

Application-aware routing with QoS support in SDN networks^①

Zhang Zexin (张泽鑫)^{②*}, Li Jun*, Wu Haibo*

(* Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, P. R. China)

(** University of Chinese Academy of Sciences, Beijing 100049, P. R. China)

Abstract

Software defined networking (SDN) offers programmable interface to effectively control their networks by decoupling control and data plane. The network operators utilize a centralized controller to deploy advanced network management strategies. An architecture for application-aware routing which can support dynamic quality of service (QoS) in SDN networks is proposed. The application-aware routing as a multi-constrained optimal path (MCOP) problem is proposed, where applications are treated as QoS flow and best-effort flows. With the SDN controller applications, it is able to dynamically lead routing decisions based on application characteristics and requirements, leading to a better overall user experience and higher utilization of network resources. The simulation results show that the improvement of application-aware routing framework on discovering appropriate routes, which can provide QoS guarantees for a specific application in SDN networks.

Key words: application aware, software defined networking (SDN) networks, quality of service (QoS), multiple constraints, path selection

0 Introduction

Traditional networks are designed to forward packets from source to destination using the shortest route possible and do not take into account resource requirements of applications and overall network state^[1]. This is because of the lack of information exchange between applications and Internet, which may result in poor routing because many flows can be forwarded through the same link and generate heavy network congestion. In order to achieve the best route, a router ideally needs global knowledge of the current network status^[2], and dynamically adjusts the forwarding path of applications to satisfy their quality of service (QoS) on the available resources.

The demands of QoS for network applications are diverse. For example, on-line gaming, VoIP or video demand more bandwidth and low-latency. However, the requirements of realtime applications cannot be always met by the standard best-effort Internet. Over the past years, a lot of literatures have proposed several architectures for QoS in IP networks, such as integrated services (IntServ), differentiated services (DiffServ), but none has been truly successful and globally implemented^[3]. This is because of the lack of visibility of

overall network state. Over large networks, it is difficult to collect the global view of the network due to the scale of the network. The problem becomes more challenging in the Internet because of its completely distributed architecture.

Software defined networking (SDN) provides a solution for an enhanced communication between networks and applications. SDN is a developing architecture where network control is decoupled from forwarding and is directly programmable. Network intelligence is logically centralized in software-based SDN controllers, which maintain a global view of the network^[4]. The applications can exploit centralized network state information to seamlessly adapt network behavior to user needs^[1]. The application-aware controllers can control the forwarding devices to adjust flows by leveraging the global view. With the new paradigm of SDN, the optimal path can be selected for different applications with diverse demands.

In this work, the advantage of the SDN technique is used to realize application-aware routing depending on a global perspective. The controller can make differentiated routing strategy according to the overall network state and QoS of applications. The contributions of this study are as follows:

① Supported by the National Basic Research Program of China (No. 2012CB315803), the Around Five Top Priorities of One-Three-Five Strategic Planning, CNIC (No. CNIC PY 1401), Chinese Academy of Sciences, and the Knowledge Innovation Program of the Chinese Academy of Sciences (No. CNIC_QN_1508).

② To whom correspondence should be addressed. E-mail: zhangzexin@cstnet.cn
Received on Sep. 17, 2015

(1) the application-aware routing in SDN is investigated and an optimization model to determine how to calculate forwarding paths for the traffic with QoS support is constructed.

(2) An application-aware routing mechanism which runs on the top of SDN centralized controller and dynamically calculates routes for network applications is proposed to fulfill the demands of different services.

(3) The new approach is evaluated and it is found that it could show competitive performance compared with traditional strategy.

The paper presents the related work in Section 1. Section 2 formulates the application-aware routing problem which is solved as a Multi-Constrained Optimal Path (MCOP) Problem. Section 3 describes the developed architecture, while Section 4 presents simulation experiments and results. Section 5 concludes the work.

1 Related work

1.1 Software defined networking (SDN)

SDN is a new networking paradigm in which the flexibility and innovation of network are increased. It separates the data plane and the control plane, and the behavior of the entire network is controlled by a logically centralized software program. The control plane is responsible for making forwarding decisions of flows. The data plane becomes simple packet forwarding devices which can be programmed via a southbound interface.

With SDN, the network devices no longer need to understand and process thousands of protocol standards but merely accept instructions from the SDN controllers. OpenFlow^[5] has been one of the most effective southbound interfaces. It standardizes information exchange between the control plane and the data plane. OpenFlow allows the forwarding path of traffic to be defined based on parameters such as usage patterns, applications, and network resource, enabling the routing to respond to real-time changes. However, current IP-based routing does not provide this level of control.

A few of works have been conducted based on the SDN technique. Plug-n-Serve^[6] was an Open-Flow based server load-balancing system which can keep track of state and control the routes. The system effectively reduces response time of web service in unstructured networks. Ref. [7] developed a load balancer system which can handle multiple services with the multiple OpenFlow controllers. Ref. [8] implemented a load-balancer architecture based on the OpenFlow to reduce the cost and increase the flexibility. B4^[9] adopted SDN architecture for the data center WAN in-

terconnect and developed centralized traffic engineering mechanisms. They claim that many links have achieved near 100% utilization.

1.2 Application-aware routing

Qazi, et al.^[10] proposed that application-awareness became even more important with SDN and presented a framework, Atlas, which has realized fine-grained, accurate and scalable application classification in SDN based on a machine learning technique.

Jarschel, et al.^[11] combined application-state information with SDN network control for network management. They had emulated a path selection scenario for an access network provider. The application station information were collected for informing the controller that an action was required for a particular flow in order to maintain the Quality of Experience (QoE) for the user.

Adami, et al.^[12] proposed a new control application that enabled differentiated routing for traffic flows belonging to different service classes. Deep packet inspection module was used to detect the type of traffic. Then, the new control application made routing decisions based on the real-time network status information and enforced a differentiated traffic routing for different applications.

Egilmez, et al.^[13] presented a new framework for optimization of forwarding decisions to enable dynamic QoS over OpenFlow networks. They applied their methodology to scalable video streaming whose QoS was improved significantly.

The proposed approach in this work differs from the prior work in two aspects: a) The application-aware routing problem formulated and an optimization model is constructed. b) QoS guarantee of all application is focused on. Unlike other solutions, on a single streaming is concentrated.

2 Problem model

In order to generalize the basic idea of the proposed solution, the application-aware routing model which finds a routing solution that it provides QoS guarantees to different applications dynamically is developed.

2.1 Problem formulation

Each application has different QoS demands. Among the various QoS parameters, the two major ones are bandwidth and end-to-end delay^[14]. Ref. [3] proposes that the delay variation is a significant QoS parameter in video streaming applications. So, band-

width, end-to-end delay and delay variation (jitter) will be employed as the constraint in this work. The parameters can be classified into additive (e. g. , delay, jitter) and non-additive (e. g. , bandwidth). The application-aware routing problem is then to find routes that satisfy some QoS requirements of every traffic flow, at a minimal cost. For the additive parameters, the cost of an end-to-end path is given by the sum of the individual link values along the path. However, for the non-additive parameters, the cost of a path is determined by the value of the bottleneck link^[15].

The application-aware routing is posed as a multi-constrained optimal path (MCOP) problem. The network is modeled as a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of links. Each link $(i, j) \in E$ is associated with a cost value $c(i, j)$, bandwidth $b(i, j)$, delay $d(i, j)$ and jitter $j(i, j)$. The forwarding paths R of application A_i are characterized by the following values:

- source node S_i and destination node T_i ;
- bandwidth requirement B_i ;
- maximal end-to-end delay constraint D_i ;
- maximal delay variation J_i

Given the three constraints, the MCOP problem is to find a path p from the source node S_i to the destination node T_i such that:

$$(1) \min \{ d(i, j) \mid (i, j) \in p \} \geq B_i$$

$$(2) w_d(p) = \sum_{(i, j) \in p} d(i, j) \leq D_i$$

$$(3) w_j(p) = \sum_{(i, j) \in p} j(i, j) \leq J_i$$

$$(4) c(p) = \sum_{(i, j) \in p} c(i, j) \text{ is minimized over all}$$

feasible routes satisfying (1) (2) (3).

The non-additive QoS parameter such as bandwidth can be preprocessed by removing all links that do not meet the demands. The additive QoS parameters such as delay, delay variation are combined into one cost metric by using the nonlinear cost function. The formula is

$$c_\lambda(p) = \left(\frac{w_d(p)}{D_i} \right)^\lambda + \left(\frac{w_j(p)}{J_i} \right)^\lambda \quad (1)$$

where $\lambda \geq 1$.

Ref. [15] has proved that the probability of finding the feasible paths will be increased if λ is increased in Eq. (1). When $\lambda \rightarrow \infty$, the chance of finding the feasible paths is the largest. The Eq. (1) can be replaced by

$$c_\infty(p) = \lim_{\lambda \rightarrow \infty} c_\lambda(p) \quad (2)$$

The largest item in Eq. (1) will dominate the $c_\lambda(p)$ as $\lambda \rightarrow \infty$. So Eq. (2) can be replaced by another cost function as follow.

$$c_\infty(p) = \max \left\{ \frac{w_d(p)}{D_i}, \frac{w_j(p)}{J_i} \right\} \quad (3)$$

2.2 Solution to the multi-constrained optimal path problem

The solution to the MCOP problem will try to find a feasible path that satisfies QoS requests from source to destination with the minimum cost. The MCOP problem is known to be NP-complete. Some ways to this problem have been proposed, such as heuristic and approximation algorithms. H_MCOP^[15] is a classical algorithm to solve the MCOP problem and the H_MCOP algorithm is used.

As shown in Table 1, the H_MCOP algorithm maintains the labels for each node u . Then, it uses the information to search the feasible paths. A pseudo code for H_MCOP is shown in Fig. 1. The algorithm executes two modified versions of Dijkstra's in backward

Table 1 Node information

Notation	Meaning
s	Source node
t	Destination node
$r[u]$	The shortest path from u to t (c_λ with $\lambda = 1$).
$R_k[u]$	The individually accumulated link weights from u to t .
$\pi_r[u]$	The predecessor of u on this optimal path.
$g[u]$	The cost of a foreseen complete path that goes from s to t via node u .
$G_k[u]$	The individually accumulated cost of link weights from s to u .
$\pi_g[u]$	The primary cost from s to t .
$c[u]$	The predecessor of u on this path.

H_MCOP($G = (V, E), s, t, c_k, k=1, 2, \dots, K$)

1 Reverse_Dijkstra($G = (V, E), t$);

2 if $r[s] > K$ then

3 return failure /* there is no feasible path */

4 end if

5 Look_Ahead_Dijkstra($G = (V, E), s$);

6 if $G_k[t] \leq c_k \forall k=1, 2, \dots, K$ then

7 return the path /* a feasible path is found */

8 end if

Fig. 1 The heuristic algorithm for the MCOP problem^[12]

and forward directions. In the backward direction, the algorithm uses Reverse Dijkstra^[16] to find the optimal paths from every node u to t . For that, it uses function $c_\lambda(p)(\lambda = 1)$ as cost metric. If there are feasible paths, the algorithm calls Look _ Ahead _ Dijkstra in the forward direction. This procedure finds the optimal path w. r. t the cost function $c_\lambda(p)(\lambda > 1)$.

The H _ MCOP algorithm only needs to run Dijkstra's algorithms twice to solve the MCOP problem with the complexity of $O(n \log(n) + m)$, where n and m are the number of nodes and links, respectively. The algorithm can reduce the AvgCost when the network contains a large number of feasible paths, and increase the utilization of network resources. Therefore, the H _ MCOP algorithm is well suited for use in dynamic application-aware routing.

3 An architecture for application-aware routing with QoS support in SDN networks

3.1 Application-aware in SDN

The packets are forwarded from source to destination using the shortest route in traditional networks. The forwarders are usually agnostic to the applications. In many cases, static network configuration may cause the network resources waste and poor user experience. However, SDN can make an intelligent network and open the door for on-demand resource allocation^[4]. The APIs of SDN, such as OpenFlow, are capable of layer L2/3/4-based policy enforcement but they currently lack higher layer application awareness^[10]. OpenFlow allows the flexibility of defining flows using layer L4 header fields. So, in this study, the port-based approach is used to achieve layer L7 awareness which can differentiate varieties of applications. When the traffic arrives, its port is initially detected by the controller. Then, the traffic flow is classified to a pre-defined class and forwarded according to the QoS demands of this application. Other methods should be developed to realize layer L7 awareness, but are out of the scope of this work.

3.2 Dynamic routing with QoS-enabled in SDN

In order to meet QoS requirements of applications, a dynamic routing mechanism, which can instantly modify the routing rules based on the visibility of overall network state, is presented. The routing rules are determined by the controller of OpenFlow. Then, the controller installs the rules in the forwarders of OpenFlow. The network forwarders transmit the traffic flows according to the routing rules. However, it is

impossible to change network routing on a per-flow basis in the current Internet architecture^[3]. The controller is the brain of SDN network and includes different components. The solution proposes a controller which depicted in Fig. 2 and consists of various interfaces and modules. The detail descriptions of these components are as follows.

Controller-Application interface: The north-bound API presents a network abstraction interface to the applications. They are used to communicate between the SDN Controller and applications running over the network.

Controller-Infrastructure interface: The controller communicates with network infrastructure devices by a secure OpenFlow protocol, which provides an open protocol to program the flow-table in different forwarders.

Topology management module: This module can help the controller have knowledge of network connectivity topology. It is realized by the Link Layer Discovery Protocol (LLDP)^[17]. To mitigate the overhead of transmitting LLDP packets, this module is triggered only when the links are initialized or broken off.

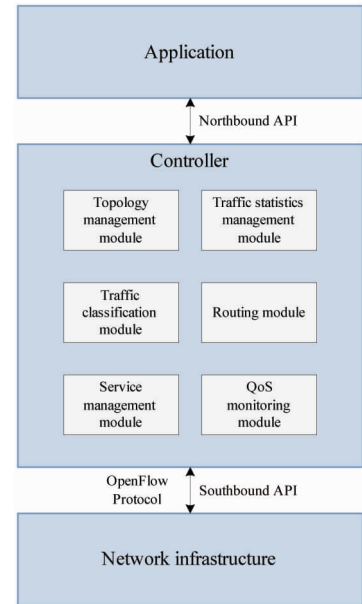


Fig. 2 The architecture of the Controller

Traffic statistics management module: This module gathers the up-to-date network state from OpenFlow forwarders, such as table, flow, port and queue, which can be used to get the precise status of overall network.

Routing module: The topology information and traffic statistics are input to this module, which calculates routes for different types of flows. The appearance

of network congestion, this module recalculates a new route for applications.

Traffic classification module: This module is responsible for traffic classification. When a new traffic flow arrives, this module classifies it to a pre-defined class.

Service management module: This module manages and stores the requests of applications, which are described by an XML format^[18] as shown in Fig. 3.

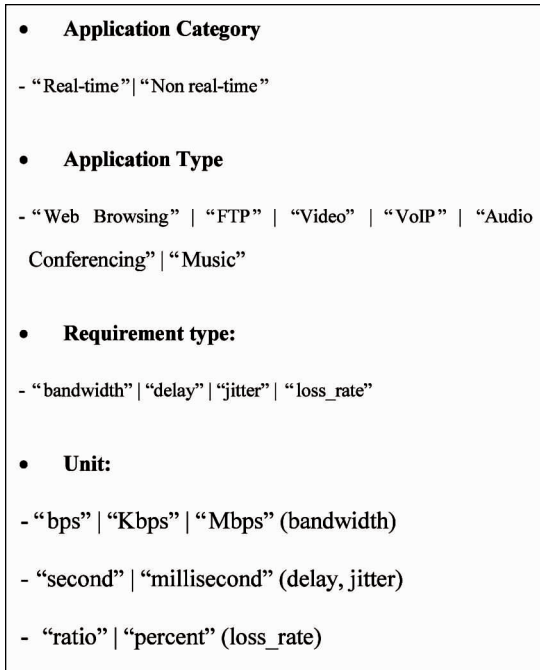


Fig. 3 An application profile grammar^[18]

QoS monitoring module: This module monitors the QoS change of network traffic. If the requested QoS level is not available, this module notifies the routing module which recalculates a new route for the traffic flow.

When the traffic arrives, it is initially received by the traffic classification module of the controller and this module recognizes the type of application. Then, the service management module determines the requests of this application. If the QoS demands of application are satisfied by the network resources, the routing module computes an optimal route and the controller uploads the flow table to the appropriate forwarders. Meanwhile, the QoS monitoring module monitors the QoS change of applications. If the requested QoS level is not available, the routing module is reactivated to determine a new route for the data flow.

There are different QoS demands for traffic of certain types. Thus, the routing module can make routing decisions for a differentiated treatment of various appli-

cations. The traffic flows are defined as different types: QoS service flows and best-effort service flows. The routing module calculates routes for each traffic flow based on its flow type. The route of QoS service flows can be dynamically recalculated based on the network state.

4 Experimental validation

To evaluate the performance of the proposed application-aware routing architecture, a comprehensive SDN test network based on POX^[19] and Mininet^[20] is created and extensive simulation using the network shown in Fig. 4 is conducted. The topology is acquired by emulating an enterprise production network where QoS service streaming is generated.

The emulated network (see Fig. 4) consists of 6 Hosts and 10 OF-enabled switches. The link capacities of network are set to be 5Mbps, 10Mbps and 15Mbps and randomly assigned to each link. The bursty behavior of network traffic in sub-second small time scales is modeled to a Poisson random process^[21]. Thus the Poisson model is used to generate the cross traffic for each link. The link delays are set to 10ms, 15ms and 20ms randomly.

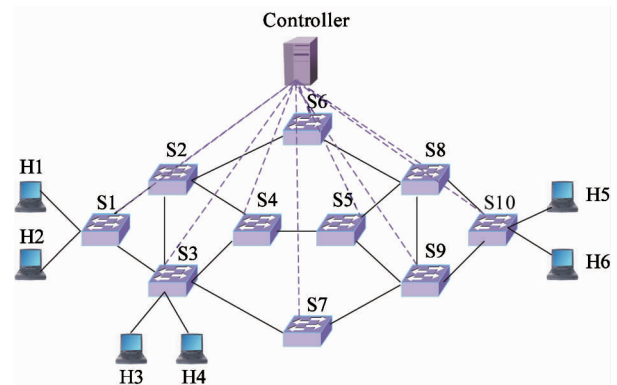


Fig. 4 The network scenario

A QoS service which satisfies the constraints $B_i = 2M$, $D_i = 10ms$, $J_i = 100ms$ is emulated. The controller calculates routes for the QoS service by using the H _ MCOP algorithm. When a new service is arriving, the proposed controller executes following steps:

- The data flow is classified to a pre-defined class.
- The QoS requests (B_i , D_i , J_i) of this application are acquired from the service management module.
- If this application is a QoS service, the routing module calculates route for it by solving the MCOP problem using the H _ MCOP algorithm.
- The QoS change of this application is moni-

tored by the QoS monitoring module for each time interval. The controller reroutes for this service when the demands are not met by the previous route.

In order to show the feasibility of the solution, two following scenarios are experimented:

Scenario 1: Routing without application-aware. In this scenario, all packets are sent using the shortest path.

Scenario 2: Routing with application-aware. In this scenario, firstly, the applications are classified by the solution. Then, the QoS services are transmitted by using the H_MCOP algorithm and the best-effort services are sent by using the shortest path.

These scenarios are simulated with the low, medium and high load network respectively. Then, the end-to-end delay, packet loss, jitter are measured.

4.1 Low load

The comparable performance for two scenarios in low load is depicted in Fig. 5. Fig. 5(a), (b), (c) show the loss rate, jitter and end-to-end delay observed during the emulation experiments, respectively. In Fig. 5(a), the average packet loss rates are 0.57% and 0.28% for two scenarios, respectively. The average jitters for two scenarios are 0.07ms and 0.04ms, which are observed in Fig. 5(b). From Fig. 5(c), it is found that the average end-to-end delays for two scenarios are 6.0ms and 5.9ms, respectively. The performance of the solution is similar to the without application-aware mode in low load, which implies that the shortest path is selected by our proposed approach to forward network traffic.

4.2 Medium load

In Fig. 6, the performance for two scenarios with medium load network is described. Fig. 6(a) describes the average packet loss rates which are 3.54% and 0.31% for two scenarios. From Fig. 6(b), it can be observed that the average jitters are 1.52ms and 0.17ms, respectively. The average end-to-end delays are reported in Fig. 6(c). They are 6.1ms and 5.9ms. It can be seen from Fig. 6 that improved loss rate and jitter are achieved by the applying application-aware routing method. At $t = 30$ s, the solution finds that the loss rate is beyond the threshold. And the routing is recalculated to maintain the quality of the application. It highlights that the control application is able to update routings according to the application status.

4.3 High load

The performance for two scenarios with high load

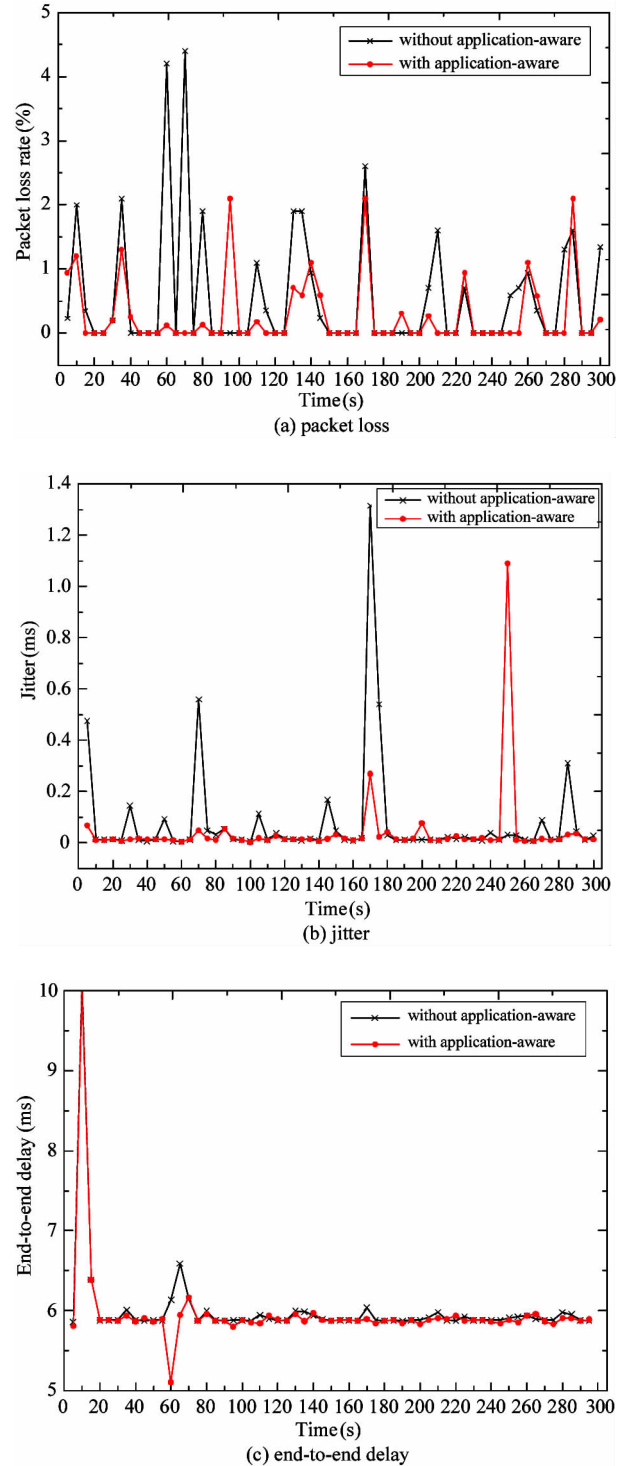


Fig. 5 Performance comparison of two scenarios in low load

is examined, as shown in Fig. 7. The shortest path method has significant amount of quality loss. The average loss rate varies from 13% to 82%. On the other hand, the average jitter and delay are 119.6ms and 10.2ms. However, the solution is not affected by the cross traffic. The proposed approach first checks the QoS requests based on pre-defined service definitions.

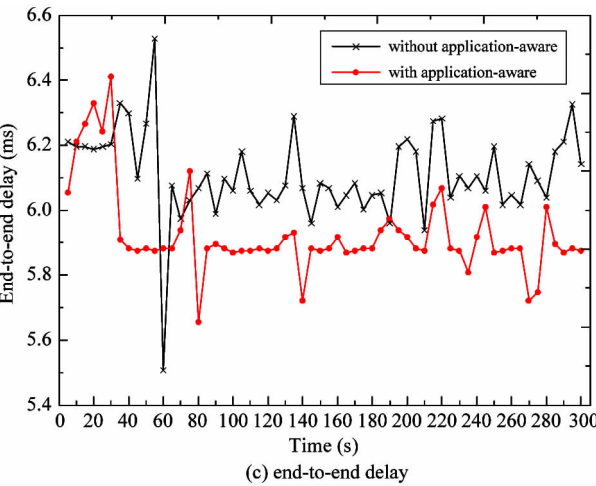
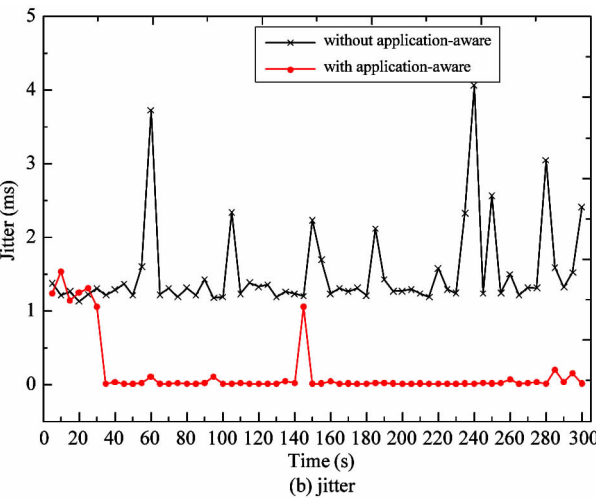
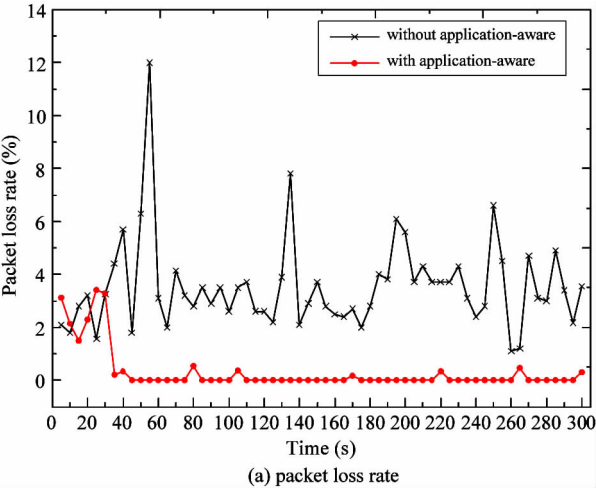


Fig. 6 Performance comparison of two scenarios in medium load

Then, the route module calculates a QoS optimized path between the source and destination. It can also dynamically update the forwarding paths of old flows according to the network and application state information.

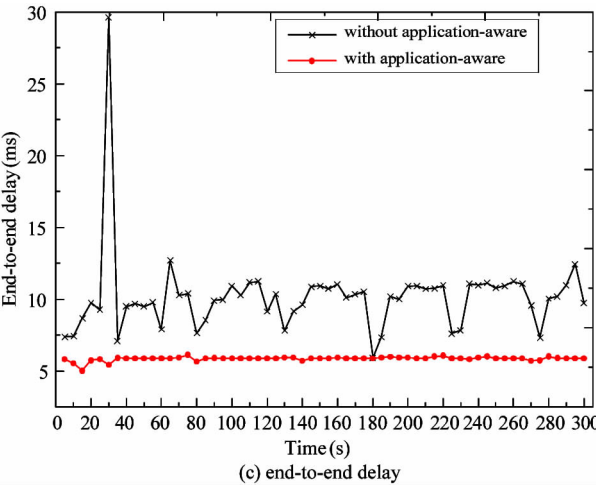
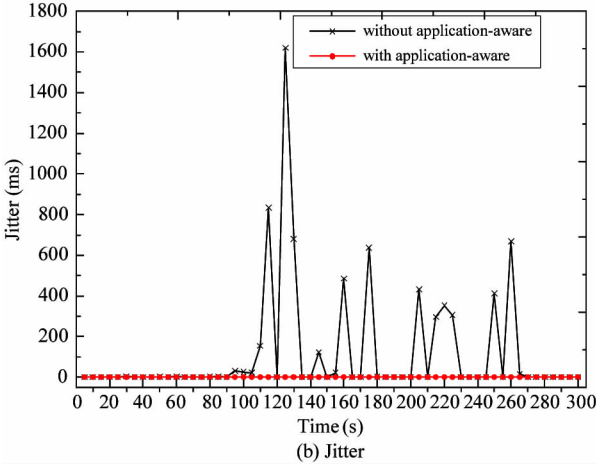
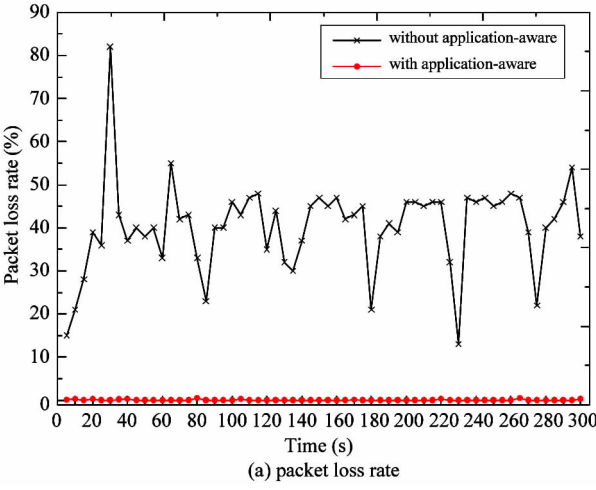


Fig. 7 Performance comparison of two scenarios in high load

5 Conclusions

The best-effort Internet cannot make any promises about QoS for network traffic. With the new paradigm of SDN, many new QoS mechanisms can be presented. In this paper, an application-aware routing architecture

is presented in SDN network to improve QoS of data flow, which shows the advantages of combing application state information with SDN central control. The solution can fully leverage SDN's advantages including well understanding of the network and flexible adjustment of the flows. In order to meet the requested QoS level of flows, it can dynamically calculate the routings for them based on the global network view. To evaluate the performance of the solution, extensive simulations are conducted by using the POX and Mininet. In this simulation, the results verify that the solution can effectively update the forwarding path of data flows and avoid the QoS against caused by network congestion. The proposed application-aware routing mechanism shows obvious improvement on guaranteeing QoS demands of traffic. The future work will be focused on implementing this solution in actual networks.

References

- [1] VerLajan S. Application-aware routing in software-defined networks. [http://www.aricent.com/pdf/Aricent Whitepaper Application Aware Routing in SDN. pdf](http://www.aricent.com/pdf/Aricent%20Whitepaper%20Application%20Aware%20Routing%20in%20SDN.pdf); Aricent, 2014
- [2] Kinsy M A, Cho M H, Shim K S, et al. Optimal and heuristic application-aware oblivious routing. *IEEE Transactions on Computers*, 2013, 62(1): 59-73
- [3] Egilmez H E, Civanlar S, Tekalp A M. An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks. *IEEE Transactions on Multimedia*, 2013, 15(3): 710-715
- [4] Open Networking Foundation. Software-defined networking: the new norm for networks. California, USA: ONF White Paper, 2012
- [5] McKeown N, Anderson T, Balakrishnan H, et al. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 69-74
- [6] Handigol N, Seetharaman S, Flajslik M, et al. Plug-n-serve: Load-balancing web traffic using OpenFlow. In: Proceedings of the 2009 ACM Special Interest Group on Data Communication (SIGCOMM), Barcelona, Spain, 2009
- [7] Koerner M, Kao O. Multiple service load-balancing with OpenFlow. In: Proceedings of the 13th High Performance Switching and Routing (HPSR), Belgrade, Serbia, 2012. 210-214
- [8] Uppal H, Brandon D. OpenFlow Based Load Balancing, Proceedings of CSE561: Networking. Washington: University of Washington, Spring, 2010
- [9] Jain S, Kumar A, Mandal S, et al. B4:experience with a globally-deployed software defined WAN. In: Proceedings of ACM Special Interest Group on Data Communication (SIGCOMM), Hong Kong, China, 2013. 3-14
- [10] Qazi Z A, Lee J, Jin T, et al. Application-awareness in SDN. In: Proceedings of ACM Special Interest Group on Data Communication (SIGCOMM), Hong Kong, China, 2013. 487-488
- [11] Jarschel M, Wamser F, Hohn T, et al. Sdn-based application-aware networking on the example of youtube video streaming. In: Proceedings of the 2nd European Workshop on Software Defined Networks, Berlin, Germany, 2013. 87-92
- [12] Adami D, Antichi G, Garroppo R G, et al. Towards an SDN network control application for differentiated traffic routing. In: Proceedings of the International Conference on Communications, London, UK, 2015. 7456-7461
- [13] Egilmez H E, Civanlar S, Tekalp A M. An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks. *IEEE Transactions on Multimedia*, 2013, 15(3): 710-715
- [14] Lorenz D H, Orda A. QoS routing in networks with uncertain parameters. *IEEE/ACM Transactions on Networking*, 1998, 6(6): 768-778
- [15] Korkmaz T, Krunz M. Multi-constrained optimal path selection. In: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Anchorage, USA, 2001. 834-843
- [16] Ahuja R K, Magnanti T L, Orlin J B. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Inc. 1993
- [17] Schafer I, Felser M. Topology discovery in PROFINET. In: Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, Patras, Greece, 2007. 704-707
- [18] Shin B, Han S Y, Lee D, et al. Application-awareness support for QoS-aware routing protocols in wireless ad-hoc networks. In: Proceedings of the 4th International Conference on the Network of the Future (NOF), POSTECH, Pohang, Korea, 2013. 1-5
- [19] Mccauley J. Pox: A python-based openflow controller. <http://www.noxrepo.org/pox/about-pox/>; Noxrepo, 2010
- [20] Lantz B, Heller B, McKeown N. A network in a laptop: rapid prototyping for software-defined networks. In: Proceedings of the 9th ACM Special Interest Group on Data Communication (SIGCOMM) Workshop on Hot Topics in Networks, New Delhi, India, 2010. 1-6
- [21] Frost V S, Melamed B. Traffic modeling for telecommunications networks. *IEEE Communications Magazine*, 1994, 32(3): 70-81

Zhang Zexin, born in 1987. She is a doctoral candidate at University of Chinese Academy of Sciences. Her research interests include the network traffic analysis, traffic classification and traffic control.