Improving Bandwidth Efficiency and Fairness in Cloud Computing

---

Improving Bandwidth Efficiency and Fairness in Cloud Computing

Xiang Sun and Nirwan Ansari
Electrical & Computer Engineering Dept., New Jersey Institute of Technology, Newark, NJ 07102, USA
E-mail: xs47@njit.edu, nirwan.ansari@njit.edu

***Abstract*— Bandwidth is a key resource in cloud networks. Every tenant wants to be assigned the bandwidth which is proportional to the price they have paid. At the cloud vender side, the link bandwidth utilization could be enhanced to support more clients. In this paper, we show that the traditional PS-N (Proportional Sharing at Network level) bandwidth allocation algorithm cannot achieve the network proportionality fairness when the network is over-subscribed. PPSN (Persistence Proportional Sharing at Network level) is proposed to solve the unfairness issue. However, the bandwidth utilization of both algorithms is not good enough to meet venders' demands. BEPPS-N (Bandwidth Efficiency Persistence Proportional Sharing at Network level) is thus proposed to enhance the bandwidth utilization by assigning more bandwidth to the communication pairs that are not passing through bottleneck links, and at the same time, keep proportionality fairness per different tenant. Finally, simulations and performance analysis have been conducted to substantiate the viability of our proposed approach.**

Keywords- cloud computing; fairness; bandwidthutilization; high utilization

I. INTRODUCTION

Cloud computing, according to NIST [1], enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly subscribed and released with minimal management effort or service provider interaction. However, the NIST model is not mature enough to be widely used. There are many dramatic challenges in cloud computing. First, how does the cloud vender set a price for renting different resources and the utilization rate of users in the cloud computing? Second, how does a vender fairly provide the services proportionally to the price that each user pays?

At the network level, the biggest resource is bandwidth. Regardless of the performance of VM (Virtue Machine), if a tenant can acquire more bandwidth, it takes less time to accomplish parallel computing or other applications. Therefore, to be fair, bandwidth allocated to tenants should be proportional to the prices they pay. Popa *et al.* [2] proposed that other than fairness, two other requirements for allocating bandwidth in a well-designed cloud network should be considered: min-bandwidth guarantee and high utilization. Min-bandwidth guarantee refers to the minimum bandwidth warranted for every communication between two VMs regardless of the network traffic. High utilization means that every communication pair is incentivized to use the available bandwidth of the link as much as possible. However, there are tradeoffs between different requirements, implying that we cannot realize the min-guarantee, network proportionality and high utilization simultaneously.

In this paper, our main focus is on designing a bandwidth allocation algorithm to realize network fairness: bandwidth allocation for each tenant is proportional to the weight of the tenant, which reflects the price value. Meanwhile, the tradeoff between high utilization and network fairness should favor in improving the bandwidth utilization as much as possible.

The rest of the paper proceeds as follows. Section 2 provides background and presents related works on fairness in bandwidth allocation and other resource allocation algorithms. Section 3 briefly introduces the PS-N algorithm and points out its drawbacks. Section 4 proposes the PPS-N bandwidth allocation algorithm based on PS-N to solve the unfairness problem in the over-subscribed condition. Section 5 proposes the BEPPS-N algorithm based on PPS-N to enhance the bandwidth utilization as much as possible. Section 6 concludes with a summary and discussion of future work.

II. BACKGROUND AND RELATED WORKS

Resource sharing in a cloud network can be classified into two major types: VM-based and network-based. In the VM-based sharing, the study focuses on how to fairly map limited physical resources, such as CPU, RAM and network card, into different VMs. In network-based sharing, how to assign the bandwidth proportional to the weight of different tenants is the key problem. Recent works have focused on how to map physical resources into VM fairly [3]-[7], but few have addressed how to share the network bandwidth according to the network topology. Gabale *et al.* [11] proposed an algorithm to realize QoE fairness of video delivery in cloud data centers. The algorithm defines the QoE fairness as the number and duration of playout stalls in the video delivery process to be proportional to the flow weight. It uses buffer size of the client side and the TCP window size to determine the flow weight. However, the number and duration of playout stalls proportional to the flow weight does not mean that a tenant's allocated bandwidth is proportional to the weight of the tenant in cloud network. So, we cannot apply this algorithm in cloud networks. Lee and Chong [8] proposed a flow control algorithm based on the per source-destination pair to realize max-min fair allocation of bandwidth utility in different applications; however, per source-destination pair cannot achieve flow rate fairness [2][9] since increasing the communication pairs between different VMs results in acquiring more bandwidth. Shieh *et al.* [10] provided every source VM with a bandwidth share that is proportional to its weight; although this allocation is fair to the source, it does not meet the network proportionality since it is not fair to destinations [2]. Popa *et al.* [2] proposed three bandwidth allocation algorithms to achieve network proportionality. Nevertheless, PS-L (Proportional Sharing at Link Level) can guarantee the link level based proportionality, but cannot ensure that tenants with the same weight get the same bandwidth in the whole network. PS-N aims to assure the network level proportionality, but when the network is oversubscribed, it cannot guarantee proportionality between tenants any more. PS-P (Proportional Sharing on Proximate link) is proposed to realize the min-bandwidth guarantee, but it cannot enforce the proportionality fairness.

Based on the above, we propose to modify the PS-N algorithm such that the bandwidth allocation can be strictly proportional to the weights of the tenants even under the oversubscribed condition. We also try to decrease the tradeoff between high utilization and network proportionality that is critical to cloud vendors

III. PROPORTIONAL SHARING AT NETWORK-LEVEL

To easily understand the proposed algorithm in the paper, we first briefly introduce PS-N. Both PS-N and our proposed algorithms are based on the IaaS (Infrastructure-as-a-Service) model, such as Amazon EC2 [12], so that tenants pay a fixed flat-rate per VM. The key idea of PS-N is to use a weight model of communication between VM *X* and *Y* as follows [2]:

$$W_{X-Y} = \frac{W_X}{N_X} + \frac{W_Y}{N_Y} \tag{1}$$

where N_X (N_Y) is the number of other VMs that $X(Y)$ communicates with on the entire network and $W_X(W_Y)$ is the weight of VMs $X(Y)$. Therefore, the weight of each communication pair can be calculated. Suppose that $\boldsymbol{C_m}$ is the set of communication pairs passing through the link L_m, so the bandwidth allocated to flow $i\leftrightarrow j$ of a certain tenant on the link L_m can be expressed as:

$$BW_{i\leftrightarrow j/L_m} = \frac{W_{i-j}}{W_{total/L_m}} \times BW_{initial/L_m} \tag{2}$$

where W_{i-j} is the weight of the flow based on expression (1); $BW_{initial}$ is the bandwidth capacity of link $L_m$and W_{total/L_m} is the total weight of all the flows on link L_m; if there are *n* tenants and every tenant has a set of **V** flows, so W_{total/L_m} can be expressed as:

$$W_{total/L_m} = \sum_{x=1}^{n}\left[\sum_{i,j\in \mathbf{V}_x \& i\neq j \& i\leftrightarrow j\in \mathbf{C_m}} \left(\frac{W_{x_i}}{N_{x_i}} + \frac{W_{x_j}}{N_{x_j}}\right)\right] \tag{3}$$

If the flow$i\leftrightarrow j$ has to pass *N* links in the network, the bandwidth allocated to it should be:

$$BW_{i-j} = \min\left(BW_{i-j\in C_1}, BW_{i-j\in C_2} \ldots BW_{i-j\in C_m} \ldots BW_{i-j\in C_N}\right) \tag{4}$$

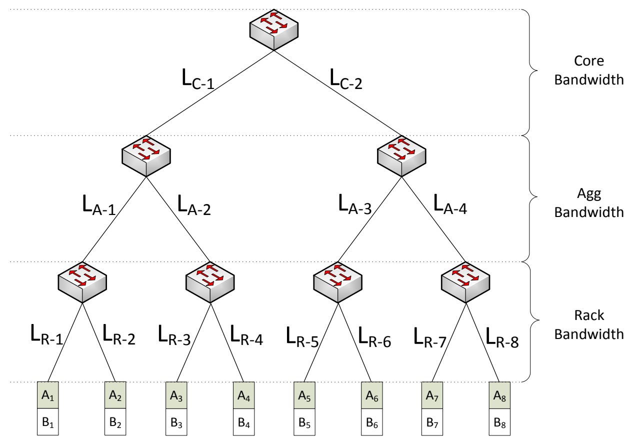
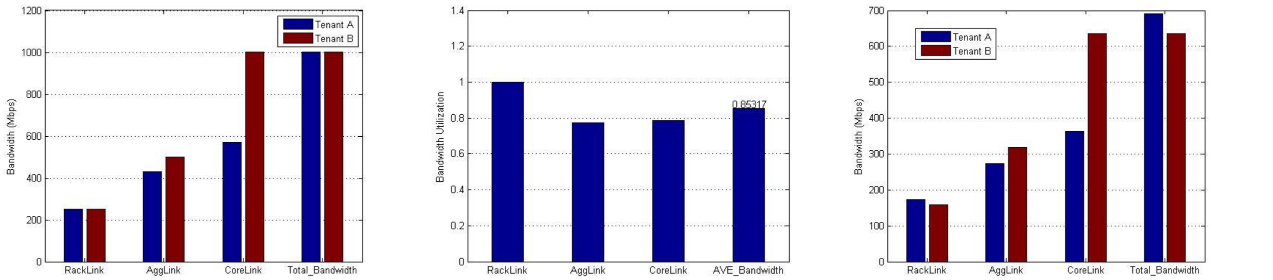


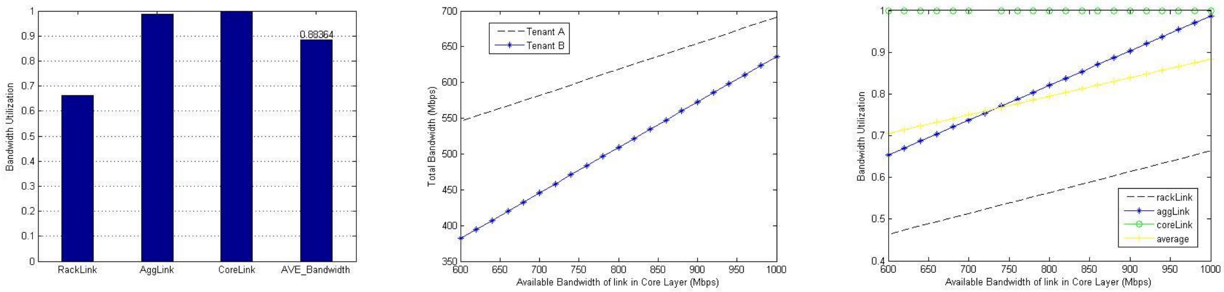
Figure 1. The network topology

For instance, as shown in Figure1, if the network is a tree-based topology with eight servers and there are two tenants *A* and *B* in the network, each tenant owns one VM on each server. We suppose that the communication type of VMs in tenant *A* is $A_i\leftrightarrow A_j$ ($i\neq j$, i,j={1,2,3…8}) and the communication type of VMs in tenant *B* is $B_i\leftrightarrow B_j$ ($i=j\pm 4$, i={1,2,3…8}). For simplicity, assume that all the VMs have unit weight and have an infinite amount of data to send each other (i.e., infinite bidirectional communication task). Core and Agg switches are fully provisioned so that the bottleneck of the network is at the rack layer. In this case, the weight of every flow of tenant *A* should be 1/7+1/7=2/7 and the weight of every flow of tenant *B* should be 1+1=2. Taking L_{R-1} link for example, there are 8 flow on L_{R-1} in which tenant *A* takes up 7 flows and tenant *B* has 1 flow so that $W_A{:}W_B=1{:}1$ on L_{R-1}. If the bandwidth capacity of L_{R-1} is *C*, then tenant *A* and *B* can each get *C/2* bandwidth. Based on Eq. (1), we can see that not only every tenant but also every flow can achieve proportional fairness based on PS-N.

Suppose that the bandwidth capacity at the Rack, Agg and Core layer's link is 500Mbps, 1.2Gbps and 2Gbps, respectively. We verify the performance of PS-N based on the topology in Figure 1. The results are shown in Figure 2(a) & (b). Tenants *A* and *B* get different bandwidth on different layers' links but the total bandwidth that both tenants own is the same. However, there is a tradeoff between proportional fairness and bandwidth utilization as we can see that the bandwidths of links on Agg and Core are not 100% utilized.



(a) Bandwidth allocation on different layer of link (b) Bandwidth utilization on different layer of link (c) Bandwidth allocation on different layer of link



(d) Bandwidth utilization on different layer of li (e) Total bandwidth allocation to tenants A and B (f) Bandwidth utilization on different layer

Figure 2. Simulation results of PS-N. (a)(b) are the results when core and agg switches are fully provisioned; (c)-(f) are the results of network oversubscribed by 4×

Popa *et al.* [2] also considered another scenario that if each of the core and aggregation layer is oversubscribed by two times, i.e., equivalently, the bandwidth capacity of each link at Agg layer and Core layer reduces to 600Mbps and 1Gbps, respectively. In this case, the core layer's links become the bottleneck. We simulate the PS-N algorithm as well and the results are shown in Figure 2(c) & (d). We can see from Figure 2(c) that the total bandwidth allocated to tenants *A* and *B* are not the same because the PS-N algorithm allocates the bandwidth to all the flows based on the rack layer which is not conform to Eq. (4). The benefit of doing that is PS-N can be implemented through a distributed mechanism similar to Seawall [10]. However, when the core switches are the bottleneck, the bandwidth value of flows passing through the core switches should be decreased. Therefore, the bandwidth allocation does not adhere to Eq. (2) any more at the rack layer. In Figure 2(d), the average bandwidth utilization is improved because there is a tradeoff between fairness and bandwidth utilization, i.e., bandwidth utilization is enhanced at the expense of fairness.

Reducing the bandwidth capacity of Core layer's links from 1G bps to 600 Mbps implies that the oversubscribed factor becomes much larger so that the flows that pass through the Core layer are assigned less bandwidth. In this case, as shown in Figure 2(e), the difference of total bandwidth allocated to tenants *A* and *B* becomes larger when the bandwidth capacity of Core layer's links decreases, i.e., the unfairness is much more pronounced. However, as shown in Figure 2(f), the bandwidth utilization is also reduced, implying that the tradeoff between fairness and bandwidth utilization is not efficiently done as the bandwidth capacity of Core layer's links decreases. So, PS-N does not perform well when the core and aggregation layers are oversubscribed.

IV. PERSISTENCE PROPORTIONAL SHARING AT NETWORK LEVEL (PPS-N)

In order to solve the unfairness problem when the network is oversubscribed, we need to improve the PS-N algorithm. Since all the flows need to pass through the rackLink, and thus PS-N can be implemented in a distributed manner. However, due to unawareness of the situation in the coreLink and aggLink, the bandwidth allocated to the links based on the rackLink may exceed the capacity of coreLink and aggLink in an oversubscribed scenario so that the bandwidth value calculated in the rackLink needs to be decreased in order to meet the bottleneck bandwidth capacity. The re-allocated bandwidth no longer satisfies Eq. (2), thus resulting in unfairness.

To address the drawbacks discussed in Section III, we propose persistence proportional sharing at the network level (PPS-N), which is based on PS-N, but the algorithm allocates the bandwidth based on the bottleneck link so that the total bandwidth of the flows at other layers would not exceed their link capacity. For instance, in Figure 1, when the core layer becomes the bottleneck, L_{C-1} and L_{C-2} are the bottleneck links in the network, the bandwidth allocated to the flows which are passing through the bottleneck links should be calculated first as follows:

$$BW_{i-j} = BW_{i-j\in L_{\lambda}}, \quad L_{\lambda} \in L_{bottleneck} \tag{5}$$

Second, the bandwidth allocation of communication pairs that are not passing through bottleneck links should be calculated based on the weight proportionality:

$$BW_{x\leftrightarrow y} = \frac{W_{x\leftrightarrow y}}{W_{i\leftrightarrow j}} \times BW_{i\leftrightarrow j\in L_{\lambda}}, \quad x \leftrightarrow y \notin L_{bottleneck} \tag{6}$$

Third, the bottleneck links should have less bandwidth capacity and more flows passing through it, and so we define bottleneck links as

$$L_{bottleneck} = \min\left\{\frac{\text{The initial bandwidth of link L}}{\text{The total weight on link L}}, L \in \{L_1, L_2 ... L_N\}\right\} \quad (7)$$

The expression of calculating bottleneck link should be:

$$L_{bottleneck} = \min\left\{\frac{BW_{available/L}}{\sum_{x=1}^{n}\left(\sum_{i,j\in V_X, i\neq j, i\leftrightarrow j\in L}\left(\frac{W_{x_i}}{N_{x_i}}+\frac{W_{x_j}}{N_{x_j}}\right)\right)}, L\in\{L_1, L_2 ... L_N\}\right\} \quad (8)$$

We simulate the PPS-N based on the network topology in Figure 1, and each of the Core and Agg layer is oversubscribed by two times as well. The results are shown in Figure 3(a) with the same total bandwidth assigned to tenants A and B, approximately 636Mbps. Thus, PPS-N can achieve proportional fairness at the network level. However, as mentioned earlier, there is a tradeoff between fairness and bandwidth utilization. So, the bandwidth utilization, as shown in Figure 3(b), decreases slightly as compared with PS-N. We also simulate the case of decreasing every link's bandwidth capacity at the core layer from 1 Gbps to 600 Mbps. As shown in Figure 4(a), the bandwidth allocated to both tenants remains the same regardless of the value of the bandwidth capacity at the core layer. However, as shown in Figure 4(b), the bandwidth utilization is reduced as the bandwidth capacity at the core layer's link decreases, and the bandwidth utilization is smaller than that achieved by PS-N, since in PPS-N, the bandwidth allocation of communication pairs, which are not passing through a bottleneck link, should follow Eq. (6) which is smaller than the one in PS-N:

$$\begin{aligned} BW_{x\leftrightarrow y} &= \frac{W_{x\leftrightarrow y}}{W_{i\leftrightarrow j}} \times BW_{i\leftrightarrow j/L_\lambda}, \text{ x} \leftrightarrow \text{y} \notin L_{bottleneck} \\ &\leq \frac{W_{x\leftrightarrow y}}{W_{i\leftrightarrow j}} \times BW_{i\leftrightarrow j} = \frac{W_{x\leftrightarrow y}}{W_{total}} \times BW_{available} \end{aligned} \quad (9)$$

Therefore, with a larger overprovisioned factor, the tradeoff between high utilization and network fairness becomes inefficient. The bandwidth utilization on rackLink is less than 40%, which is not acceptable to cloud venders.

V. Bandwidth Efficiency Persistence Proportional Sharing in Network Level (BEPPS-N)

To alleviate low bandwidth utilization, we propose the Bandwidth Efficiency Persistence Proportional Sharing in Network level (BEPPS-N) algorithm. The key idea of BEPPS-N is to increase the bandwidth allocation to the flows which are not passing through bottleneck links but to keep the bandwidth allocation proportionality at the same time. Since the flows that do not pass through bottleneck links in PPS-N are constrained by Eq. (6), so that even though there is available bandwidth on the link, such as L_{A-1}& L_{A-2} at the agg layer and L_{R-1}&L_{R-2} at the rack layer, the communication pairs cannot acquire it. BEPPS allows this non-bottleneck link's available bandwidth to be allocated to the communication pairs, and in order to keep tenant-level fairness, the total allocated bandwidth to different tenants should be proportional to their weights. However, BEPPS-N no longer keeps proportional fairness in the communication pair level anymore since the bandwidth assigned to each communication pair is not proportional to its weights value. More bandwidth is assigned to the pairs that are not passing through bottleneck links.

Consider an example to explain the principle of BEPPS-N in detail. If the network topology shown in Figure 1 is used and the core layer and agg layer are oversubscribed by four times, after all the communication pairs are allocated bandwidth by PPS-N, coreLink is still the bottleneck. Tenants A and B have the same weight so that we need to increase the same bandwidth for both tenants to improve the bandwidth utilization. Suppose the increment of bandwidth for both tenants is *ΔB*, we need to allocate this bandwidth to the flows that are not passing through bottleneck links.

The allocation of *ΔB* depends on the network topology and we divided the network topology into three different types. Our later work will explain how BEPPS allocate *ΔB* to the corresponding flows and here we only explain the intuitive of BEPPS.

The total bandwidth assigned to different flows on a certain link should not exceed the link's capacity. So at the core layer, the increment *ΔB* of tenant *B* cannot exceed the bandwidth allocated to tenant *A* in the PPS-N algorithm.

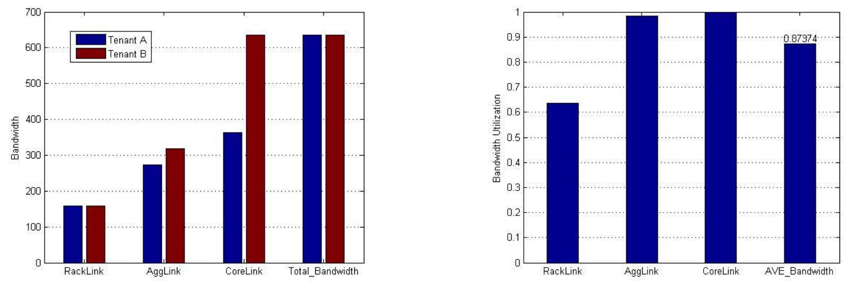
$$\Delta B_c \leq \sum_{AiAj\in L_{core}} BW_{AiAj} \quad (10)$$

where $\boldsymbol{L_{core}}$ is the set of flows which pass through a certain core layer's link; BW_{AiAj} is the weight of the communication pair between VM_{Ai} and VM_{Aj}.

So, for the same reason for any link on the agg:

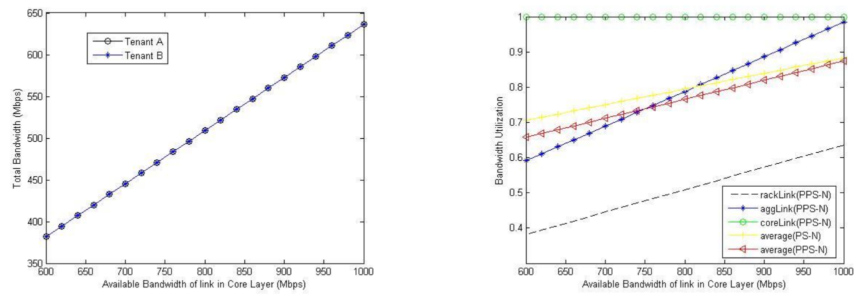
$$\begin{aligned} &\Delta B_a \times \frac{\sum_{BiBj\in(L_{agg}\cap L_{bottleneck})} W_{BiBj}}{\sum_{BiBj\in L_{bottleneck}} W_{BiBj}} - \Delta B_a \times \frac{\sum_{AiAj\in(L_{agg}\cap L_{bottleneck})} W_{AiAj}}{\sum_{AiAj\in L_{bottleneck}} W_{AiAj}} \\ &+2\times\Delta B_a \times \frac{\sum_{AiAj\in(L_{agg}\oplus L_{parent})} W_{AiAj}}{\sum_{AiAj\notin L_{bottleneck}} W_{AiAj}} \leq BW_{agg-left} \end{aligned} \quad (11)$$

where $\boldsymbol{L_{agg}}$ is the set of flows which pass through a certain agg layer's link; $\boldsymbol{L_{parent}}$ is the set of flows on the parent link of certain agg layer's link. For instance, as shown in Figure 1, if we need the maximum value of ΔB_a on the L_{A-1} link, then the parent link of L_{A-1} is L_{C-1} and $\boldsymbol{L_{parent}}$ $=\{A_1A_5, A_1A_6, A_1A_7, A_1A_8\ldots\}$; $\boldsymbol{L_{bottleneck}}$ is the set of flows on the bottleneck links; $BW_{agg\text{-}left}$ is the available bandwidth on certain agg link after the PPS-N algorithm has been executed.



(a) Bandwidth allocation on different layer of link (b)Bandwidth utlization on different layer of link

Figure 3. Simulation results of PPS-N in the condition of network oversubscribed by 4×



(a) Total bandwidth allocation to tenants A and B (b) Bandwidth utilizationon different layer

Figure 4. Simulation results of PPS-N by changing the available bandwidth at the core layer

Any link at the rack layer should follow:

$$\Delta B_r \times \frac{\sum_{BiBj \in (L_{rack} \cap L_{bottleneck})} W_{BiBj}}{\sum_{BiBj \in L_{bottleneck}} W_{BiBj}} - \Delta B_r \times \frac{\sum_{AiAj \in (L_{rack} \cap L_{bottleneck})} W_{AiAj}}{\sum_{AiAj \in L_{bottleneck}} W_{AiAj}} + 2 \times \Delta B_r \times \frac{\sum_{AiAj \in (L_{rack} \oplus L_{father})} W_{AiAj}}{\sum_{AiAj \notin L_{bottleneck}} W_{AiAj}} \le BW_{rack-left} \quad (12)$$

$\boldsymbol{L_{rack}}$ is the set of flows which pass through a certain rack layer's link; $BW_{rack\text{-}left}$ is the available bandwidth at a certain rack link as a result of the PPS-N algorithm.

So, the value of ΔB in the whole network should be $\min\{\max(\Delta B_c), \max(\Delta B_r), \max(\Delta B_a)\}$.

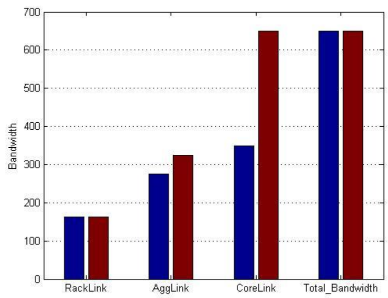
We simulate the BEPPS-N algorithm as well and the network condition is the same as those of PPS-N and PS-N with four times oversubscription. The results in Figure 5(a) show that BEPPS-N can realize fairness since the total bandwidth allocated to both tenants is proportional to their weights. Meanwhile, the average bandwidth utilization of BEPPS-N, as shown in Figure 5(b), is better than that of PPS-N and PS-N in Figure 3(b) and Figure 2(b), respectively. Then, we decrease the bandwidth capacity in the core layer from 1Gbps to 600Mbps; Figure 5(c) & (d) show that when the network resource is constrained, BEPPS-N can improve bandwidth utilization better than PPS-N and PS-N can, and it can still keep the fairness when the bottleneck link's bandwidth capacity changes. However, as shown in Figure 5(c), the total bandwidth allocation for and the average bandwidth utilization of different tenants do not increase linearly like those for PPS-N as shown in Figure 4(a). The average bandwidth utilization decreases when the initial bandwidth in coreLink is around 720Mbps and the reason for this is attributed to two aspects.

- The decreasing of ΔB

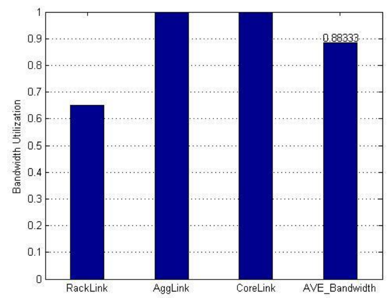
As shown in Figure 5(e), with the increase of the coreLink bandwidth capacity, the value of ΔB becomes smaller. This is because when the initial coreLink bandwidth capacity is less than 720Mbps, the value of ΔB is constrained by Eq. (10) at the core layer. Since, as shown in Figure 5(f), the bandwidth assigned to flows of tenant A in the core layer is diminished to 0, the whole bandwidth of the core layer is occupied by tenant B. Still, when the coreLink bandwidth capacity is greater than 720Mbps, the value of ΔB is constrained by Eq. (11) at the agg layer; in Figure 5(d), the bandwidth utilization at the agg layer reaches 100%, and in Figure 5(f), after 720 Mbps, the bandwidth of tenant A has residual bandwidth at the core layer. With the bottleneck links having more bandwidth capacity, the flows that passing through the bottleneck links can get more bandwidth as well; however, the initial bandwidth at the agg layer does not increase which makes $BW_{agg\text{-}left}$ in Eq. (11) much smaller so that the $\max(\Delta B_a)$ is also decreased.

- Bandwidth increment rate is smaller than decrement rate at the rack layer

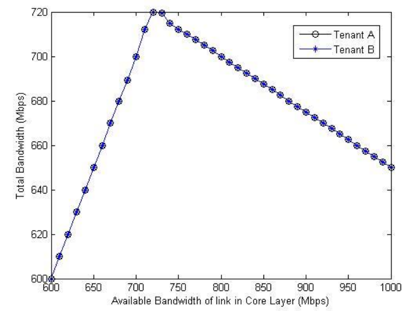
In Figure 5(d), we can see when the initial coreLink bandwidth is larger than 720Mbps, the bandwidth utilization reaches 100% at the agg layer, and so the only factor affecting the average bandwidth utilization is at the rack layer.



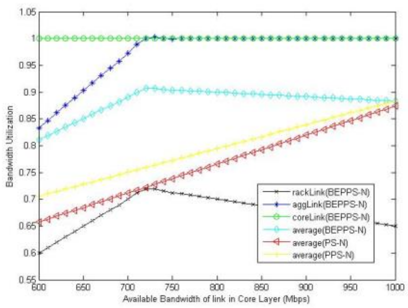
(a) Bandwidth allocation at different layers



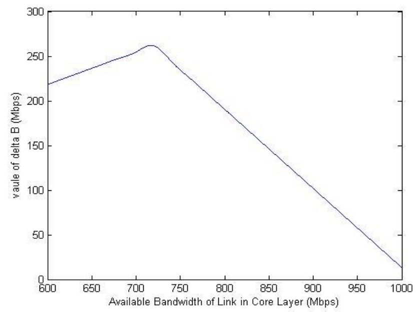
(b) Bandwidth utilization at different layers



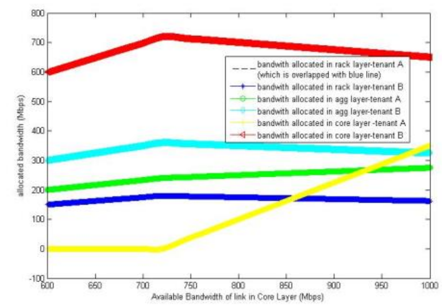
(c) Total bandwidth allocated to different tenants



(d) Bandwidth utilization of different layers



(e) Value of ΔB



(f) Bandwidth allocation for different tenants by using BEPPS

Figure 5. Simulation results of BEPPS-N in the condition of network oversubscribed

When the coreLink bandwidth capacity increases, as shown in Figure 4(b), the bandwidth utilization at the rack layer is enhanced by running PPS-N. However, when BEPPS-N wants to assign unused bandwidth (which is ΔB) to communication flows, this unused bandwidth that can be allocated becomes less when the coreLink bandwidth capacity is greater than 720Mbps. The rate of decreasing unused bandwidth assigned to the rack layer in BEPPS-N is faster than the rate of increasing bandwidth utilization at the rack layer in PPS-N when the initial bandwidth in coreLink is larger than 720Mbps.

VI. Conclusion and Future Work

Existing algorithms cannot achieve proportional fairness when the cloud network is oversubscribed. We have thus proposed PPS-N to realize restricted proportional fairness in an oversubscribed scenario. We have further proposed BEPPS-N to address the low bandwidth utilization problem. That it cannot realize proportional fairness at the finer granularity (at the flow level) is the shortcoming of BEPPS-N to be addressed in the future.

We will also investigate on implementing BEPPS-N in various topologies. We will divide the topology into three different types. On the other hand, how to decide the weight of a VM remains a difficult issue to be addressed.

References

[1] P. Mell and T. Grance. "The NIST definition of cloud computing," [online] National Institute of Standards and Technology (NIST). Available on csrc.nist.gov/groupslSNS/cloud-computingiclouddef-vI5.doc, Accessed on 2 October 2010.

[2] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, I. Stoica, “FairCloud: Sharing the network in could computing,” Proc. ACM SIGCOMM, Helsinki, Finland, Aug 13-17 ,2012

[3] B. Anwer, et al., “Network I/O Fairness in Virtual Machines”, Proc. ACM SIGCOMM, New Delhi, India, Aug 30-Sep 3, 2010.

[4] D. Dolev, et al., “No Justified Complaints:On Fair Sharing of Multiple Resources,” Proc. Innovations in Theoretical Computer Science Conference (ITCS 2012), Cambridge, MA, Jan 8-10, 2012.

[5] H. Kim, et al, “Task-aware virtual machine scheduling for I/O performance,” Proc. 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, Pittsburgh, PA, Mar 17-19, 2009. .

[6] S. Kuribayashi, “Optimal joint multiple resource allocation method for cloud computing environments,” International Journal of Research & Reviews in Computer Science, Vol. 2, No. 1, pp. 1-8, Jan, 2011. .

[7] T. Takuro and S. Kuribayashi, “Congestion control method with fair resource allocation for cloud computing environments,” Proc. 2011 IEEE Pacific Rim (PacRim 2011), Victoria, Canada, Aug 23-26, 2011. .

[8] H-W. Lee and S. Chong, “A distributed utility max–min flow control algorithm,” Computer Networks, Vol. 50, No. 10, pp. 1816-1830, Aug, 2006. .

[9] B.Briscoe,“Flow Rate Fairness:Dismantling a Religion”, Computer Communication Review, Vol. 37, No. 2, pp. 63-74, Apr, 2007. .

[10] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, “Sharing the Data Center Network”, Proc. 8th USENIX on networked System Design and Implementation, Boston, MA, Mar 30-Apr 1, 2011. .

[11] V. Gabale, P. Dutta, R. Kokku and S. Kalyanaraman, “Insite: QoE-Aware Video Delivery from Cloud Data Centers”, Proc. 2012 IEEE 20th International Workshop on Quality of Service (IWQoS), Coimbra, Portugal, Jun 4-5, 2012. .

[12] Amazon web service. http://aws.amazon.com.