SDN over IP: Enabling Internet to Provide Better QoS Guarantee

Chao Hu[①], Qun Wang[②] and Xiuyue Dai[②]

[①](Institute of Command Information Systems, PLA University of Science and Technology, Nanjing 210007, China)

[②](College of Information Science and Technology, Zhejiang Shuren University, Hangzhou 310015, China)

Email: {huchaonj@126.com, Qun.w@263.net, studdxy@126.com}

***Abstract*—Many multimedia applications and online services have risen on the Internet, and they demand high quality of service for their real-time interaction. Although DiffServ has been employed in the network core, it neither provide end-to-end QoS guarantee, nor realize fine-grained flow control, which decrease the effect. To address these issues, we propose an incremental deployed approach, namely SoIP, to enhance the capability of QoS guarantee in Internet, and the core idea is to build a software-defined overlay network over IP network, and take advantage of the characteristic of per-flow management in SDN to satisfy the resource demands of these applications. Simultaneously, IP network preserves the differentiated services. Moreover, we present the coupling mechanism to implement the collaboration between software-defined networks and the IP network to achieve better resource utilization. Finally, we validate the feasibility and effectiveness of the proposed approach by simulation experiments.**

Keywords-software defined networking; IP Architecture; QoS guarantee; differentiated services

I. INTRODUCTION

In the past decades, Internet achieves rapid development, and becomes one of the indispensable infrastructures of human society. The design philosophy of Internet conforms to end-to-end arguments, which pushes the implementation of complex functions to the top of protocol stack and the network edge, and the network core merely makes its best effort to forward the arriving packets [1]. This argument ensures the features of flexibility, generality and openness, which effectively promote the boom of Internet. However, this design philosophy focus on the fair sharing of network resources, so those applications that have certain requirements on bandwidth, latency and packet loss are difficult to derive QoS guarantee. Currently, there are many multimedia and online services emerging in Internet, and these applications not only contribute a large number of traffic, but also have great impacts on politics, economy and human livings. On the other hand, they ask for higher demands on QoS for their real-time communication requirements. Therefore, it is imperative to reinforce the QoS guarantee capability of Internet to deal with this trend.

Internet Engineering Task Force (IETF) has formulated a series of standards related to the QoS of IP network until now. Nevertheless, some of these approaches either adopt complicated resource reservation and scheduling mechanisms [2], or lack fine-grained control on flow [3]. In fact, these proposals are trying to patch the current Internet architecture to provide QoS guarantee for the applications. However, the network infrastructure has ossified after long-term expansion, and it is extremely tough to implement any innovations and revolutionary evolution. This is the root of few improvements on QoS guarantee. At present, the voice of designing a clean slate architecture for future Internet is increasingly raising, while it faces the potential requirements of network applications on mobility, security, management in the future. But this opinion abandons the existing Internet infrastructure, which wastes enormous and valuable equipment, thus it lacks of enough motivation and is not realistic. In [4], Xie et al. also demonstrate that the validation and deployment of future Internet is necessarily evolvable, although the design of future Internet can start from zero. In contrast to other solutions, software-defined networking (SDN) is a practical and feasible approach. The conception of SDN is to separate control plane from data plane, and the network managers can achieve centralized control on these physical devices, which makes the network flexible and programmable. These characteristics of SDN also bring the opportunities to improve the ability of QoS guarantee of Internet by planning appropriate resource allocation and scheduling mechanisms. In order to solve the issues and challenges of top-level design of QoS guarantee in Internet, it is an alternative approach to introduce SDN and incremental method into current Internet infrastructure.

Inspired from Fabric [5], we present a novel approach, i.e. SDN over IP architecture (SoIP), to provide better QoS guarantee for end users and applications. The basic idea of SoIP is to update or reconstruct the network edge and build SDN-based overlay networks, while the network core maintains the existing differentiated services based on the Type of Service (ToS) field of IP protocol header. The characteristic of SDN is its intrinsic ability to implement the per-flow management and centralized control on network resources, and hence the network edge can accomplish fine-grained resource allocation and scheduling according to the QoS requirements of each flow. Simultaneously, the network edge can update on the fly, and other parts of Internet will not be impacted. Besides, the infrastructure and physical devices that has been established or used in Internet can be reserved, and SoIP is evolvable and can keep effective in different scale networks. Therefore, the advantages of SoIP are not only scalable, but also can achieve fine-grained management,

978-1-4673-9295-2/15 $31.00 © 2015 IEEE
DOI 10.1109/FCST.2015.17

which enhances the faculty of resource allocation on demand. More importantly, this approach is incremental and elastic, which implies that it is adaptive to the transformation of Internet.

The contributions of this paper can be summarized as follows:

- Based on the rising technologies, we propose a novel and incremental framework of network management and resource allocation to enable current Internet to provide better QoS guarantee for end users and applications.
- We present the resource scheduling mechanisms for QoS guarantee in SDN-based overlay networks and design the coupling and cooperation schemes to achieve the seamless combination of SDN and IP networks.
- We implement preliminary simulation experiments to evaluate the performance, which validates the availability of the proposed approach.

The remainder of this paper is organized as follows. Section II summarizes the related work. Section III outlines the framework of the proposed approach and its characteristics. Section IV presents the details of the coupling and cooperation mechanisms. Section V introduces the results of simulation experiments. Finally, section VI concludes our work.

II. RELATED WORK

Since the early 1990s, how to embed QoS guarantee mechanisms into Internet architecture and improve the resource utilization for supplying end users with high quality services are always the hot spots in network area. Consequently, IETF engaged in long-term standard establishment, and the representatives are two different QoS service models, which are Integrated Services (IntServ) and Differentiated Services (DiffServ) respectively.

The basic idea of IntServ is to achieve QoS guarantee via resource reservation so as to simultaneously support real time applications and other traditional services, and the focus of IntServ model is RSVP protocol. In practical process, RSVP sends signal packets to all physical links and network devices that are involved in the transmission path, and then sets up and maintains soft state of resource reservation hop by hop. Based on soft state, network implements packet classification, flow supervision and scheduling. After the transmission is finished, RSVP protocol resets the soft state of resource reservation. Therefore, IntServ must undertake complex resource reservation, access admission, QoS routing and scheduling, which leads to too much overhead and weaken the scalability.

In order to overcome the limitation of IntServ model, IETF tried to address the QoS issue from a novel perspective, and presented another proposal, i.e. DiffServ model. In DiffServ model, some types of services and their priorities are defined in advance, and the packet classification is completed in the ingress routers. During the packet classification, the protocol headers of IP packets are modified, and DiffServ code point (DSCP) is set. According to the DSCP, the core routers identify the type of packets, and provide the corresponding services for them. Moreover, DiffServ model also implements traffic aggregation and shaping. Although DiffServ model achieves higher scalability than IntServ, it is only able to realize coarse-grained resource management.

Neither IntServ model nor DiffServ model can offer general solution to satisfy all the QoS requirements of end users. In order to support end-to-end QoS, combining IntServ with DiffServ becomes an alternative feasible approach. In [6], the authors proposed to implement IntServ framework on DiffServ networks, and introduced the corresponding schemes to ensure user’s QoS. In [7], the authors adopted explicitly admission control in DiffServ model to achieve resource reservation. In [8], the authors designed a new network architecture, namely stateless core network (SCORE), and this architecture pushed the complicated state information to the nodes in network edge, while the nodes in network core did not need to participate in per-flow management and they were only responsible for ordinary packet forwarding. Besides, the authors introduced bandwidth broker (BB) into DiffServ model in [9], and BB concentrated on the management of network elements and resources to improve the efficiency of resource scheduling. Moreover, IETF proposed to integrate DiffServ with multi-protocol label switching (MPLS) [10], and provided end-to-end QoS guarantee for users by transforming the QoS of Layer 3 to that of Layer 2. In IPv6, the designers also tried to address QoS issue by inserting flow label field into the head of IPv6 datagram.

However, the above solutions don’t solve the problem of QoS guarantee completely, and the fundamental cause is due to the ossification of Internet. The rapid development of Internet architecture leads to the difficulty of innovation on network, and all works are patches on current architecture, which makes the efforts far away from the expected purpose. Therefore, it is necessary to introduce new architecture and redesign the QoS guarantee mechanisms from top-level. FIND project proposed a novel routing metric [11], which can adapt to dynamic congestion and motivate selfish users to share resources for QoS guarantee. SLA@SOI [12] proposed by European Union also focused on the implementation of dynamic services under the framework of service level agreement, but it still needs to solve the problem of the predictability and reliability of QoS. In contrast to these methods, McKeown et al. presented OpenFlow [13], which had initiative effect. OpenFlow is an instance of SDN, and it divides the traditional network into control plane and data plane. Under this architecture, network managers can accomplish network programming and assign the routing for per-flow to satisfy some certain requirements by controllers, which indicates that SDN is able to realize flexible network control and management. Due to these characteristics of SDN, Wang et al. attempted to adopt OpenFlow into load balancing of on-line services [14], while Gutz et al. implemented flow isolation for network operation via SDN. Therefore, SDN is helpful to promote the QoS guarantee of Internet.

III. THE FRAMEWORK OF SOIP

Current Internet is difficult to deal with QoS issue, while clean-slate architecture abandons the numerous resources in existing infrastructure and may results in new problems. Hence, it is necessary to adopt evolutionary method to improve the status of QoS guarantee in Internet. SoIP is an approach that can achieve incremental deployment in Internet, and it reforms the capability of QoS guarantee gradually without changing the core network and impacting the operation of Internet. SoIP incrementally introduces SDN into the network edge and reconstructs the access network to improve the integration and flexibility of network management and control. Simultaneously, it implements packet classification, flow label and state maintenance to assist the core network to ensure the QoS of important applications. Fig. 1 shows the framework of SoIP.

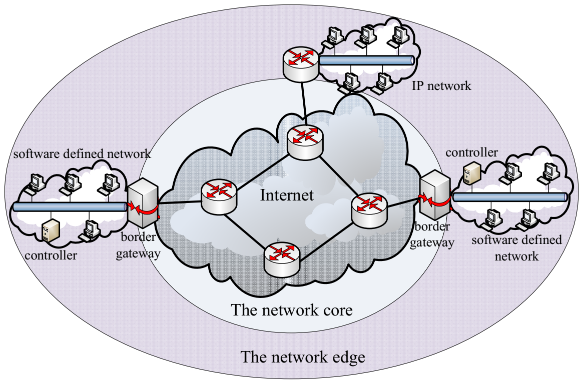

Figure 1. The framework of SoIP

From the perspectives of topology, the network edge employs SDN, for example OpenFlow, to organize and rebuild its elements, and then connects to the network core via border gateway in SoIP. The controllers in the network edge carry out centralized management and scheduling on network devices and resources, and they pay attention to ensure users' requirements on QoS. In the network core, DiffServ services and MPLS are still used to provide QoS guarantee.

The main features of SoIP are as follows:

- *High scalability and resilience*. Internet adopts end-to-end arguments, i.e. simple network core and complex network edge, to design the architecture of Internet, and realizes efficient operation. In SoIP, the core of Internet still keeps ordinary functions, including providing forwarding services for the packets based on the ToS field in IP layer or the labels in data link layer. At the same time, the complicated missions, including packet classification, traffic supervision and state maintenance are all undertaken by the network edge, which ensures the high scalability of SoIP. On the other hand, the border between the network core and the network edge in SoIP is alterable, and the area covered by the network edge can expand with the progress of SDN technologies. Under the circumstance, the level of network control and management is elevated further. Therefore, SoIP is resilient to the change of Internet.
- *Smooth transition from current Internet architecture to SoIP*. In SoIP, the network core maintains IP architecture, and k the existing Internet resources, and the reconstruction of the network edge can be accomplished incrementally. This characteristic allows Internet to evolve without impact on current network operation and achieve smooth transition to SoIP.
- *Independency between access networks*. The core of the network in SoIP is not changed, and those access networks in the network edge still communicate with each other over the network core. Therefore, the network edge only needs to guarantee the connection with the network core. Consequently, the access networks are independent, and isolate the impact of local network failure due to the misconfiguration of SDN controllers.
- *Multiple usable mechanisms for QoS guarantee*. Single mechanism cannot address the QoS issue, and it is necessary to simultaneously employ multiple schemes to improve the capability of QoS guarantee of Internet. SDN fulfills the purpose of centralized and flexible management on the network devices and resources, including adjusting the delivery path, modifying the field of packet header, limiting the transmission rate of traffic. As a result, packet classification, flow isolation and admission control can be accomplished in the SDN-based network edge. Moreover, the priority of per-flow is able to be determined according to the type of applications and the level of users. And then a whole QoS guarantee system is established.

Although the framework of SoIP which builds SDN-based overlay on IP networks contains bright characteristics and technologies, there still have some practical issues needing to be addressed.

- *The mechanism design of QoS guarantee in the network edge*: This issue involves how to reconstruct the network edge and actualize the QoS guarantee mechanisms to meet the QoS requirements of certain flows transmitting in the network edge.
- *The coupling and cooperation schemes between the network edge and the network core*: Due to the differences of the QoS guarantee mechanisms between the network edge and the network core, this issue of collaborating these two parts of network to implement seamless combination in the conjunct should be solved.

IV. QoS Guarantee Mechanisms in SoIP

A. *Supporting QoS in the Network Edge*

In the SDN-based networks, the central controllers can be aware of the network topology and manipulate the routing of per-flow. Therefore, these advantages of centralized control and flexible scheduling should be sufficiently utilized for the implementation of QoS guarantee.

1) Resource Request and Allocation: The essence of QoS guarantee is to provide the corresponding services for certain important applications to satisfy their demands on some performance metrics, latency and bandwidth for example. Accordingly, the flow should be marked, and then the network guarantees its QoS requirements. Specifically, the phases for QoS guarantee in network edge are summarized as follows:

- *Policy preset*. Before the network starts to deliver packets, the policies of packet forwarding should be preset in the controllers by the network managers. These policies involve the level of users, the type of services for each application and so on. According to these policies, the controllers decide whether the QoS requirements of the arriving flows should be guaranteed and can be conscious of their corresponding demands on performance metrics. Furthermore, the QoS requirements can be customized by end users.
- *QoS registration*. When the first packet of a flow enters the SDN-based network, it will be forwarded to the controller to determine the routing from source to destination, and then the following packets travel along this established path. In order to assist the controller to provide QoS guarantee, some customized requirements of end users can be piggybacked in the first packet to the controller for QoS registration. And then the controller can derive the information of QoS demands that this flow expects.
- *Routing computation*. After receiving the first packet, the controller will obtain the flow's QoS requirements according to the customized field or the preset policy. Then the controller finds out a path that meets the QoS requirements based on current network topology and the resource utilization of each link. If there are multiple candidates, the shortest path is selected.
- *Flow establishment*. When the routing is determined, the controller sends the flow entry to the involved SDN switches, and the flow entry also contains the resource requirements that should be reserved for this flow. Subsequently, these switches provide corresponding bandwidth resources for this flow until it finishes its transmission.
- *Path reset*. When the transmission of the flow is ended, none of these packets belonging to this flow will pass through the switches in the path. But maintaining these flow entries will waste computing and storage resources, so the switches can remove these entries if the flow is inactive after a while such as one minute.

Fig. 2 shows the process of the QoS guarantee mechanism.

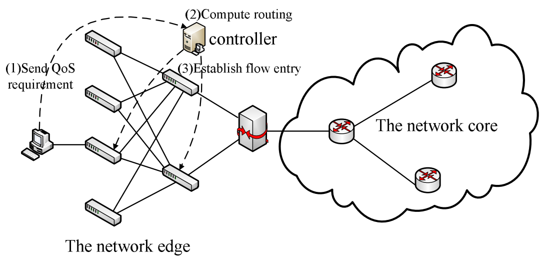

Figure 2 The process of QoS guarantee mechanism

During routing computation, the controller should find out all potential paths from source to the border gateway based on the information of network resource reservation. If any link in the path cannot supply enough resources for the new arriving flow, this candidate is unqualified. And the specific process of routing computation can be summarized as follows: Firstly, the controller excludes these switches/nodes whose available resources are deficient from the global network topology; Secondly, the controller employs Dijkstra algorithm to discover the shortest path by supposing the weights of all links being equal. Fig. 3 shows the formalized description of routing computation algorithm.

Algorithm: Routing computation algorithm
Input: Resource requirement R, Available resource r_i of node i
Output: The shortest path P

```
1.  foreach node i in the SDN-based network
2.    if r_i<R
3.      remove i from the node set;
4.      remove all links associating with i;
5.    end if
6.  end for
7.  insert source node into N;
8.  initialize the path for all nodes;
9.  while |N| is less than the number of node set
10.   foreach node i connecting to any node in N;
11.     foreach j in N
12.       if i is linked to j
13.         path[i]=path[j]+j;
14.         break;
15.       end if
16.     end for
17.     if i is the destination node
18.       return path[i];
19.     end if
20.     insert i into N;
21.   end for
22. end while
23. return NULL;
```

Figure 3 Routing computation algorithm

2) Load Balancing and Admission Control: Due to the diversity and burst of network traffic, the load distribution may be skew, and the total resource requirements in some links may exceed the resource supply, and the leaving resources are unable to guarantee the QoS demand of the arriving flow. This case will result in the failure of routing computation and the absence of the path that fulfills the QoS requirement. According to the routing computation algorithm, it will return null in line 23. Under the circumstance, some necessary adjustment should be executed to solve this problem. In practical, there are two feasible approached as follows.

- *Routing adjustment*. Although some physical links undertake over loading, this situation may be caused by the unbalance of traffic distribution, and many other links are still idle. In this case, the routing computation can be finished via flow adjustment. Specifically, the controller queries the instant information of resource allocation, and implements dynamic programming for each flow based on the global traffic state. Once there are adequate resources to satisfy the QoS requirement of the arriving flow in other paths, it switches the routing of some flows to the relatively spare physical links, and then these reserved resources are released. Finally, the controller can continue the phase of routing computation.

- *Admission control*. In some extreme situation, there does not have any possible path in the network to guarantee the QoS requirements of the new arriving flow, and load balancing mechanism cannot be applied any longer. At this time, it is appropriate to introduce admission control. There are two feasible ways to carry out admission control, and the first one is to restrict the entry of new arriving flows until there are sufficient resources for their requirements, while the other one is to suspend the transmission of some uncritical flows according to the type of applications and the level of users. The rules of second way should be set down in advance. The practical implementation is to find the flow table in the corresponding network devices and delete these entries via the controller, and then these flows are unable to continue transmitting. After the network loads are alleviated, the controller inserts the entries into flow table again.

B. Coupling and Cooperation Mechanisms between the Network Edge and the Network Core

The access networks in the network edge employ novel SDN technology to ensure the QoS requirements of end users, while IP networks in the network core still adopt DiffServ model and MPLS for QoS guarantee. Due to the difference of implementation mechanisms, it is necessary to design related coupling scheme to achieve the collaboration of the whole QoS guarantee architecture. Specifically, network reachability and QoS mapping should be addressed.

1) Network Reachability: The core IP networks in the network core are connected by various inter/intra-domain routing protocols, including OSPF, EIGRP, IS-IS, BGP etc. In SoIP, we do not change the communication mode, thus we need to accomplish silimar routing functions in the SDN-based network edge.

Because most of routing protocols exchange routing information by multicast addresses, the SDN-based network edge should establish a path for the information exchange from the border gateway to the center controller, and the network devices involved in this path must create the corresponding flow entries that contain the multicast addresses of routing protocols. Thus, all packets generated by routing protocols in the network core will be forwarded to the controller. When the controller receives the control packets of routing protocols, it firstly recognizes the types of routing protocol via the multicast address and the protocol header of control packets. Subsequently, it refers to the same protocol format for encapsulation of routing information, and then sends these packets back to the border gateway along the established path. Finally, the gateway exchanges the routing information with those directly connected routers in the network core. Therefore, the communication between the network core and the network edge is successfully completed.

2) Qos Mapping: The SDN-based network edge achieves QoS guarantee via per-flow management, and the controller can control and coordinate multiple network devices to reserve required resources, while IP networks provide differentiated services according to the ToS field in the header of IP packets or append MPLS in the datalink layer. Both the principle and implementation are different in these two parts of network, and hence related scheme for solving the problem produced by the difference of QoS guarantee methods should be designed to actualize the transformation of QoS requirements.

In order to achieve QoS mapping, the characteristic of network programmability in SDN can be utilized. In practical, the controller notifies the border gateway the information of QoS requirements when a new flow is forwarded to it. According to the fixed policy, the border gateway can rewrite the ToS field in the header of IP layer if a packet arrives, and the type of service should be suited to the flow's demand, thereby the ToS field of all packets arriving at the border gateway will be modified with appropriate content before they leave the network edge.

After QoS mapping in the border gateway, the packets will enter the network core, and then they can obtain the corresponding services according the ToS field. The router will provide suitable services for these flows, or encapsulate these packets with MPLS label, and then execute another QoS mapping from the third layer to the second layer.

Before the packets arrive in the destination network edge, they should experience reverse QoS mapping. Due to the two communication entities are located in distinct network edge, and even belong to distinct ISP, they are managed by different controller, and the policies of QoS guarantee for end users may be diverse. The reasonable approach is to implement reverse QoS mapping according to the policy in the destination SDN-based network edge. Although this situation impacts the QoS guarantee, it will be gradually mitigated by combining content distribution network.

V. Performance Evaluation

A. Simulation Scenario

To evaluate the feasibility and performance of SoIP, we implement a simulation experiment, and this simulation is an event-driven program. In the experiment, the sending and receiving of all packets can be abstracted as events, and these events are been labeled with timestamp when they are generated, and then the program schedules these events according to the sequence of their timestamp. During the process of simulation, all types of simulation results are recorded. We design a simulation scenario with dumbbell structure, and Fig. 4 shows the practical network topology.

In Fig. 4, three switches, i.e. A, B and C, are managed by controller, and all network devices are connected with 100Mbps links. In the beginning of simulation, host Y sends TCP flow to host V, while host Z sends TCP flow to host W.

After 10 seconds, host X starts to send another TCP flow to host D, and this flow lasts 30 seconds. In the first experiment, the flow sent by host X asks for 50Mbps bandwidth guarantee, while the others does not have QoS requirement. In the second experiment, both host X and host Z require 40Mbps bandwidth provision, while host Y has no demand.

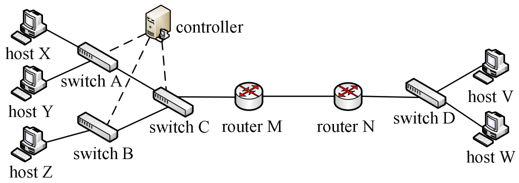

Figure 4 The network topology for simulation

B. *Experimental Results and Analysis*

As shown in the network topology, the bottleneck is in switch C, and we are able to derive the experimental results by gathering the statistical information recorded in router M. In practical, we collect the state of packet transmission every 100 milliseconds, and then transform them to bandwidth utilization. The simulation lasts 1 minute, and Fig. 5 shows the sending rate of the three hosts in router M.

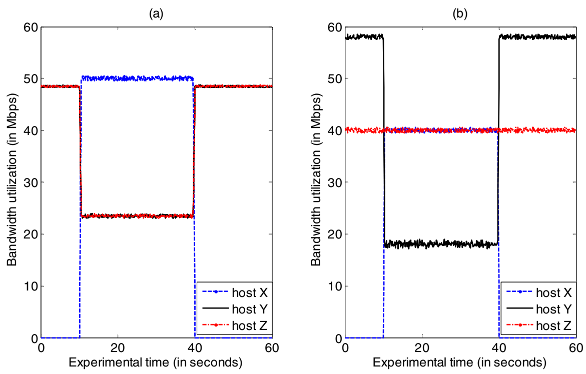

Figure 5 The curve of sending velocity of three hosts

Fig. 5(a) shows that the bandwidth occupied by host Y and host Z are both 50Mbps due to the fairness of TCP in the beginning, but the bandwidth resources of switch A and switch C are going to be reserved for the flow generated by host X when it starts to send packets in the 10th second, and the flow derives about 50Mbps bandwidth during its transmission process while the bandwidth utilization of the other hosts decreases from 50Mbps to 25Mbps. In Fig. 5(b), the bandwidth requirements of host X and host Z are also satisfied. This result indicates SoIP has the potential to ensure the QoS requirements of end users and applications on performance metrics.

VI. CONCLUSIONS

TCP/IP suite based Internet has experienced long-term rapid development in the past decades, but this process accompanies the ossification of Internet. On the other hand, the rising applications yet bring higher demands on the capability of QoS guarantee. This paper aims at the promotion of QoS guarantee in Internet, and proposes an evolutionary approach based on SDN, namely SoIP. This approach sufficiently considers the Internet architecture, and gradually improves the QoS guarantee support of Internet for important applications and users without discarding the existing infrastructure, so our approach is more practical and feasible. Moreover, SoIP is scalable and resilient, and it can achieve incremental deployment in Internet. The results of simulation also preliminarily validate the availability of SoIP, and our future work will focus on the practical implementation and fine-grained resource scheduling mechanism.

Acknowledgment

This work is supported by the State Key Development Program for Basic Research of China under Grant No. 2012CB315806, the National Natural Science Foundation of China under Grant No. 61379149, Jiangsu Province Natural Science Foundation of China under Grant No. BK20140070.

References

[1] M. Blumenthal, D. Clark. Rethinking the design of the Internet: The end to end arguments vs. the brave new world. ACM Transactions on Internet Technology, 2001, 1(1): 70-109.

[2] R. Braden, D. Clark, S. Shenker. Integrated services in the Internet architecture: An overview. RFC 1633, 1994.

[3] S. Blake, D. Black, M. Carlson, et al. An architecture for differentiated service. RFC 2475, 1998.

[4] G. Xie, Y. Zhang, Z. Li, et al. A survey on future Internet architecture. Chinese Journal of Computers, 2012, 35(6): 1109-1119.

[5] M. Casado, T. Koponen, S. Shenker, et al. Fabric: A retrospective on evolving SDN. Proceedings of the SIGCOMM 2012 Workshop on Hot Topics in Software Defined Networking, Helsinki, 2012: 85-90.

[6] Y. Bernet, R. Yavatkar, P. Ford, et al. A framework for integrated services operation over diffServ networks. RFC 2988, 2000.

[7] Y. Bernet. Format of the RSVP DCLASS object. RFC 2996, 2000.

[8] I. Stoica, H. Zhang. Providing guaranteed services without per flow management. Proceedings of the ACM Sigcomm, Cambridge, Massachusetts, USA, 1999: 81-94.

[9] Z. Zhang, Z. Duan, L. Gao. Decoupling QoS control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed services. Proceedings of the ACM Sigcomm, Stockholm, Sweden, 2000: 71-83.

[10] F. L. Faucheur, L. Wu, B. Davie, et al. Multi-protocol label switching support of differentiated services. RFC 3270, 2002.

[11] B. Awerbuch, B. Haberman. Algorithmic foundations for Internet architecture: Clean slate approach. NSF NeTS FIND Initiative, http://www.nets-find.net/Funded/Algorithmic.php.

[12] Empowering the Service Economy with SLA-aware Infrastructures. European Union 7th Framework Program, http://sla-at-soi.eu.

[13] N. McKeown, T. Anderson, H. Balakrishnan, et al. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38, (2): 69-74.

[14] R. Wang, D. Butnariu, J. Rexford. OpenFlow-based server load balancing gone wild. In Proc. of the 1st USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services. Boston: USENIX Association, 2011.

[15] S. Gutz, A. Story, C. Schlesinger, et al. Splendid isolation: A slice abstraction for software-defined networks. HotSDN 2012.