A STUDY ON APPLICATION-TOWARDS BANDWIDTH GUARANTEE BASED ON SDN

Cao Shaohua[1*], Tong Mengzhu[1], Zhihan Lv[2], Dingde Jiang[3]
1. College of computer and communication engineering, China University of Petroleum, Qingdao, China
2. SIAT, Chinese Academy of Science, China
3. Northeast University, China
*shaohuacao@upc.edu.cn

***Abstract*—With the rapid development of cloud computing and big data technologies, there are more and more applications in data centers. In order to ensure the users' experience and the service providers' revenue, how to guarantee application-bandwidth efficiently is attracting a cloud of research attention. Based on the advantages of SDN which separates controlling from forwarding, this paper investigated the application-towards bandwidth guaranteed method. Firstly, this method performed coarse granularity classification of the various applications, then carried out the ATBG (Application-towards Bandwidth Guarantee) algorithm, and realized the application-bandwidth guarantee efficiently in multiple priority levels for various applications through a centralized control. This paper evaluated this method in the floodlight & mininet simulation environment.**

Keywords—SDN; OpenFlow; Data center; Guaranteed Bandwidth;

I. INTRODUCTION

Nowadays, Internet has become an important infrastructure, it affects day life, such as E-mail, shopping online, social contact App, online video. These applications are growing in an exploding speed. Network applications make daily life colorful greatly. On the other hand, they bring a great challenge on network management. As users', they hope they can get high priority, high quality, constant services to support their jobs run successfully. They require high quality network. From service providers' perspective, they should try their best to meet every user's requirement, such as end to end delay, dither, packets loss rate etc. This is related to users' experience evaluation and will have an influence on providers' profits. Therefore, doing researches on how to guarantee application service quality make much sense.

Researches on application services QoS are the focus of attention in both of academic and industrial fields [1-2]. Contributions include technical improvement and innovation in data link layer, transport layer and application layer. Relevant evaluation standards include bandwidth, delay, dither, packages loss rate, bit error rate etc. Given that network application service is a king of resource, then the researches on application services QoS are resources optimization and try to take into account the interests of all parties. Research content can be divided into four types [3] abstractly. The first one is resource allocation such as link bandwidth and queue space allocation etc. Second one is job scheduling, making multi-jobs run orderly and efficiently on finite physical resources. Third one is network resources deployment, for example, how to use minimum cost to deploy servers in clusters. The last one is network parameter settings, such as setting reasonable congestion windows for different systems. In particular, application service bandwidth guarantee can improve users' QoS experience directly. However, the distribution of Internet leads to high cost to implement this method. The intensive management and regular structure of the data centers network can implement the method at low cost. Hence, with the rapid development of cloud computing and the improvement of data centers' state, doing researches on how to guarantee application service quality is significant.

Supported by Outstanding young scientist research award fund program of Shandong Province. (BS2014DX021)

In order to guarantee link bandwidth efficiently under data centers network environment, the paper introduces an SDN (Software Defined Network) technique. By means of SDN's network architecture, the paper allocates and schedules link bandwidth globally. There are two contributions: one is dividing the applications into three categories based on real bandwidth requirements, different category has different priorities. The other is ATBG (Application-towards bandwidth guaranteed) algorithm that can guarantee link bandwidth efficiently, intensively according to different applications.

The structure of this paper is organized as follows. Section II describes SDN technique and the research status about data centers bandwidth guarantee. Section III introduces how to guarantee application-oriented bandwidth in detail. Section IV is about implementation and evaluation. Finally, the last section summarizes the whole work and outlines the future work.

II. BACKGROUND AND RELATED WORK

A. SDN Network Architecture

SDN comes from Stanford University Clean Slate research group in America. This technique separates the controlling function from forwarding function in traditional network devices and improves the network protocols innovations [4]. It also solves many problems about maintenance and extension in traditional IP network. If the application changes, the network devices need to change at the same time in traditional network. This is a complicated process. SND separates the controlling function from network devices and achieves the intensive management and controlling without dependency on bottom network devices.

978-1-5090-2482-7/16/$31.00 ©2016 IEEE

SDN architecture [5,6] can be classified three layers: application layer, controlling layer and data layer, as shown in Fig. 1. The north interface communicates with application layer, customizes the controlling layer settings and achieves programmable controlling. The south interface communicates with data layer, updates the infrastructure status and issues the forwarding strategies, the representative protocol is OpenFlow.

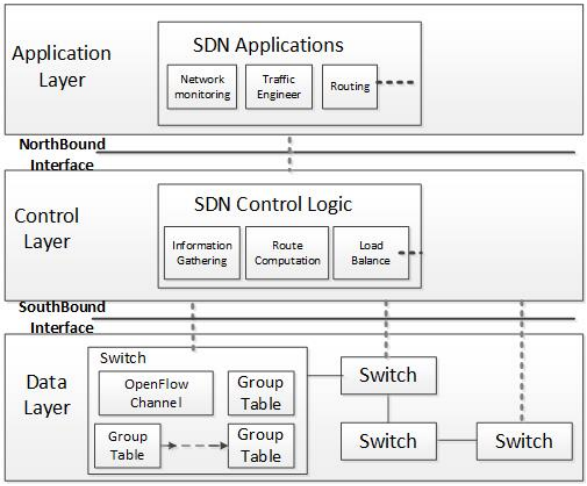

Fig. 1. SDN Network Architecture

Controlling layer is the core of whole architecture [7-10], it centralizes all of the network nodes' controlling function logically, including collecting streams information, routing calculating and preserving north and south interfaces. Based on the programmable interfaces in controlling layer, application layer provides all kinds of network services, such as monitoring network status, and traffic engineering. According to the flow table controlling layer issues, application layer forwards data packages. Because of the flexibility of SDN architecture, our method guarantee bandwidth for different applications and different priorities through intensive link resources management.

B. Current Researches on Bandwidth Guarantee

There are many researches on application services bandwidth guarantee problems in academic field. The scenarios include Ad Hoc network, wireless network, MAN (Metropolitan Area Network), data center network. Along with the impacts of cloud computing and big data, researches on data center network application bandwidth guarantee is becoming a hot spot.

Literature [11] presents a virtual network architecture over cloud computing data centers. In order to guarantee the bandwidth between network application VM-Pairs' communication, this architecture uses VDC (virtual data center) Manager to manage link bandwidth intensively. But it doesn't guarantee bandwidth according to different applications and thus leads to a heavy burden to VDC Manager. Literature [12] improves the reliability of bandwidth guarantee algorithm to increase the cloud computing data centers' physical resources ratio, and evaluates performances if the bandwidth requirements are different. But it doesn't take application categories into consideration, either. Literature [13] considers different application communication models has influences on bandwidth guarantee algorithm performance, it changes the application bandwidth requirement description and optimizes the algorithm. However, as above two papers, it doesn't take application categories into consideration. Literature [14] pays more attention to application difference model Diffserv and studies the principle of queue scheduling algorithm and improves the weighting loop algorithm WRR. It realizes the goal of higher priority higher bandwidth guarantee. But this method distinguishes applications excessively and increases the burden on bandwidth guarantee system.

This paper makes coarse granularity classification of the various applications to solve above problems. And with the help of intensive controlling under SDN, it presents ATBG algorithm to realize bandwidth guarantee efficiently

III. APPLICATION-TOWARDS BANDWIDTH GUARANTEE BASED ON SDN

There are three parts to support application-towards bandwidth guarantee under SDN: intensive management structure of SDN, multi-application management according to classification and ATBG algorithm. To clarify this method clearly, it will be described as follows.

A. Architecture Structure

SDN technique separates controlling from forwarding, and increases the flexibility and agility of network management greatly. The bandwidth guarantee method this paper provides is mainly deployed on SDN controller, core module consists of application classification module and routing module, as shown in Fig. 2. Application classification module achieves coarse granularity classification of the various applications. Routing module runs the ATBG algorithm based on application category and bandwidth requirement and achieves the goal of application data stream bandwidth guarantee.

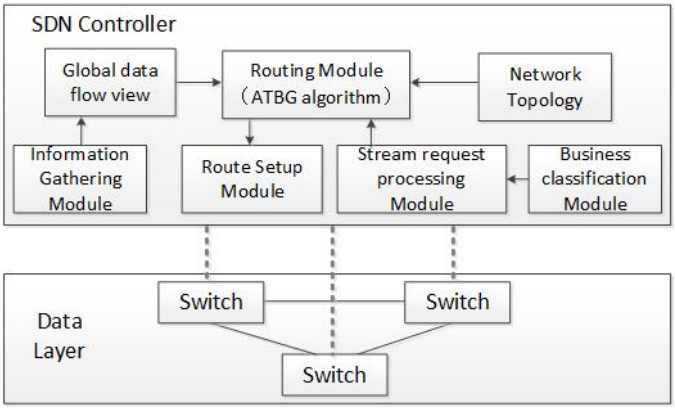

Fig. 2. System architecture

Firstly, the stream requirement processing module in controller accepts the bottom requirements. Then based on the classification standard in the application classification module, decide the category the application belongs to and the decision is as the inputs of ATBG algorithm in routing module. Thirdly, routing module runs the ATBG algorithm and allocates links for data stream and guarantee the bandwidth requirements. If

the bandwidth requirements are not met, the request will be rejected. Finally, the controller updates the whole network stream graph periodically according to stream information gathering function OpenFlow protocol provides. If finding that the application streams cannot be guaranteed, it will run ATBG algorithm again to plan the transport path.

B. Multi-application Classification Management

There are all kinds of applications running on the data centers, because of the finite physical resources, the problem complexity and resolution cost will increase greatly if every application is guaranteed real time. In order to reduce the complexity of the problem and the cost, based on bandwidth requirement and delay, the paper divides multi-application into three categories coarsely.

- Level 0: applications are sensitive to bandwidth and delay. These application need high priority bandwidth guarantee to ensure users' experience.
- Level 1: applications are sensitive to bandwidth but not delay. Because of the low sensitivity of delay, these applications have the second highest priority to guarantee bandwidth, when the bandwidth isn't enough, it will buffer the data stream.
- Level 2: applications belong to neither level 0, nor level 1. Because they don't have sensitivity to bandwidth, they have lowest priority.

C. ATBG Algorithm

The core of application-towards bandwidth guarantee method under SDN is routing module, and the core of routing module is ATBG algorithm. This algorithm achieves to guarantee bandwidth according to different applications. The inputs include application type (type), application bandwidth requirement (bw), source address and destination address.

Algorithm : Application-toward Bandwidth Guarantee

```
1   The following procedure runs on the SDN controller independently.

2   procedure Get_largertypeflows (int type, Path path)
3     allflows=getallflows(path)
4     for(flow in allflows)
5       if(flow.type<type)
6         flows.add(flow)
7   end procedure

8   procedure Search_flowtodelete(Collection sd_paths, int type)
9     for(path in sd_paths)
10      init total_bw and total_flows
11        flows= Get_largertypeflows (type, path).sortby _type_bw()
12      for(flow in flows)
13        total_bw+=flow.bw
14        total_flows.add(flow)
15        if(total_bw>bw)
16          return < path, total_flows>
17  end procedure

18  procedure Alloc_path(int type, bandwidth bw, Address src, Address dst)
19    sd_paths=getallpaths(src, dst)
20    for(path in sd_paths)
21      if(the remaining bw of path >= bw)
22        return path
23    if(type==0)
24          <path, total_ flows>=Search_flowtodelete(sd_paths,1)
25      if(<path, total_flows >==null)
26          <path, total_ flows>=Search_flowtodelete(sd_paths,0)
27        Endif
28      Else
29          <path, total_ flows>=Search_flowtodelete(sd_paths, type)
30      if(<path, total_flows >==null)
31        refuse to provide service
32        Return
33      Else
34        cease flows of <path, total_flows>
35        return path of <path, total_flows>
36  end procedure
```

Under data centers environment, there are many paths between end and end. Line 19 first gets all possible paths between the source and the destination. Then check the paths whether the left bandwidth can meet the requirement, if can, allocate this path to new job, as shown Line 20~22. If not, check the paths one by one and delete the low priority data stream to release resources for high priority stream, as shown line 23~29. The detail description is: if type=2, the reject to allocate bandwidth due to lowest priority. If type=1, then check the paths one by one and delete the lower priority jobs so that there is enough bandwidth for new application. If type =0, then try to delete type=2 data stream, if it isn't enough, then try to delete type=2 and type=1 data stream to vacate enough bandwidth. Line 30~35 is to judge whether it finds the lower priority data stream to delete, if do, then stop the stream and return its path, if not, then reject to distribute path for new job because of lacking of resource.

IV. EVALUATION AND ANALYSIS

A. Experiment Environment

To complete the data center simulation, we deployed Ubuntu 13.10 virtual machine under VMware Workstation, installed Mininet [15] in Ubuntu and built custom topology in Mininet.

1) Build Network Topology

Open the Ubuntu terminal, input commands: sudo mn -custom ~/Final/topo-final.py -topo mytopo --controller=remote, ip=192. 168.37.154, port=6653 –mac. In particular, "topo-final.py" is the custom topology script file to realize the network topology structure.

The custom data center simulates the topology and builds a 3-switch and 3-host structure, as shown in Figure 3. One host is used for outside user in data center and other two are used for two servers. The servers are connected to access switches and outside user is connected to aggregation switch. In addition, to validate the method this paper provides, every port of OpenFlow switch must keep five queues for different applications scheduling.

2) Experiment Environment Deployment

After building the custom topology in Mininet, use VLC and vsftpd to deploy a video server and an FTP server on h2 and h3 nodes. Open different host nodes in Mininet by "Xterm" command, due to Mininet features, every host node can be regarded as an independent, using same system and kernel host, so it can provide services.

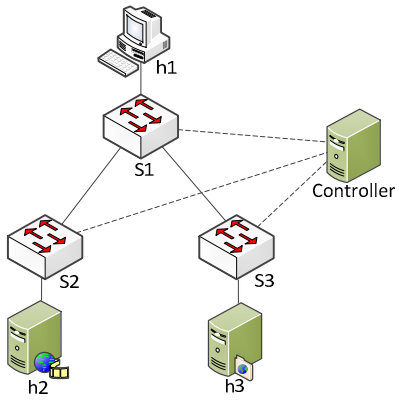

Fig. 3. Experiment Environment Network Topology

Steps to deploy video server on host h2: open the terminal of h2 and the VLC software, select "Media" then "Stream" in the VLC menu, then build a stream media service based on HTTP protocol and 8800 port.

Steps to deploy FTP server on host h3: open the terminal of h3 and execute commands "vsftpd /etc/vsftpd.conf &" and "start vsftpd".

B. *Practical Test*

Add a new module to controller Floodlight [16,17] to support application bandwidth guarantee. In order to carry out the comparative analysis of the test results and draw a conclusion, run test module program under bandwidth guarantee function and without this function respectively. After editing and modifying the Floodlight controller, type "ant" command to compile and deploy, and execute "java -jar target/floodlight.jar" command to open Floodlight controller.

1) Switch Multi-port Queue Query

To check the scheduling results of switch port five queues, take switch S1 as an example, Fig. 4 shows the five queues of its port.

```
neo@ubuntu:~/Final$ sudo ovs-ofctl queue-stats s1
OFPST_QUEUE reply (xid=0x2): 15 queues
  port 3 queue 0: bytes=19206, pkts=89, errors=0
  port 3 queue 1: bytes=0, pkts=0, errors=0
  port 3 queue 2: bytes=0, pkts=0, errors=0
  port 3 queue 3: bytes=0, pkts=0, errors=0
  port 3 queue 4: bytes=0, pkts=0, errors=0
  port 1 queue 0: bytes=19533, pkts=98, errors=0
  port 1 queue 1: bytes=0, pkts=0, errors=0
  port 1 queue 2: bytes=0, pkts=0, errors=0
  port 1 queue 3: bytes=0, pkts=0, errors=0
  port 1 queue 4: bytes=0, pkts=0, errors=0
  port 2 queue 0: bytes=14644, pkts=77, errors=0
  port 2 queue 1: bytes=0, pkts=0, errors=0
  port 2 queue 2: bytes=0, pkts=0, errors=0
  port 2 queue 3: bytes=0, pkts=0, errors=0
  port 2 queue 4: bytes=0, pkts=0, errors=0
```

Fig. 4. The result of queue information

According to the query results, default packets such as LLDP packet will be forwarded through port queue Q0 without forwarding strategies or data center services. So all of the applications will be forwarded through Q0 except video and FTP applications. There is no impacts on application stream and information gathering in the module.

2) Bandwidth Guarantee Function Test

a) Test Function Without Bandwidth Guarantee

Modify the bandwidth guarantee module in Floodlight controller and make it do not guarantee bandwidth requirement. When data center user h1 requests video service and FTP service at the same time, there is obvious waiting process and blocking phenomena in video service requirement due to no bandwidth guarantee. The stream ratio information on video and FTP service is shown as Table 1.

TABLE I. SERVICE STREAM RATIO WITHOUT BANDWIDTH GUARANTEE

service	Stream ratio（TXBytes）/Bps					
Video	48448.0	49457.3	33812.7	47438.7	54504.0	37345.3
FTP	75700.0	83270.0	85288.7	70653.3	71158.0	90840.0

b) Test Function With Bandwidth Guarantee

Modify the bandwidth guarantee module in Floodlight controller and make it guarantee bandwidth requirement. When data center user h1 requests video service and FTP service at the same time, compare the results with no bandwidth guarantee results, there are obvious improvements in video playing effects and fluency. The stream ratio information on video and FTP service is shown as table 2.

TABLE II. SERVICE STREAM RATIO WITH BANDWIDTH GUARANTEE

Service	Stream ratio（TXBytes）/Bps					
Video	59685.3	35104.3	79232.7	90012.7	77174.0	61794.3
FTP	0.0	84830.7	16149.3	31289.3	25738.0	31289.3

C. *Experiment results analysis*

When there is no video bandwidth function, service stream ratio table shows us the servers try their best to provide services because they do not have limits and strategies to any services. However, according to the service property and server performance differences, FTP service which does not need real time occupies the bandwidth mostly. As a result, there is fewer bandwidth for video service, users cannot get fluent video service and achieve the network performance requirement. The ration information broken line graph is shown on Fig. 5.

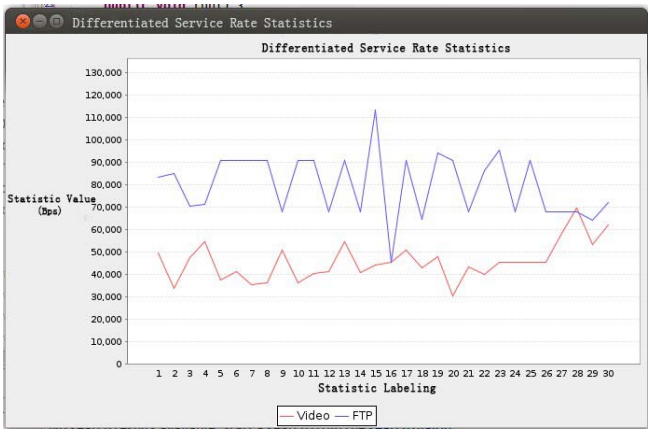

Figure.5. Ration Information Broken Line Graph without Bandwidth Guarantee

When there is video bandwidth function, service stream ratio table shows us evident changes on the service ratio because the system sets strategies and limits on video and FTP service. When system gets the FTP service stream ratio is greater than 80000.0Bps, the video service bandwidth is too low to provide good service, so restrict FTP ratio and guarantee video service. We can conclude from the graph that if the FTP ratio gets 84830.7Bps, then there is restriction on FTP ratio and the ratio is maintained around 20000~30000Bps. The video ratio increases to 70000~80000Bps from 35104.3Bps. So the video bandwidth is guaranteed. The ratio is shown as Fig. 6.

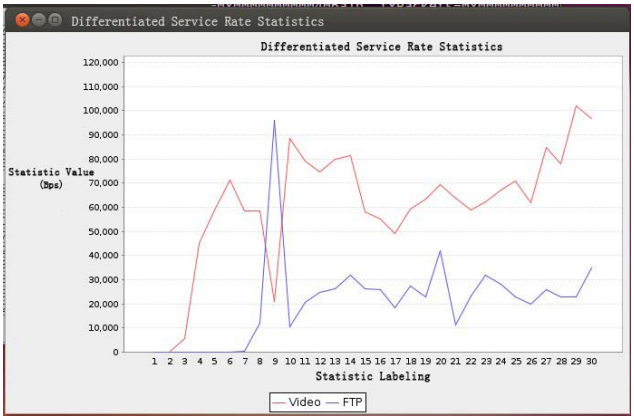

Figure.6. Ration Information Broken Line Graph with Bandwidth Guarantee

Compare the two broken line graphs, then understand the impacts Floodlight controller have on stream ratio under two different status. While the bandwidth guarantee module in controller uses the application classification bandwidth guarantee function, if there are video service and FTP service at the same time, controller guarantee the video bandwidth first. When the FTP service ratio is too high (given value is 80000Bps), restrict the FTP service bandwidth so that improve the video bandwidth occupation. Thus, if there is congestion on the network, video service will be guaranteed first.

We also can observe the network topologies and switches information through Floodlight controller home page (Localhost：8080). Information about aggregation switches, video service and FTP service stream is shown as Fig. 7. Domain matching and instructions are the same as settings in module (enqueue:1 represents the queue which enters switch port 1). And there is data forwarded according to these two flow table, this indicates that flow table executes its tasks and achieves the functions.

0	0x0	32768	in_port=2 eth_type=0x800 ip_proto=6 tcp_src=8800 ipv4_src=10.0.0.2 ipv4_dst=10.0.0.1	enqueue:1	n/a	n/a	n/a	n/a	n/a	n/a	61	123531	1	60
0	0x0	32768	in_port=3 eth_type=0x800 ip_proto=6 tcp_src=20 ipv4_src=10.0.0.3 ipv4_dst=10.0.0.1	enqueue:1	n/a	n/a	n/a	n/a	n/a	n/a	1	65226	1	60

Figure.7. Aggregation Switch Flow Table Information

While we should consider: when there are video service and FTP service on the network at the same time, video service will be guaranteed. However, if the video service ends, FTP service stream rate won't be restricted any more. The real test results are shown as Fig. 8.

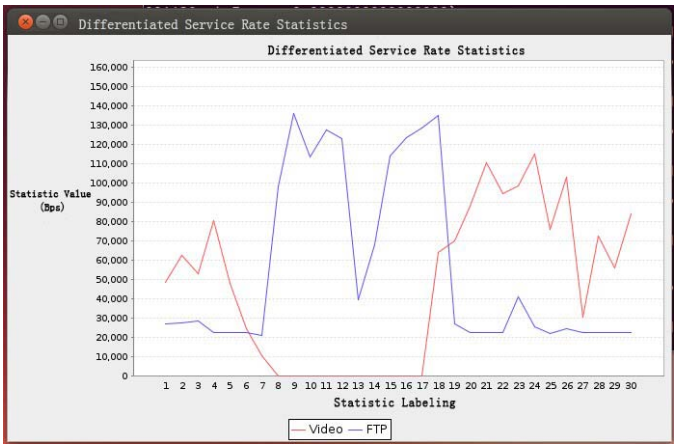

Figure.8. Video Service Stop/Continue Ratio Broken Line Graph

The broken line graph indicates: if video service is paused, the FTP service stream ratio increases again and there is no controller bandwidth guarantee strategy limit. When video service continues, its bandwidth will be guaranteed first. In conclusion, application classification guarantee module can realize its functions successfully.

V. SUMMARY

From the experiment results and data analysis, SDN technique can guarantee different service bandwidth, and resolve the important service guarantee problem in data centers. We can monitor and manage the data centers network by SDN controller bandwidth guarantee module. When there are many applications on the network, network may congest and important packets would be lost because of link capacity etc. We need to dispatch data streams, to set ports and queues, to issue flow table and adjust forwarding strategies, and to guarantee important service bandwidth first so that the network concentrates on bandwidth guarantee. It is different from the QoS module and other modules complexity on functions and performance. This improves the handling ability and efficiency of whole network. In the meantime, SDN technique features improve the horizontal scalability of data center network topology. From physical devices, we just need enlarge the OF island data center monitoring and control. In addition, it is easier to deploy bandwidth guarantee function in data centers using SDN than using traditional distribution controlling.

Bottom switches will forward OpenFlow Message to controller once they get the packets. Controller analyzes the OpenFlow Message and make sure which service it belongs to. At the same time, controller updates the switches information positively by multi-thread so that it can decide which strategy suits for the service. Based on this, controller issues the flow table to switches and depends on flow table items to forward data center network packets and realizes application-toward bandwidth guarantee. From the usage and experiment results of controller module, it is feasible to guarantee application classification bandwidth through SDN technique.

REFERENCES

[1] Lin C, Wang Y Z, Ren F Y. Research of New generation Network QoS. Chinese Journal of Computers，2008.9，Vol.31.No.9：pp.1525-1535

[2] Li Y Z, Sun M， et al. The design and implementation of QoS management system based on OpenFlow Technology. Journal of Guangxi University: Natural Science Edition， 2011.10， Vol:36 supplement extra edition 1:pp.42-46.

[3] Lin C, Li Y, Wan J X. Optimization Approacher for QoS in Computer Networks: A Survey. Chinese Journal of Computers. 2011, Vol. 34, No.1:pp. 1-12.

[4] Zhou T Q, Cai Z P, Xia J, Xu M. Traffic Engineering for Software Defined Networks. Journal of Software. 2016,27(2)：pp. 394-417.

[5] Gustavo A. A. Santana, Date Center Virtualization Fundamentals. Beijing：Posts and Telecom Press,2015.2

[6] Zhang S M, Zou F M. Survey on Software Defined Network Research. Application Research of Computers. 2013, 30(8):pp.2246-2251.

[7] Zhang W F. Depth resolution SDN: Benefit, Strategy, Technology, Practice. Beijing: Publishing House of Electronics Industry,2014.1.

[8] Jean Tourrilhes, Puneet Sharma, et al. SDN and OpenFlow Evolution: A Standards Perspective. IEEE CS, 2014.12:pp.22-29.

[9] Li，C，OpenFlow1.3 Study Guide. http://www.muzixing.com/pages/2014/08/10/openflow13xue-xi-bi-ji.html，2014.8

[10] Thomas D.Nadeau, Ken Rray, SDN: Software Defined Networks [M].Beijing: Posts and Telecom Press,2014.5.

[11] Guo C, Lu G, Wang H J, et al. Secondnet: a data center network virtualization architecture with bandwidth guarantees[C]//Proc of the 6th International CoNEXT. New York: ACM, 2010.

[12] Zhu J, Li D, Wu J, et al. Towards bandwidth guarantee in multi-tenancy cloud computing networks[C]//Network Protocols (ICNP), 2012 20th IEEE International Conference on. IEEE, 2012: pp.1-10.

[13] Jeongkeun Lee, Yoshio Turner, et al. Application-Driven Bandwidth Guarantees in Datacenters[C]. SIGCOMM, 2014.

[14] Yang S E. Research on dynamic bandwidth allocation algorithm for multi service networks._Communication technology, 2011, 44(4):pp.82-84.

[15] Sdnhub. Mininet Walkthrough http://sdnhub.cn/index.php/mininet-walkthrough-chinese/, 2014.6.

[16] David Erickson, The Beacon OpenFlow Controller. http://conferences.sigcomm.org/sigcomm/2013/papers/hotsdn/p13.pdf, 2013

[17] KC.Wang.Floodlight Document[M]. https://floodlight.atlassian.net/wiki/display/floodlightcontroller/Floodlight+Documentation