. They showed that network performance varies significantly among different applications, because these applications are coexisting in the same data center and they are untrusted by each other. The competition for the use of network resources makes the utilization of data center networks very poor; in the meantime, the network interference between tenants makes the network performance of applications unpredictable. Therefore, it is critical to provide strong network performance isolation in data centers.

Many researchers [8-13] have appealed to bandwidth allocation for providing predictable network performance and network isolation among tenants. On the one hand, they provide minimum bandwidth guarantees [8, 12] by reserving fixed bandwidth for each application. The drawback is, even if the reserved bandwidth is not fully used by one application, it could not be used by other applications any more. On the other hand, they offer fair bandwidth allocation [9-11, 13] by allocating bandwidth according to the weight of each tenants, applications, or virtual machines. Those weight-based methods do not have minimum bandwidth guarantees if there are too many applications coexisting in the same data center. What's more, all these mechanisms only focus on link bandwidth. Network latency has rarely been considered as the primary concern of network performance.

In this paper, we argue that the network performance of applications is not only influenced by the throughput obtained from bandwidth allocation, but also influenced by the network latency acquired from the path selection. For example, the throughput-oriented applications, such as file transfer and map-reduce like applications, they prefer to request for large bandwidth to reduce the completion time of the tasks. But the latency-sensitive applications, such as multiple tier web services and financial transactions, they prefer to use a shorter path. As in-between applications such as video on-demand, they need both high throughput and low latency. Therefore, we are motivated to propose a utility-driven bandwidth allocation scheme in data center networks. We build utility function for each application based on their sensitivities, and the objective of our bandwidth allocation is to maximize the overall utility among all applications.

The primary contributions of this paper are summarized as follows:

*Corresponding author: Hongbo Wang; E-mail: hbwang@bupt.edu.cn
DOI: 10.6138/JIT.2017.18.3.20130430

- We propose a novel bandwidth allocation solution for the network-sharing problem in data center networks. The gist of our solution is, we penalize the traffic use paths with high delay while encouraging traffic to use less popular shortest path to avoid congestion.
- We formulate the bandwidth allocation problem as a utility maximization problem. The utility of an application is the benefit brought by bandwidth increases minus the expenditure charged by congestion delay growth. The overall utility in a data center network is defined as the linear weighted sum of all applications.
- We prove that the formulation of our utility-driven bandwidth allocation problem is a convex optimization problem. Standard interior point algorithm is used to derive the optimal solution. Simulation results show that our solution outperforms current bandwidth allocation mechanisms.

2 Related Work

Most of the previous researches focus on providing bandwidth allocation mechanism to meet the network bandwidth demands of tenants. These mechanisms mainly include bandwidth reservation and weighted bandwidth allocation. Until recently, there has been few works considering network latency demands of tenants.

SecondNet [12] offers three priority bandwidth guarantee, including type 0, type 1 and best effort. Type 0 provides fixed reserved bandwidth between virtual machines, which is analogous to Integrated Service [14]. Type 1 provides only last and/or first hop guarantee, and best effort type does not have any guarantee. The authors of Oktopus [8] proposed two class virtual network abstractions to meet different application requirements. They distinguished data-intensive applications from others so that "Virtual Cluster" and "Virtual Oversubscribed Cluster" are abstracted respectively. Both of the two-class abstraction can guarantee fixed switch-to-VM bandwidth. The only difference is that the latter is interconnected with an oversubscription factor. However, even for the highest priority application, i.e., priority 1 in [12] and "Virtual Cluster" in [8], the bandwidth each virtual machine obtained is bounded by a fixed value. Applications cannot benefit from spare network resource in data centers. What's more, if the fixed bandwidth were oversubscribed, tenants would lose money that they invested for applications and spare network resources are wasted in the meanwhile.

NetShare [11] and Seawall [13] allocate bandwidth according to the weights of applications in centralized and distributed ways respectively. The pure weight-based approaches cannot guarantee predictable performance under the worst cases. Suppose a scene where a great number of applications compete for one congestion link. In this situation, even the application with the largest weight cannot acquire a minimum bandwidth guarantee. Terry et al. bridged this gap in an extended version [10] of NetShare [11] via introducing an admission control strategy. Nevertheless, we argue that the weight-based methods cannot meet the network latency requirements of tenants. In [9], the authors proposed a network resources sharing scheme that allocate bandwidth in proportion to the payments of tenants, the nature of the sharing is still based on weight, where the payment is equivalent to weight.

Some researchers have begun to guarantee heterogeneous or uncertain bandwidth demands in multitenant data center networks. Li et al. [15] proposed a heterogeneous bandwidth demand guarantee method, where the tenant can specify a diverse set of bandwidth demands for their virtual machines. In [16], the authors proposed a novel virtual cluster abstraction, where the bandwidth requirements between virtual machines can be stochastic. In order to capture the time-varying bandwidth requirements of cloud applications, temporally-interleaved virtual clusters TIVC [17] was introduced to reduce over-reservation of network bandwidth in fixed-bandwidth abstractions. Kraken [18] allows tenants to dynamically request and update minimum network bandwidth demands, it can be achieved through an online resource reservation scheme.

However, none of the bandwidth guarantee mechanisms above has considered network latency as a primary metric for evaluating the performance of applications. The authors of HULL [19] considers the latency in the network switching nodes, they cap the amount of bandwidth available on a link in exchange for significant reduction in latency. Parley [20] provides service-centric minimum bandwidth guarantee, which can help services maintain low tail latency. In [21], the authors explored several choices for a cloud provider to infer network latency demands of the tenants.

To the best of our knowledge, this paper is one of the first researches to consider both the bandwidth and the latency features to evaluate the network performance of applications running in data center networks. We allocate bandwidth to each application on multiple paths as long as there is available bandwidth on that path. Therefore, our scheme has higher bandwidth utilization than approaches proposed in [8, 12]. We construct utility functions to reflect the throughput-sensitive and delay-sensitive features of different applications; hence we believe that bandwidth allocation according to the characteristics of applications can get more fine-grained service differentiation than [9-11, 13].

In [22], we proposed a similar model to allocate

bandwidth in data center networks. However, the mathematical formulation in that paper is hard to resolve. And the simulation in that work is weak. We revised and extended that work to give an elaborate utility-based bandwidth allocation scheme in detail.

3 Overview

In data center networks, central controller is commonly adopted to allocate IP addresses to virtual machines as well as to conduct some other operations. Taking advantage of this controller, we design the bandwidth allocation mechanism in a centralized way and then we briefly describe the architecture of our scheme in this section.

For convenience, we take one of the well-known DCN topology fat-tree [23] as an example to present our scheme. It should be noted that the deployment and implementation of our scheme is independent of the DCN topologies. As shown in Figure 1, fat-tree is split into three layers, which are labeled edge, aggregation and core respectively. There are k Pods, each Pod contains two layers of $k/2$ switches. Each k-port switch in the edge layer is directly connected to $k/2$ servers. The remaining $k/2$ ports are connected to $k/2$ aggregation layer switches. There are $(k/2)^2$ k-port core layer switches. Each core switch has one port connected to one of the aggregation layer switches in each Pod. In general, a fat-tree that is built with k-port switches can support $k^3/4$ servers. In Figure 1, $k = 4$, so it can support at most 16 servers.

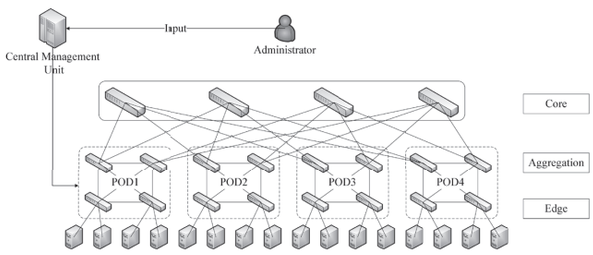


Figure 1 Fat Tree Topology

As shown in Figure 1, the central management unit is the key component of our mechanism. The main function of central management unit is to maintain routing matrix and compute the bandwidth needed for allocation according to the requirements of applications that are input by the administrator of data centers. Application requirements should at least contain the following three parameters: throughout-sensitive coefficient, delay-sensitive coefficient and minimum bandwidth requirement. The specific meaning of these parameters will be introduced in the next section.

In order to generate the routing matrix, we need some symbols to differentiate each physical links. We label all physical links in the topology using a similar method that Radhika et al. [23] used to allocate IP addresses to switches. In brief, starting from Pod 1, we mark the links among edge switches and aggregation switches with number 1 – $k/2$ (from left to right). Then the links between aggregation layer and core layer can be labeled in the same way. The remaining links in other Pods could be labeled continually. In Section 6, we will give an example of labeled topologies.

In virtualized cloud data centers, each packet is processed by a virtual switch before the packet is sent out or forwarded to virtual machines. Virtual switch is usually implemented in the virtualization layer of physical servers. As a software switch, many specific functions can be developed according to our demands. As long as the bandwidth allocation results are received from central management unit, the virtual switch will be triggered to allocate bandwidth on each corresponding path. The major technologies that could be applied here are rate limiting and multi-path routing. Discussion in further detail can be found in Section 5.

4 Utility-Driven Bandwidth Allocation

4.1 System Model

We model the DCN topology as a weighted undirected graph and denote it by $G = (N, L)$, where N is the set of switches and L is the set of physical links. L is denoted by $L = 1, 2, ..., l (l \geq 2)$. The bandwidth capacity and remaining bandwidth capacity of links are denoted by vector $C = (c_1, c_2, ..., c_l)(l \geq 2)$, $\gamma = (\gamma_1, \gamma_2, ..., \gamma_l)(l \geq 2)$ respectively. A tenant usually rents a group of virtual machines to run their applications. We use VM-VM pair [*srcVM*, *dstVM*] to represent the communication between virtual machines. Data center networks are commonly constructed with multi-root tree topology [23-25]. Virtual machines can communicate with each other using multiple paths. For example, in fat-tree topology, the number of paths for inter-Pod and intra-Pod VM communication is determined by the number of core switches and the number of aggregation switches in each Pod respectively. We use p_i to represent the number of paths that are available for VM-VM pair i. Then the bandwidth allocated to the i_{th} VM-VM pair can be denoted by vector $X_i = (X_{i1}, X_{i2}, ..., X_n)(i \geq 2)$, x_{ij} denotes bandwidth allocated to pair i on path j. For convenience, we assume that all applications running in the data center use n VM-VM-pairs, the global bandwidth allocation vector can be acquired with $X = (X_1, X_2, ..., X_n)(n \geq 2)$. Accordingly, routing matrix can be denoted by

$$R_{l,p} = \begin{pmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,p} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ R_{l,1} & R_{l,2} & \cdots & R_{l,p} \end{pmatrix}$$

Where

$$p = \sum_{i=1}^{n} p_i$$

The number of rows means that there are l physical links in data center network. And p columns mean there are n VM-VM pairs each with p_i redundant links. The value of elements in matrix can be determined by using the following indicative function:

$$R_{i,j} = \begin{cases} 1, & \text{if link } i \in \text{path } j \\ 0, & \text{if link } i \notin \text{path } j \end{cases}$$

4.2 Service Model

For fine-grained differentiated bandwidth guarantees, we propose an utility-based bandwidth guarantee interface whereby tenants can specify application performance requirements according to the characteristics of applications. The interface can be denoted by using a set of rules of the format: [*ApplicationID*, *srcVM*, *dstVM*, B_{min}, α, β]. Wherein the interface, B_{min} denotes the minimum bandwidth requirement of the application, α and β mean the throughput-sensitive and delay-sensitive coefficient of the application respectively. For example, [app_0, vm_0, vm_1, 10, α, β] specifies that communication between vm_0 and vm_1 in app_0 requires at least 10 units bandwidth guarantee with utility coefficient [α, β].

As Amazon Elastic Compute Cloud [3] specifies tens of instances for different requirement combinations of CPU, Memory, I/O resources. We differentiate network resource requirements by setting the combination of [α, β] for different applications. The number of combinations can be set according to the need of Cloud provider flexibly. For example, if we want to specify up to 100 instances, we can define α, β ∈ $\{x|1 \le x \le 10, x \in Z\}$, then the combination of [α, β] can express bandwidth differentiation for 100 different application types.

Based on the interface we defined above, we can construct utility function to show the network performance for different types of applications:

$$U_k = \sum_{u:u\in pair(k)} \sum_{v:v\in pair(u)} \sum_{w:w\in link(v)} (\alpha_k x_{kw} - \frac{\beta_k x_{kw}}{\gamma_w}) \qquad (1)$$

pair (k) denotes the set of VM-VM pairs which belong to application k. *path* (u) denotes the set of paths used by VM-VM pair u. And *link* (v) denotes the set of links used by path v. x_{kw} denotes bandwidth allocated to application k on link w, which can be obtained using bandwidth allocation vector X and routing matrix R. The term $1/\gamma_w$ denotes the expected congestion delay on link w from an $M/M/1$ delay function [26-28], where γ_w denotes the residual bandwidth capacity on physical link w. α_k and β_k reflect the throughput and delay sensitive characteristic of application k respectively. Hence the utility of application k is consisted of the utility that the application obtained from all links, paths and VM-VM pairs it uses.

The meaning behind the utility is a tradeoff between income brought by bandwidth increases and expenditure charged by congestion delay growth. For a throughout-sensitive application e.g., MapReduce application, the coefficient α is always set larger to reflect that bandwidth affects its utility obviously; for delay-sensitive applications e.g., user-facing Web applications, β is usually set larger to reflect delay affects their performance significantly. Specific benchmarks can be conducted to assist setting the values of [α, β].

To be mentioned, we provide an interface to tenants where the tenants can submit their network performance requirements by specifying B_{min}, α, β. What behind the interface is that, in order to get higher quality of service, the tenant always appear for larger α and/or β which means a higher payment. From the perspective of cloud providers, it is necessary to meet the requirements of tenants who pay more first. In consideration of this, we can naturally denote the weight to each application by calculating the throughput-sensitive and delay-sensitive coefficient

$$\lambda_k = \sqrt{\alpha_k^2 + \beta_k^2} \qquad (2)$$

λ_k being small implies that both coefficients are small and being large means that at least one coefficient is large.

In Table 1, the key notations that are used throughout the paper are summarized.

Table 1 Key Notations in the System and Service Model

Symbol	Description
l	Number of physical links
n	Number of VM-VM pairs
c_i	Bandwidth capacity of physical link i
γ_i	Residual bandwidth of physical link i
X_i	Bandwidth allocated to VM-VM pair i
B^k_{min}	Minimum bandwidth requirement of application k
α_k	Throughout-sensitive coefficient of application k
β_k	Delay-sensitive coefficient of application k
U_k	Utility function of application k
λ_k	Generalized weight of application k

4.3 Formulation

The objective of our utility-driven bandwidth allocation problem is to find an optimal solution that can maximize the weighted sum of overall network utility of all applications. Hence the problem can be formulated as a mathematical optimization problem. The mathematical model is shown as following:

$$Maximize \sum_{k=1}^{K} \lambda_k U_k \tag{3}$$

$$Subject\ to\ R_{l,p} \times X_{1,p}^{T} \prec (c_1, c_2, ..., c_l) \tag{4}$$

$$\sum_{j=1}^{p_i} x_{ij} \geq B_{min}^{k},\ \forall i, k, i \in pair(k) \tag{5}$$

$$0 \leq x_{ij} \leq c_l,\ \forall i, j \tag{6}$$

Inequality (4) shows the bandwidth allocation is subject to the constraint of the bandwidth capacity of physical links. Inequality (5) guarantees that the bandwidth allocated to each VM-VM pair meet the minimum requirement of the application that the pair belongs to. Inequality (6) is the boundary of bandwidth that can be allocated.

5 Solution

5.1 Transforming in a Convex Problem

In the aforementioned formulation, we use the *M*/*M*/1 queuing formula to denote the queuing delay which is experienced by each application. However, when the physical link is over utilized, in other words, $\gamma_w > 0$, the delay function will not be defined. By the way, it is very time expensive to calculate the unique solution of the problem. To overcome this problem, we resort to mathematical approximation of the *M*/*M*/1 formula.

In order to obtain the unique solution of our problem, we need to make some changes to let the formulation of our problem to be convex.

Changing the objective from *maximize* $-\sum_{k=1}^{K} \lambda_k U_k$ to *minimize* $-\sum_{k=1}^{K} \lambda_k U_k$, we can transform the previous optimization problem (3) to the following problem:

$$Minimize -\sum_{k=1}^{K} \lambda_k U_k \tag{7}$$

Subject to (4), (5), (6)

For mathematical convenience and quick convergence, similar to [29-30], we use a piecewise linear approximation function $f(u_w)$ instead of the formula $1/\gamma_w$. In the approximation function, u_w is the utilization of link w, which is the ratio of allocated bandwidth on link w to the capacity of link w. We define $f(u_w)$ as a continuous function with $f(0) = 0$ and use a derivative in the utilization u_w of

$$f'(u_w) = \begin{cases} 0.1 & 0 \leq u_w < 0.2 \\ 0.2 & 0.2 \leq u_w < 0.4 \\ 0.4 & 0.4 \leq u_w < 0.6 \\ 0.8 & 0.6 \leq u_w < 0.8 \\ 1.6 & 0.8 \leq u_w < 1 \\ 100 & 1 \leq u_w \end{cases} \tag{8}$$

In a word, the delay function is changed from $\frac{\beta_k x_{kw}}{\gamma_w}$ to $\beta_k x_{kw} f(u_w)$. Since $f(u_w)$ is non-decreasing, twice differentiable and convex, $\beta_k x_{kw} f(u_w)$ is convex. And $-\alpha_k x_{kw}$ is linear and convex, so $-U_k$ is convex. Because the non-negative weighted sum of convex functions is convex, $-\sum_{k=1}^{K} \lambda_k U_k$ is convex. Moreover, all of the constraints are affine or linear. Therefore, the optimization problem (7) is a convex problem [31].

Standard convex optimization solvers could be used to solve this problem. In this paper, we will use interior point algorithm, which is integrated in the Matlab optimization toolbox to solve our utility driven bandwidth allocation problem.

5.2 Implementation Discussion

Our mechanism provides utility driven bandwidth guarantee for different types of cloud applications by implementing bandwidth allocation in the virtualization layer of physical servers. As long as the virtual switch deployed in the virtualization layer receives the optimal solution from our central management unit, it will automatically set a bandwidth capping to each VM-VM pair with

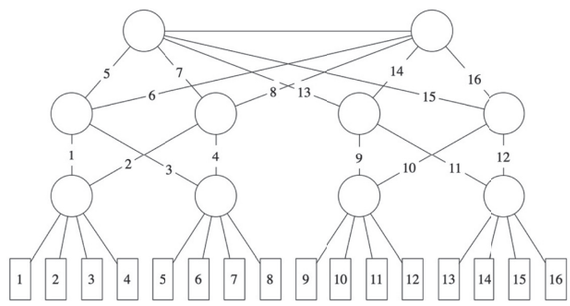
$$up_i = \sum_{j=1}^{p_i} x_{ij} \tag{9}$$

Another thing the virtual switch needs to do is to allocate the bandwidth x_{ij} to each corresponding path *j*. We could make a simple modification on current commonly used Equal Cost Multiple Path (ECMP) routing protocol in data centers. By adding a weight to each path, we can distribute packets to each path according to $X_i = (X_{i1}, X_{i2}, ..., X_{ip_i})$. The weight can be calculated by

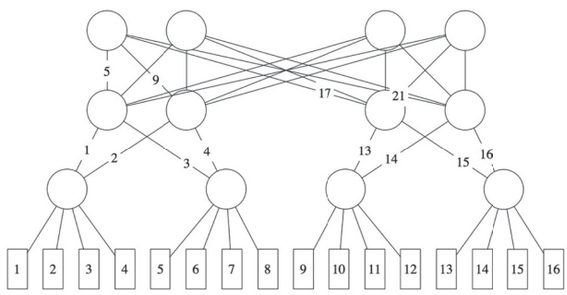
$$w_{ij} = \frac{x_{ij}}{\sum_{k=1}^{p_i} x_{ik}} \tag{10}$$

6 Simulations

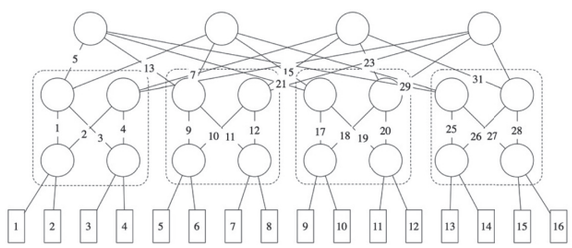
We conduct extensive simulations to show that the proposed utility driven bandwidth allocation mechanism outperforms bandwidth-based algorithm and delay-based algorithm. The bandwidth-based allocation algorithm can be seen as weight-based bandwidth allocation algorithm, which is commonly used in recently proposed data center network bandwidth allocation mechanisms [9-11, 13]. And the delay-based bandwidth allocation algorithm was proposed by Javed et al. [29], which is used to minimize the end-to-end delay experienced by inelastic traffic in Internet. All experiments are performed on a server with 16G memory and 3.3GHz Six-Core Processor using Matlab optimization toolbox. The algorithm parameter is set to be "Interior point." We use three typical structures: Tree [24], VL2 [25], and Fat-Tree [23], which represent data center networks of different architectures. Small scale instances of the three topologies are shown in Figure 2.



(a) Tree



(b) VL2



(c) Fat-Tree

Figure 2 Three Typical Data Center Architecture

Tree structure is commonly used in enterprise data centers, which is built from high-cost hardware. Due to the cost of the equipment, the capacity between different branches of the tree is typically oversubscribed by factors of 1:5 or more, which limits the communications between servers/virtual machines. VL2 and Fat-Tree are designed for cloud-oriented data centers built from commodity switches, providing extensive path diversity between servers. Compared to Fat-tree, VL2 has more redundant paths for each VM-VM pairs. We mark each links according to the approach we proposed in Section 3, the number is shown on each physical links.

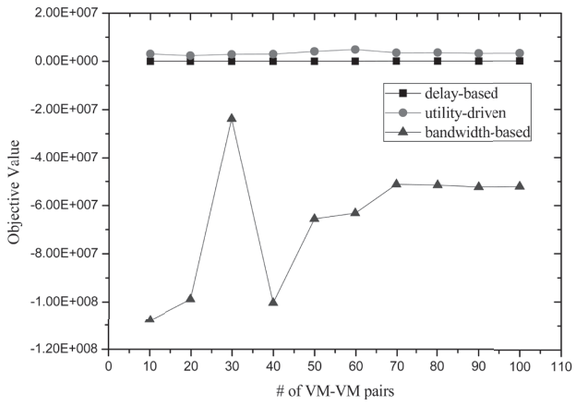
In all simulations, bandwidth capacity of each link is set to be 10 Gbps. We simulate a group of applications (VM-VM pairs) running in all network topologies. The scales range from 10 to 100 VM-VM pairs. The throughput-sensitive and delay-sensitive coefficients of applications are set to be random within [1, 10], the minimum bandwidth requirements are set to be random within [1, 10] Mbps. In each experiment, we set parameters in the interior point algorithm with maximum function evaluation times being 100,000 to guarantee the algorithm will converge to the optimal solution.

6.1 Comparisons of the Objective Value

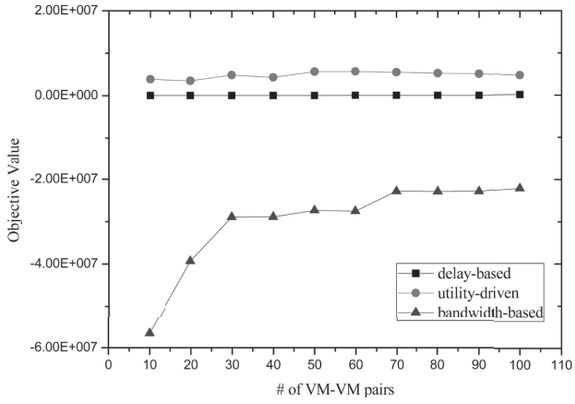
We set the derivative of the delay function according to Equation (8). Figure 3 shows the comparisons of the objective value in three different types of bandwidth allocation algorithm. In all three network topologies, the utility-driven bandwidth allocation algorithm has the highest objective values in different settings. According to the setting of the derivative of the delay function, the value of the delay function plays a more important role in the evaluation of the objective value. So the delay-based algorithm is better than bandwidth-based mechanism. Because the bandwidth-based algorithm doesn't consider the delay-sensitive feature of cloud applications, it's the worst bandwidth allocation algorithm in this scenario.

Though the result of delay-based scheme approaches our scheme, if we compare these two schemes in detail, we can find that the objective value of our scheme has average 85 times more than the value of delay-based mechanism in tree topology. And the number is 108 and 165 in VL2 and Fat-Tree topology respectively.

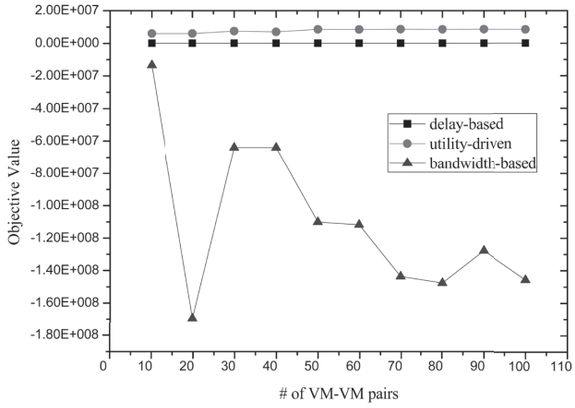
From Figure 3, we can also find the allocation result is fluctuant in Tree and Fat-Tree topologies when we use bandwidth-based algorithm. The reason is that these two topologies have less redundant paths than VL2 topology. Take the communication between VM 1 and VM 16 as an example, the number of redundant paths is 8, 16, 4 respectively in Tree, VL2 and Fat-Tree topology.



(a) Tree



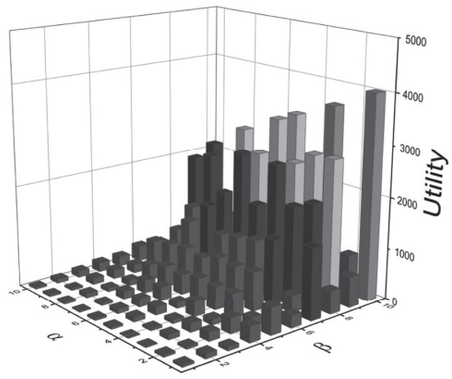
(b) VL2



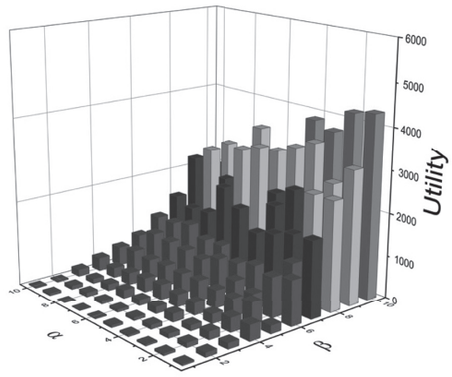
(c) Fat-Tree

Figure 3 Comparisons of the Objective Value

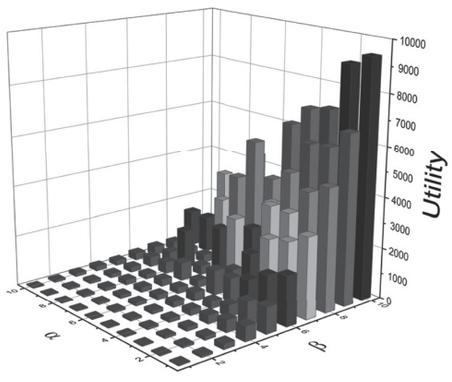
coefficient and delay sensitive coefficient range from 1 to 10. The results are shown in Figure 4.



(a) Tree



(b) VL2



(c) Fat-Tree

Figure 4 Comparisons of Application Utilities

6.2 Comparisons of Application Utilities

To show how our utility-driven bandwidth allocation scheme works for different types of cloud applications, we conduct experiments in three structures with 100 different types of applications (VM-VM pairs), each pair represent one type of application. Both the throughput sensitive

It is shown that in all topologies, when we fix throughput sensitive coefficient α, the utility that application obtained increases with the growing of delay sensitive coefficient β. However, when we fix delay sensitive coefficient β, the utility appears to be fluctuant. This phenomenon most probably resulted from two

reasons: (1) the minimum bandwidth requirements vary from different applications, value of minimum bandwidth requirement determines the basic utility of throughput sensitive application. (2) VMs belonging to throughput sensitive applications may be placed in the same physical server. Therefore, the utility introduced by throughput is constrained by the capacity of access links. The inherent reason behind these two reasons is: the placement of virtual machines/VM-VM pairs affects the result. Taking VM positions into consideration to share the data center network is a direction of our future research.

The utility of applications is not only determined by the combination of throughput sensitive and delay sensitive coefficient, but also determined by the capacity of the physical links it uses and other applications who compete for those links. In Figure 4(a), because the Tree topology has the least number of physical links (16 in our simulations), all applications compete for those rare capacity. Some applications such as application with $\alpha = 2$, $\beta = 10$ does not get deserved utility. This condition has been improved with the use of cloud-oriented topology Fat-Tree and VL2. Furthermore, the VL2 topology has the largest number of redundant path for each application. Though the largest utility in VL2 is less than the one in Fat-Tree topology, the utility each application got is more regular.

7 Conclusions

In this paper, we studied the network sharing problem in cloud data centers. Instead of only considering how many link bandwidths should be allocated to applications, we also consider the path delay that is experienced by each application. Utility function is constructed for each application according to the importance of throughput and delay sensitivities. We reformulate the utility maximization problem to a convex problem, hence the unique optimal solution can be found by applying standard convex solvers. Extensive numerical simulations verified that our proposed mechanism outperforms bandwidth-only and delay-only allocation schemes. The mechanism benefits both tenants and cloud providers. Besides, it opens the door for designing differentiated bandwidth pricing model and the on-demand access to network resource offered by cloud data centers becomes possible.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities (No. 2013RC1104); the 863 Program of China (No. 2013AA013303); the National Natural Science Foundation of China (No. 61002011).

References

[1] Netflix, http://www.netflix.com/.

[2] Amazon Elastic Compute Cloud (Amazon EC2), http://aws.amazon.com/ec2/.

[3] A. Kivity, Y. Kamay, D. Laor and U. Lublin, A. Liguori, Kvm: The Linux Virtual Machine Monitor, *Linux Symposium*, Ottawa, Ontario, 2007, pp. 225-230.

[4] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield, Xen and the Art of Virtualization, *ACM Symposium on Operating Systems Principles*, New York, 2003, pp. 164-177.

[5] L. I. B. López, Á. L. V. Caraguay, L. J. García, D. López, Trends on Virtualisation with Software Defined Networking and Network Function Virtualization, *IET Networks*, Vol. 4, No. 5, pp. 255-263, September, 2015.

[6] S. Kandula, S. Sengupta, A. Greenberg, P. Patel and R. Chaiken, The Nature of Data Center Traffic: Measurements & Analysis, *ACM SIGCOMM Internet Measurement Conference*, Chicago, IL, 2009, pp. 202-208.

[7] T. Benson, A. Anand, A. Akella and M. Zhang, Understanding Data Center Traffic Characteristics, *ACM Sigcomm Computer Communication Review*, Vol. 40, No. 1, pp. 92-99, January, 2009.

[8] H. Ballani, P. Costa T. Karagiannis and A. Rowstron, Towards Predictable Datacenter Networks, *ACM Sigcomm Computer Communication Review*, Vol. 41, No. 4, pp. 242-253, August, 2011.

[9] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy and I. Stoica, FairCloud: Sharing the Network in Cloud Computing, *ACM Sigcomm Computer Communication Review*, Vol. 42, No. 4, pp. 187-198, August, 2012.

[10] V. The Lam, S. Radhakrishnan, R. Pan, A. Vahdat and G. Varghese, Netshare and Stochastic Netshare: Predictable Bandwidth Allocation for Data Centers, *ACM Sigcomm Computer Communication Review*, Vol. 42, No. 3, pp. 5-11, July, 2012.

[11] T. Lam and G. Varghese, *Netshare: Virtualizing Bandwidth within the Cloud*, Technical Report, February, 2009.

[12] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu and Y. Zhang, SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees, *ACM Conference on Emerging Networking EXperiments and Technologies*, Philadelphia, PA, 2010, pp. 620-622.

[13] A. Shieh, S. Kandula, A. Greenberg, C. Kim and B.

Saha, Sharing the Data Center Network, *USENIX Conference on Networked Systems Design and Implementation*, Boston, MA, 2011, pp. 309-322.

[14] D. Clark, B. Braden and S. Shenker, *Integrated Service in the Internet Architecture: An Overview*, IETF RFC, 1994.

[15] D. Li, J. Zhu, J. Wu, J. Guan and Y. Zhang, Guaranteeing Heterogeneous Bandwidth Demand in Multitenant Data Center Networks, *IEEE/ACM Transactions on Networking*, Vol. 23, No. 5, pp. 1648-1660, Octobor, 2015.

[16] L. Yu and H. Shen, Bandwidth Guarantee under Demand Uncertainty in Multi-tenant Clouds, *IEEE International Conference on Distributed Computing Systems*, Madrid, Spain, 2014, pp. 258-267.

[17] D. Xie, N. Ding and Y. C. Hu and R. Kompella, The Only Constant Is Change: Incorporating Time-Varying Network Reservations in Data Centers, *ACM SIGCOMM*, Helsinki, Finland, 2012, pp. 199-210.

[18] C. Fuerst, S. Schmid L. Suresh and P. Costa, Towards Elastic Performance Guarantees in Multi-tenant Data Centers, *2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Portland, OR, 2015, pp. 433-434.

[19] M. Alizadeh, A. Kabbani, T. Edsall, B. Prabhakar, A. Vahdat and M. Yasuda, *Less Is More: Trading a Little Bandwidth for Ultra-Low Latency in the Data Center, 9th USENIX Conference on Networked Systems Design and Implementation*, San Jose, CA, 2012, p. 19.

[20] V. Jeyakumar, A. Kabbani, J. C. Mogul and A. Vahdat, *Flexible Network Bandwidth and Latency Provisioning in the Datacenter*, arXiv:1405.0631, May, 2014.

[21] J. C. Mogul and R. R. Kompella, Inferring the Network Latency Requirements of Cloud Tenants, *15th USENIX conference on Hot Topics in Operating Systems*, Santa Clara, CA, 2015, p .24.

[22] Y. Li, H. Wang, J. Dong and S. Cheng, Application Utility-based Bandwidth Allocation Scheme for Data Center Networks, *2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, Switzerland, 2012, pp. 268-273.

[23] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya and A. Vahdat, Portland: A Scalable Fault-tolerant Layer 2 Data Center Network Fabric, *ACM SIGCOMM*, Barcelona, Spain, 2009, pp. 39-50.

[24] Cisco, *Cisco Data Center Infrastructure 2.5 Design Guide*, 2007.

[25] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel and S. Sengupta, VL2: A Scalable and Flexible Data Center Network, *ACM SIGCOMM Computer Communication Review*, Vol. 39, No. 4, pp. 51-62, October, 2009.

[26] Y. A. Korilis, A. A. Lazar and A. Orda, Architecting Noncooperative Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 13, No.7, pp. 1241-1251, September, 1995.

[27] Y. Korilis, A. Lazar and A. Orda, Capacity Allocation under Noncooperative Routing, *IEEE Transactions on Automatic Control*, Vol. 42, pp. 309-325, March, 1997.

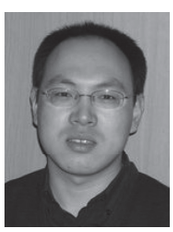
[28] A. Orda, R. Rom and N. Shimkin, Competitive Routing in Multiuser Communication Networks, *IEEE/ACM Transactions on Networking*, Vol. 1, pp. 510-521, August, 1993.

[29] U. Javed, M. Suchara and J. He and J. Rexford, Multipath Protocol for Delay-sensitive Traffic, *First International Conference On Communication Systems and Networks*, Bangalore, India, 2009, pp. 438-445.

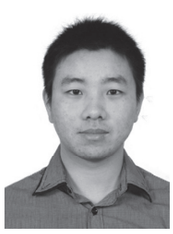
[30] B. Fortz and M. Thorup, Optimizing OSPF/IS-IS Weights in a Changing World, *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 4, pp. 756-767, May, 2002.

[31] S. S. Sapatnekar, *Convex Optimization: Wiley Encyclopedia of Electrical and Electronics Engineering*, Wiley, 1999.

Biographies



Hongbo Wang received his BS degree in Computer software from Hebei University, China, in 1998 and PhD degree in Computer application technology at the Beijing University of Posts and Telecommunications (BUPT) in 2006. Currently he is an Associate Professor in State Key Laboratory of Networking and Switching Technology of BUPT. His main research interests cover cloud computing, big data, data center network, overlay network, and Internet measurement, in which he has published over 50 technical papers in referred journals and conference proceedings.



Yangyang Li received his BS degree in information engineering from Nanjing University of Information Science and Technology, China, in 2009 and PhD degree in computer science from Beijing University of Posts and Telecommunications, China, in 2015. He joined the Innovation Center, China Academy of Electronics and Information Technology, Beijing, China, as

a research engineer in 2015. His research area of interests includes cloud computing, data center networking and future networking.

Shiduan Cheng received her BS degree in department of Wired network at the Beijing University of Posts and Telecommunications (BUPT) in 1963. Currently she is a professor of BUPT. In 1980's and 1990's she twice joined Alcatel Bell, Belgium as a visiting scholar and involved in ISDN and ATM research work. From 1992 to 1999 she was the director of the National Laboratory of Switching Technology and Telecommunication Networks and a member of the steering committee of communications in national 863 program. Her current research interests are next generation Internet, QoS of Internet and data center networks.