Fairly Scheduled Bandwidth for the Data Centre Traffic

Manjur Kolhar
Dept. Computer Science
College of Arts and Science
Prince Sattam Bin Abdulaziz University

Abdalla Alameen
Dept. Computer Science
College of Arts and Science
Prince Sattam Bin Abdulaziz University

ABSTRACT
Maintaining the bandwidth across a data center to meet the service level agreement is a major challenge for network administrators. If the bandwidth is not scheduled according to application requirements, the applications will be executed with increased or decreased provisioning or moved to other data centers. Hence, we developed a bandwidth scheduling algorithm that operates according to application requirements and partitions the available bandwidth to accommodate the available traffic. We simulated this algorithm with the SimPy tool and found that the proposed algorithm performs better than the general allocation and partition schemes.

Keywords
Data center, cloud computing, bandwidth reservation, resource sharing.

1. INTRODUCTION
Internet users have grown in number over the past few years, and the Internet traffic associated with their applications has significantly increased [1]. Internet or application service providers allocate applications to large-scale data centers. Data center traffic is dynamic in nature. For example, only a small amount of traffic exists at time t0; however, the traffic might increase at time t1. The amount of traffic eventually reaches a threshold, at which point new requests are not allowed in the data center because of the lack of bandwidth. However, network administrators cannot discard the application request to use the data center because of the service level agreement (SLA). SLA is an agreement between the provider and the client for a service. Such scenarios are handled by moving applications to another data center. This concept will definitely degrade the provider and underutilize shared resources [2]. The following are the issues that arise from the migration of applications to other data centers.

- Implementation issues
- Cost and effect of migration
- Effect of associated traffic
- Bandwidth demand and its cost are prohibitive

A data center typically consists of routers and switches, with more specific and expensive network equipment moving up the network hierarchy. However, even these expensive and specific equipment may only serve 50% of the bandwidth of the data center at increased cost. Furthermore, non-uniform bandwidth assignment among data center nodes results in issues in the hosted applications; these issues affect the performance of applications [3]. Hence, providing sufficient bandwidth to applications is a significant challenge for network administrators [4-6]. Our proposed resource-sharing methodology employs firmness when dealing with bandwidth.

After considering these points, we designed a framework that dynamically allocates the required bandwidth to applications. The contribution of our work can be summarized in the following three aspects.

a With regard to application bandwidth partitioning, we partition the bandwidth according to network equipment present in an aggregation layer of the data center of a multi-tier architecture.

b To the best of our knowledge, our scheme is the first to support scalable bandwidth partitioning according to the traffic service classes of applications.

c Dynamic bandwidth allocation is implemented to accommodate different traffic classes.

Fig. 1 shows the categories of service. We differentiated services based on application requirements (real time or non-real time) and not on the network interface that is utilized by the system of the end user. When a request is forwarded to the data center from the end user, the request travels along multiple network paths and links and is subjected to various policies set by network administrators. A request can be a file download or upload, video conference, voice chat, browsing, and so on. The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the proposed scheme. The remaining sections present the performance of the proposed scheme, a discussion of the proposed scheme, and the conclusion of the study.

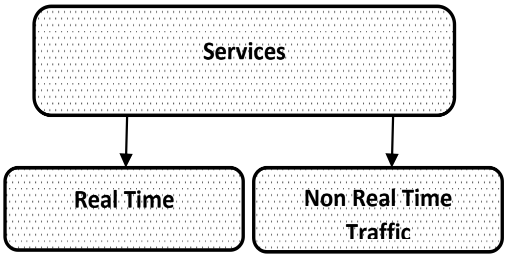

Fig. 1 Different services offered by the cloud

2. RELATED WORK
Bandwidth allocation for real-time and non-real-time services poses a challenge to the research community because of the increased number of end users and content delivery, which must be routed to the end user or cloud client. Several proposed solutions for the bandwidth allocation process are built on the concepts of market trading, game theory, and other relevant algorithms [7–11]. Desktop users offer their bandwidth share at a set price to mobile users. Bandwidth sharing is analogous to the concept of the Stackelberg game, in which desktops are the leaders and mobile users are considered the followers [7–8]. However, this solution is only for a small number of mobile users who are near desktop users. This solution is thus suitable for home users, provided that they are limited in number.

Task scheduling, which is associated with assigning virtual machines (VMs) to cloud users, was initially used for bandwidth utilization apart from important resources, such as CPU and memory [9]. Non-linear programming is used to divide the tasks/VMs in a cloud computing environment. This solution of dividing tasks is a useful approach with respect to CPU and memory. However, dividing a task/VM based on its bandwidth usage under cloud elasticity is not fit for this method. The obvious nature of the cloud computing environment is not considered.

A virtual local area network (VLAN) is implemented to provide scalable traffic management in the cloud by exploring a small search space; this process avoids the NP-complete VLAN mapping problem. However, it still lags in providing a solution for irregular demands from cloud clients. Even software solutions have been proposed to avoid the interference induced by hardware components [9]. Such solutions only burden the CPU and memory and eventually affect the performance of the entire system. An asymptotic evolution algorithm has been proposed to allocate bandwidth [10]. It calculates the pre-allocation factor at the router by considering the load and instantaneous queue length. Hence, with this calculation, the sender is aware of the rate of transmission to reach the stable state. Fair-share methods were implemented by Chaisiri et al. to control network resources and avoid having shared resources interfere with one another. The unevenly distributed transmission control protocol (TCP) flows were rectified with the aid of fair-share methods. However, the fair-share method underperforms when differentiating quality of service (QoS) for real-time and non-real-time traffic. Furthermore, the fair-share method can be altered by misbehaving users.

Hindman et al. suggested using bandwidth shifting and redistribution at the gateway to provide QoS-guaranteed bandwidth. With this algorithm, the gateways are allowed to receive bids from mobile devices that are searching for bandwidth. The mobility of the nodes causes the proposed system to consider bandwidth shifting; however, this is insufficient to establish the QoS guarantee. The shifting of a mobile node from one gateway to another induces delay. Furthermore, the new gateway needs to know the bidding value of the previous gateways. Hence, this algorithm is unfeasible for a real-time environment.

To provide the SLA agreed-upon bandwidth in a cloud environment, the dynamic bandwidth allocator (DBA) was suggested by Nan Guofang et al. DBA manages bandwidth allocation according to the requirements of applications by dropping packets in a virtual machine. This process is implemented in the virtual machine at the driver level. Hence, the data of the virtual machine always cross the Internet links and routes that are far from the data center; such issues exert a significant effect on the system. Application-level feedback is employed to address resource allocation problems. However, application-level feedback is also inaccurate because of its dependence on cloud dynamics. Application-level feedback is supported by a fuzzy controller to measure the QoS metrics to control individual objectives. Fuzzy controllers are well suited for resources, such as the CPU and memory. However, bandwidth resources are under the control of a network, so providing a fuzzy controller induces network delay. Other network elements can also greatly contribute to the delay.

The virtual data center (VDC) is an architecture proposed for bandwidth guarantee within the data center. VDC involves a virtualization architecture called SecondNet, and it can be scaled by mapping virtual machines to physical machines; the hypervisor addresses the issues of routing and bandwidth reservations [12]. A small group of VM topology smoothly runs this algorithm for bandwidth allocation, but densely allocated VM negatively affects the bandwidth. The provider has to multiplex the VM to achieve suitable bandwidth allocation.

Cao et al. [13] suggested and demonstrated bandwidth management for data centers by using wavelength routing, which provides complete connectivity with reconfiguration and provisioning of bandwidth for hotspots. Bandwidth guarantee (BG) and time guarantee (TG) are the two reservation categories proposed to achieve an adaptive bandwidth allocation and pricing scheme. The system selects either BG or TG. If BG is selected, the bandwidth is guaranteed. TG provides an assurance of data transfer of the intended data size [14].

In [15], a VM allocation algorithm was proposed for BG in data centers. An online VM distribution method for the heterogeneous bandwidth demands of tenants was also proposed. The diverse bandwidth requirement of tenants is considered in the method, and tenants are given flexibility to assign different bandwidths for the user-opted application-specific VM. The sharing allocating switch buffer (SAB) for data centers was also proposed [16]. SAB is a transport protocol for bandwidth delay products (BDP). SAB possesses a congestion window to buffer the size of data flow in network equipment, such as switches. If the allocated bandwidth exceeds the required flow, then the bandwidth is fully utilized. Hence, SAB traffic will not cross more than the required flow; its bandwidth flow depends on the flow completion time [17]. Flow-level-based bandwidth allocation is a suitable technique for virtualization environments, such as data centers. In [18], flow-level bandwidth provisioning was used for the switches to reduce switching problems existing in the systems and enhance bandwidth assignment.

However, most of the flow-based bandwidth provisioning systems offer fine-grained allocation of bandwidth, but none of them achieved this. They also suffer in terms of implementation, exhibit poor performance, and are unable to accommodate the varying demand for bandwidth. The ability to offer a definite amount of bandwidth for tenants and their application is important to the development of data centers because predicting and offering bandwidth to the applications running in a data center are important. However, establishing suitable mechanisms for a multi-tenant environment, especially when applications have varied demands for bandwidth, is difficult for researchers.

3. FAIRLY SCHEDULED BANDWIDTH ALLOCATION SCHEME

Our fairly scheduled bandwidth allocation scheme, which is called FSB, is shown in Figure 2. For the proposed architecture to achieve bandwidth utilization for data centers, the traffic should be classified according to their requirements. Bandwidth should be partitioned and allocated such that it would never be underutilized. Hence, we designed a method to dynamically allocate bandwidth to applications by considering all the necessary fundamentals to efficiently utilize bandwidth.

Fig. 2 shows a typical data center with multi-tier functionality. The public interface allows end users to access the services provided by the data center. The aggregation layer allows the provider to set rules on the network traffic for hosted services. The aggregation module allows the administrator to monitor traffic by using intrusion detection systems and load balance on servers and by applying proper security patches for applications. The aggregation layer manages the policies of the access layer with respect to traffic flows. Furthermore, the aggregation layer is equipped with many networking modules, such as switches, load balancers, routers, firewall, security layers (e.g., SSL), intrusion detection systems, and network analyzers. An access layer is a place where all the physical machines are located and where services are hosted.

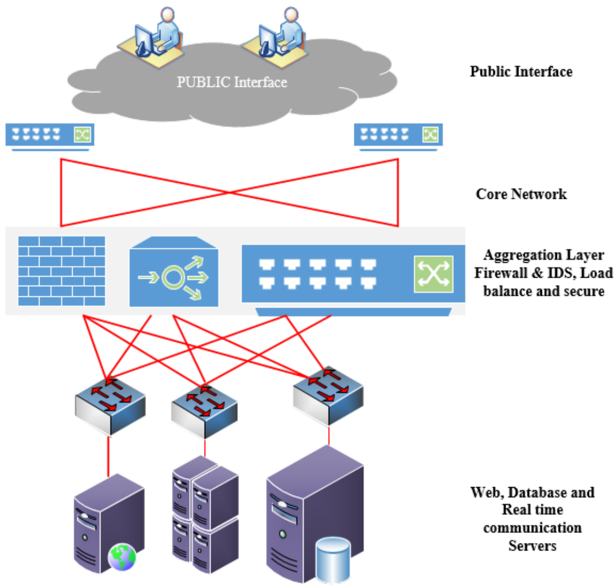

Fig.2 Unified Bandwidth Allocation Architecture

We explain application classification in the following section. We discuss bandwidth partitioning according to applications in Section 3.2. With all these modules, we build a dynamic bandwidth allocation scheme.

3.1 Classification of Applications

Users who opt for real-time services are grouped under one category because of their QoS and packetization requirements. Users who opt for non-real-time services are grouped under another category because non-real-time asymmetric applications generate best-effort traffic. Hence, we adopted the classifications shown in Table 1. Furthermore, the real-time traffic service classes need to be categorized again based on their usage of services, such as audio and video. Table 1 shows the QoS requirements of the three different traffic classes.

3.2 Bandwidth Classification

For a given data center with n nodes, connectivity graph G is a directed graph represented as (R, C), where R is the equipment of the aggregation layer, R = {1, 2, 3, 4…n}, and C is the connection between them, C = {i, j}. Links i and j exist between them. Each link has a bandwidth capacity Bcap (i, j), which is the SLA agreed-upon bandwidth, if a link exists between them. A set of services, S = {S1, S2...Sn}, uses the bandwidth according to the data center or SLA. Each application has two variables that are associated with the source and destination nodes in the data center. A flow, fb, is the bandwidth assigned to each link R. Hence, each application has a bandwidth inflow as follows.

$$f_b^{S_i} \geq 0 \qquad [1]$$

The following is the bandwidth for flow f within the data center, where Bcap is the bandwidth for services Si corresponding to the flow of bandwidth between the sources and the destination within the aggregation layer. Where Rn is the total number of equipment present in the data center. The total bandwidth partition to the data center can be written as follows:

$$\sum_{i,j \succ 0}^{n} f_{B_{cap}}^{Si}(i,j) \qquad [2]$$

$$\sum_{i,j \succ 0}^{n} f_{B_{cap}}^{si}(lr_{i,,j}) \cdot R_n \qquad [3]$$

In the inflow and outflow of bandwidth within the aggregation layer, the reserved bandwidth is RBi (Eq. 3 is written as RBi.). Our framework is then forced to split the available bandwidth into the same number of traffic classes, as mentioned in Table 1. Therefore, we must define the QoS requirements in terms of their acceptance rate and utilize the predefined bandwidth. We will discuss what happens when a traffic class requires more bandwidth than our framework can allocate dynamically later in the article.

Table 1. Classification traffic class with respect to the QoS requirement

Traffic Service Class (TSC)	QoS Requirement and Policy
Video Conference	High
Audio Conference	Moderate
WWW	Low

3.3 This paragraph is a repeat of 3.1

Although partitioning the bandwidth (QoS and non-QoS) is a potential approach to provide guaranteed QoS for traffic service classes, it also uses bandwidth inefficiently when the expected traffic varies with time. For example, if the traffic is more than the expected amount at the QoS-aware bandwidth partition, then the blocking rate of that particular bandwidth portion will increase until it reaches a threshold. If the non-QoS bandwidth partition has a few users, then the entire system will be underutilized. Therefore, to increase the bandwidth utilization, the system should accept new requests (disregarding traffic service classes) to use portions (high, moderate, or low) of the bandwidth.

An approach is to allow the system to accept bandwidth requests from different traffic service classes. However, this approach will flood the bandwidth partition with requests that are not native. This problem can be solved by dividing the bandwidth partition into two sectors: marked bandwidth for the native traffic service class and a commonly utilized bandwidth sector for the non-native traffic service class. The marked RBi and the common area SBi can be presented as follows, by using [18].

$$R_{bi} = \beta \cdot B_i, \quad [4]$$

$$S_{bi} = B_i - \beta$$
$$\therefore$$
$$= (1-\beta)B_i, \quad [5]$$

Where β is the reservation parameter that takes a value between 0 and 1. If β increases, the SBi area decreases; when β is equal to 1, SBi is 0. Meanwhile, if β decreases, the SBi area increases; it becomes equal to Pi only when β is equal to 0. Then, the bandwidth can accommodate native as well as non-native traffic service classes. Each node R's shared bandwidth can be obtained by rewriting the equation above.

$$(1-\beta)\cdot\left(\sum_{i,i,j\succ 0}^{n} \mathrm{F}^{si}{}_{B_{cap}\left(lr_{i,,j}\right)} \cdot R_n\right) \quad [6]$$

3.4 Bandwidth Allocation Scheme

The shared partition can service both native and non-native traffic service classes, but the reserved partition is only for native classes. Therefore, we have two options for bandwidth allocation. First, we allocate the shared bandwidth to the non-native traffic service class. Second, we accommodate native traffic in the reserved bandwidth partition. Fig. 3 shows the bandwidth allocation scheme. At time t0, non-native traffic class NSi arrives with a request to use the bandwidth. At times t1 and t2, native and non-native traffic arrive and are allowed to use the shared bandwidth. After time t2, any new non-native traffic will be blocked and suspended until the shared bandwidth is freed by the traffic in the shared partition. To further enhance bandwidth utilization, we combine the bandwidth from the available shared and reserved partitions; hence, we can reduce bandwidth fragmentation. In the previous sections, we explained the framework of our proposed scheme to accommodate traffic service classes. In this section, we present an algorithm that uses our dynamic bandwidth allocation process to schedule bandwidth to accommodate traffic. If all these modules fail to accommodate the request, then it will be blocked until the bandwidth is released and free to use.

4. PERFORMANCE EVALUATION

We used the SimPy tool for our proposed method (https://simpy.readthedocs.org/en/latest/). The performance of our scheme was analyzed in conjunction with commonly used schemes, such as the general allocation scheme (GAS) and the general partition (GP) scheme. GAS allows all traffic service classes to access the bandwidth with reservations for real-time traffic. The GP bandwidth allocating scheme does not reserve any bandwidth and continues allocating to the traffic that arrives at the data center. To analyze the real-time provisioning of our proposed scheme, we considered the following in Table 2. Text (browsing) is regarded as a non-real-time application, and its portion in the bandwidth partition was set to a very small amount (e.g., 10%). The bandwidth for a video-based conferencing system requires needs to be large to provide QoS. Its bandwidth partition was set to 50% of the partition, and the remaining bandwidth partition was assigned to audio-based conferencing systems.

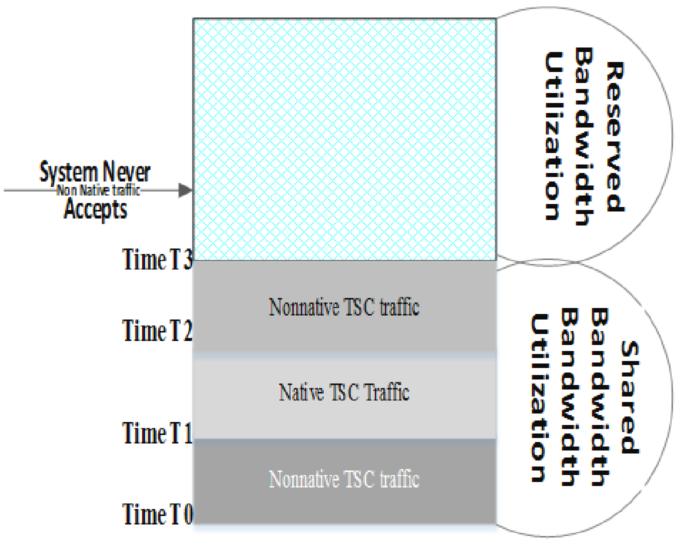

Fig 3. Bandwidth allocation process in shared and reserved bandwidth partitions

Table 2. Simulation Parameters

Traffic Class	Required Rate	Service Level Agreement
Video	1.5 Mb/s	Highest
Audio	95 kb/s	Moderate
Text	50 kb/s	Low

As shown in Fig. 4 and Fig. 5, the blocking probability of GAS and FSB for real-time communication is consistently less than or equal to 0.01. Given that the QoS parameter for audio over the network is very small in terms of data rate and packetization, the bandwidth reservation employed by FSB and GAS is for real-time traffic. GP has always shown a high blocking property for real-time traffic; in particular, the blocking probability is higher in video-based communication than in audio-based communication. During text-based communication, GAS and GP have a higher blocking probability rate than our FSB because FSB uses dynamic bandwidth utilization.

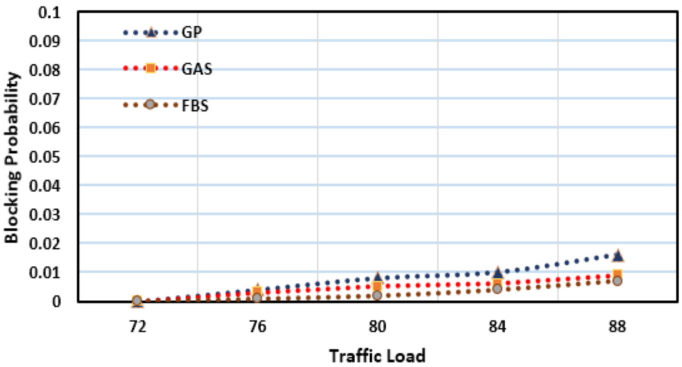

Fig. 4. Blocking Probability of Audio based activities

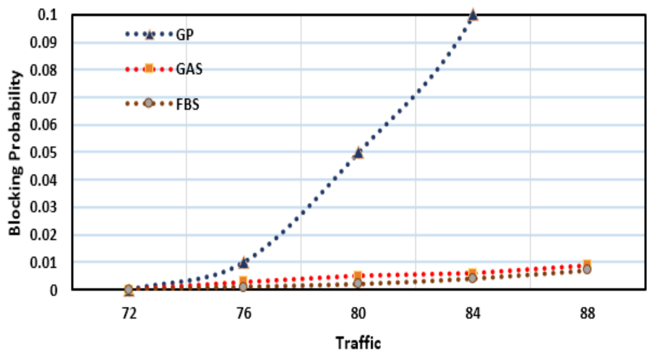

Fig. 5. Blocking Probability of Video based activities

Fig. 6 shows the text-based retrieval graph. The bandwidth for text-based retrieval is consumed by the real-time traffic classes in the GAS and FSB systems; however, FSB shows better performance than GAS. As expected, GP shows a low probability of blocking in text-based retrieval. FSB and GAS perform well in QoS-based traffic, and we noted that FSB performs better than GAS under a normal traffic load.

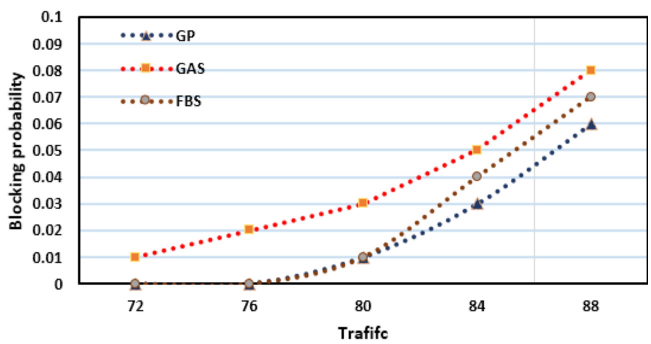

Fig.6 Blocking Probability of Text based activities

To investigate the advantages of FSB over GAS, we observed their performance under different traffic loads. By varying the values of the reservation parameter, we obtained different reservation partitions for the bandwidth. Thus, the text-based traffic class' blocking probability was also reduced. In this case, we only tried video and text communication traffic, which can be accomplished when the reservation parameter is 0.5. This reduces the blocking probability of text-based communication and thus increases bandwidth utilization, as shown in Fig.7 and Fig.8.

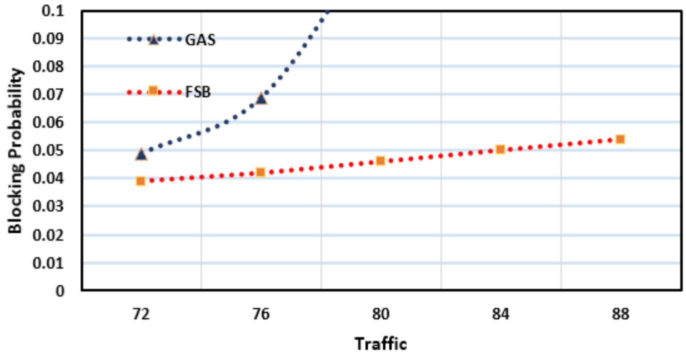

Fig. 7 Blocking Probability of Text based activities

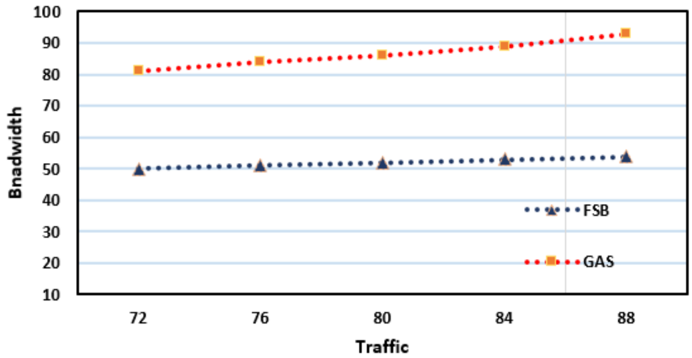

Fig. 8. Utilization

As the reservation parameter reduces the commonly shared part of the reserved bandwidth, the blocking probability increases for real-time communication and decreases for non-real-time traffic. To test these conditions, we varied the values of the reservation parameter. We conclude that when confronting small variations of traffic load configurations, FSB demonstrates better performance than GAS in terms of blocking probability and capacity utilization. Bandwidth partitioning is a solution to satisfy the requirements of real-time and non-real-time traffic. We selected data centers only because they are completely isolated from the main Internet network. Thus, their policies are restricted only to themselves. However, a data center is large and requires many administrators to manage and apply policies to sustain the bandwidth-striving traffic. We suggest using bandwidth partitioning and applying policies to separate the real-time and non-real-time applications so as to maintain a minimum blocking probability and increase the bandwidth utilization.

5. CONCLUSION

To solve the bandwidth assignment problem, we developed a novel fairly scheduled bandwidth allocation scheme for data center traffic. In our approach, the bandwidth provision of the aggregation layer is exploited. The simulation results and analysis indicate that if the bandwidth is managed well according to the needs of the applications, the aggregation layer can overcome the overall performance of the applications or services by using the dynamic bandwidth provisioning system.

6. REFERENCES

[1] Divakaran DM, Hegde S, Srinivas R, Gurusamy M. Dynamic resource allocation in hybrid optical–electrical datacenter networks. Computer Communications. 2015 Sep 15;69:40-9.

[2] Hindman B, Konwinski A, Zaharia M, Ghodsi A, Joseph A, Katz R, Stoica I. Mesos: A platform for fine-grained resource sharing in the data center. In: USENIX 2011 Networked Systems Design and Implementation; 1 April 2011; Boston, Massachusetts, USA: USENIX. pp. 22-22.

[3] Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture. ACM SIGCOMM Computer Communication Review. 2008 Oct 1;38(4):63-74.

[4] Farrington N, Alexey A. Facebook's data center network architecture. In: IEEE 2013 Interconnects Conference; 5-8 May 2013; Santa Fe, New Mexico: IEEE. pp. 5-7.

[5] Benson T, Akella A, Maltz D. Network traffic characteristics of data centers in the wild. In: ACM 2010 SIGCOMM Internet Measurement; 1-3 November 2010; Melbourne, Australia: ACM. pp. 267-280.

[6] Lee J, Yoshio T, Myungjin L, Lucian P, Sujata B, Joon-Myung K, Puneet S. Application-driven bandwidth guarantees in datacenters. In: ACM 2014 SIGCOMM Conference; 19-21 August 2014; Chicago, Illinois, USA: ACM. pp. 467-478.

[7] Nan G, Zhifei M, Mei Y, Minqiang L, Honggang W, Yan Z. Stackelberg game for bandwidth allocation in cloud-based wireless live-streaming social networks. IEEE Syst J 2014; 1: 256-267.

[8] Ghosh P, Kalyan B, Sajal D. A game theory-based pricing strategy to support single/multiclass job allocation schemes for bandwidth-constrained distributed computing systems. IEEE T Parall Distr 2007; 18: 289-306.

[9] Ye Z, Yong L, Guang S, Depeng J, Li S, Lieguang Z. Game theory based bandwidth allocation scheme for network virtualization. In: IEEE 2010 Global Telecommunications Conference; 7-9 December 2010; Miami, Florida, USA: IEEE. .pp. 1-5.

[10] Di N, Chen F, Baochun L. A theory of cloud bandwidth pricing for video-on-demand providers. In: IEEE 2012 INFOCOM; 25-30 March 2012; Orlando, Florida, USA: IEEE. pp. 711-719.

[11] Kasbekar S, Saswati S. Spectrum pricing games with bandwidth uncertainty and spatial reuse in cognitive radio networks. In: ACM 2010 International Symposium on Mobile Adhoc Networking and Computing; 20-24 September 2010; Chicago, Illinois, USA: ACM.

[12] Guo C, Lu G, Wang HJ, Yang S, Kong C, Sun P, Wu W, Zhang Y. Secondnet: a data center network virtualization architecture with bandwidth guarantees. InProceedings of the 6th International COnference 2010 Nov 30 (p. 15). ACM.

[13] Cao Z, Proietti R, Clements M, Yoo SB. Experimental Demonstration of Flexible Bandwidth Optical Data Center Core Network With All-to-All Interconnectivity. Journal of Lightwave Technology. 2015 Apr 15;33(8):1578-85.

[14] Divakaran DM, Gurusamy M. Towards flexible guarantees in clouds: Adaptive bandwidth allocation and pricing. Parallel and Distributed Systems, IEEE Transactions on. 2015 Jun 1;26(6):1754-64.

[15] Li D, Zhu J, Wu J, Guan J, Zhang Y. Guaranteeing heterogeneous bandwidth demand in multitenant data center networks. Networking, IEEE/ACM Transactions on. 2015 Oct;23(5):1648-60.

[16] Zhang J, Ren F, Yue X, Shu R, Lin C. Sharing bandwidth by allocating switch buffer in data center networks. Selected Areas in Communications, IEEE Journal on. 2014 Jan;32(1):39-51.

[17] Jin H, Pan D, Liu J, Pissinou N. Openflow-based flow-level bandwidth provisioning for cicq switches. IEEE Transactions on Computers. 2013 Sep;62(9):1799-812.

[18] Skoutas DN, Makris P, Skianis C. Optimized admission control scheme for coexisting femtocell, wireless and wireline networks. Telecommunication Systems. 2013 Jul 1;53(3):357-71.