Handling Spatio-Temporal QoS Requests for Dynamic Network Provisioning

Huu-Duy Nguyen
Home Appliance Research Institute
VinSmart Research and Manufacture Joint Stock Company
Hanoi, Vietnam
Email: huuduyce@gmail.com

Van-Quyet Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
Email: quyetict@utehy.edu.vn

Hong-Nam Quach
Department of Electronics and Computer Engineering
Chonnam National University
Gwangju, South Korea
Email: quachhongnam1995@gmail.com

Kyungbaek Kim
Department of Electronics and Computer Engineering
Chonnam National University
Gwangju, South Korea
Email: kyungbaekkim@jnu.ac.kr

***Abstract*—** Providing the flexible and personalized network services, which are suitable for the requirements of customers has become an essential issue for network service providers. In this paper, we propose a method for dynamic network provision which considers users' spatio-temporal QoS requests including location, usage duration and QoS. We conducted several mininet-based experiments with various network settings and various user request settings, and through this evaluation we observed that our proposed method accepts more spatio-temporal QoS requests than a method called LARAC.

Keywords — software-defined networks, network provisioning, dynamic routing, Spatio-temporal QoS, location-aware network

I. INTRODUCTION

In recent years, with the wide development of the network infrastructures, services as well as the high demand for network QoS, the dynamic network provisioning is provided for users in order to help network service providers in supporting their customer more flexible and personalized network services accordingly. However, more and more demands come in terms of their advanced networks even though their network resources and infrastructure are finite. They have to not only take care of QoS but also the locations and durations of usage from user requests. This fact has raised a question of how to leverage the limited network resources more efficiently to support personalized network services and response to a variety of user requests with various constraints along with QoS constraints and locations and durations of user requests. In order to deal with this problem, network service providers have been seeking a framework which can utilize their given network resource and be able to accept as many spatio-temporal QoS requests as possible, not mentioned to maximizes the network output.

Besides, the Software-defined Networking (SDN) is known with a lot of merits, by decoupling network control block from underlying routers. SDN leverages a logically centralized controller and realizes the programmability of the network. Additionally this separation between the control plane and the data plane also provides the way for dynamic control and management of packet forwarding and processing in switches, which help to management the entire network more easily and improve the utilization of network capacity and the performance of networks in terms of delays and losses [1].

In this paper, we propose a method to create network topologies which meet users' STQ (Spatio-Temporal QoS) requirements, dynamically. To handle STQ requests, a time-slice concept is used for finding the shortest path between any two locations in terms of delay and bandwidth constraint. In addition, we designed a web application interface through which users can generate STQ requests. Besides, we also created a program to receive and handle users' STQ requests and applying the generated network topology into mininet dynamically.

II. RELATED WORK

In our previous studies [3], [4], we have presented an implementation of a framework for location-aware dynamic network provisioning. This framework supports obtaining users' requirements such as locations and QoS, mapping the requested locations with the network infrastructure, selecting routes to meet the requested QoS and deploying a prepared network service into SDN (Software-defined Networking) enabled network infrastructure. Besides, M. Huang et al. proposed online algorithms with an auxiliary graph for unicast and multicast requests including a bandwidth constraint, and maximized network throughput.[5] In another research, W. Liang et al. also adopted algorithms with an auxiliary graph for NFV-enable request including bandwidth and delay constraints in order to minimize the operational cost and maximize the network throughput.[6] Although the above studies have solved the QoS constraints of user's requests, they have not considered STQ which takes the locations and durations of user requirements supposing that several requests are delivered at different times or they arrive at the same time.

Hence, how to do so for network providers to decide whether requests are accepted within their current network resources has become a vital question. Because of these existing limitations, this paper proposes an approach to accept STQ requests that are independent of quantity and maximized the network throughput with the given network assets.

III. SPARTIO-TEMPORAL QUALITY OF SERVICE REQUEST

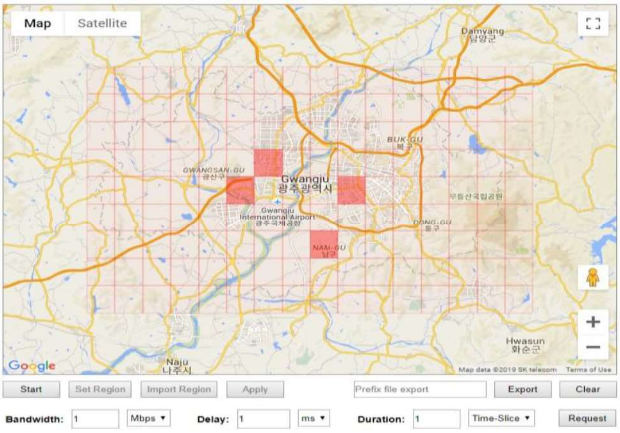
In this section, we firstly introduce a concept of spatio-temporal quality of service requests which is generated by users. Then we also describe how we are taking these requests from users:

A. *Spatio-Temporal QoS Request (STQ Request)*

A spatio-temporal QoS request is a user request which is composed by (1) a list of requested locations from the user for deploying their own network service, (2) a QoS constraint level for service (bandwidth, delay) and (3) a requested time duration for using network resources. For example, a university has several campuses which are located in some different places (in a city or a country), they want to open several online classes for their students from all campus and each online class lasts in a specific duration within a day. This needs to make an STQ request including a set of locations in which each campus is located, the time duration of an online class and QoS guarantee for online video streaming while a class is opening.

B. *Web application interface for taking STQ Request*

In this part, we describe our web application interface which supports taking STQ requests from the user and pushing these requests to the central server for request handling as Fig. 1. The following figure is a map-based region selector Web UI in which a user can pick their desired locations. Also, through this Web UI, users specify required QoS level through network parameters such as bandwidth and delay as well as indicate the time duration for the requested network servie.



Bandwidth: 1 Mbps
Delay: 1 ms
Duration: 1 Time-Slice

Fig. 1. Web-based application for capturing user requests

IV. TIME-SLICING BASED NETWORK TOPOLOGY PLANNING

In this section, we present the system model and notations and define the problem precisely. The, we show how to generate network topology plans for STQ requests.

A. *System model*

We represent a software-defined network as a graph $G = (V, E)$, in which V is considered as a set of OpenFlow (OF) switches, and E is a set of network links connecting the OF switches. Assuming that there is an SDN controller in network G which is responsible for setting up and installing the forwarding rules into the flow table in switches. Each switch $v \in V$ can hold totally F_v flow entries and each link $e \in E$ incurs two QoS constraints including delay ∂_e and bandwidth capacity β_e

B. *STQ requests*

We assume that STQ requests come in the system one by one. Each k^{th} request is denoted by $RQ_k = (L_k, B_k, D_k, P_k, T_k)$ where L_k is a list of switches mapped with the users' requested locations, B_k is bandwidth-constraint, D_k is delay-constraint, P_k indicates a specific start time for using network resources and T_k indicates the duration of usage. For simply, L_k is represented by (u_k, v_k) in which u_k, v_k are two switches in L_k determining source switch and destination switch accordingly.

C. *Time-slice concept*

For handling STQ requests, a time-slice concept is employed for finding the shortest path between two switches in terms of bandwidth and delay constraints. A time-slice means a divided time window with each subdivided time unit. The objective of employing the time slice is to provide a basic time unit to process STQ requests. That is, a network plan for an STQ request is supported with multiple segmented network paths on different time slice. To do this, we split the total time of one day to a number of time slices Δt or $N.\Delta t$, so each time slice:

$$t_{i+1} = t_i + \Delta t \qquad \forall i \in [0, N) \tag{1}$$

Denote by $\beta_e(t_i)$ the available bandwidth on each link $e \in E$ and $\omega_e(t_i)$ as the weight of link e at time slice t_i

$$\omega_e(t_i) = f(\beta_e(t_i)/\beta_e) \quad 0 \le \omega_e(t_i) \le 1 \tag{2}$$

Similarly, $\beta_e(t_i)$ is the number available entries in each switch $v \in V$ and $\omega_e(t_i)$ is the weight of switch v at time slice t_i

$$\omega_v(t_i) = f(F_v(t_i)/F_v) \quad 0 \le \omega_v(t_i) \le 1 \tag{3}$$

Within the time-slice concept as above, parameters of RQ_k now are equivalent as follows. P_k is corresponding to a specific time t_i while T_k is multiple of Δt ($T_k = K.\Delta t$)

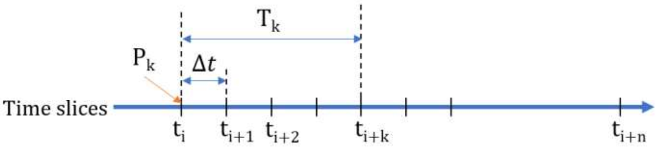


Fig. 2. Illustration parameters of RQ_k within timthe e-slice concept

D. Problem definitions

Given a software-defined network $G = (V, E)$ where V is a set of OpenFlow switches and E is a set of network links connecting the OF switches in the network G. With the assumption that a number of STQ requests come in the system in a sequence. Some of these requests to use network resources at the same time and the others to use at different times. The network service providers need to decide which requests could be accepted while their network resource usage is maximization. In other words, how a network provider can accept as many STQ requests as possible and maximize the network throughput with the given network resource.

E. Network topology planning

In a traditional network, a network path between two anonymous switches is usually fixed until the network topology changes. By the support of the dynamic network, the network path could be adjusted due to the occurrence of several configured conditions even though the network topology does not fluctuated. Most of those conditions relate to the weights of network links between two adjacent switches in the network. A popular approach to weigh a network link is based on network parameters such as the bandwidth of the link, the delay on the link, switch's flow capacity and so on. In fact, when the network topology makes no change, the network path connecting two switches trends to maintain if it is reserved for a user request during the request's duration. In the case of the location-aware network, more and more STQ requests are treated by the same network path when they request connecting two same locations. Consequently, some network links are exhausted while other links remain abundant bandwidth capacity, especially, the network links are shared by a large number of requests.

To deal with this problem, instead of using only one network topology for a request, we generate a network topology for a request in each time-slice within its duration as shown in Fig. 3. It means that if a request duration is divided into N time-slices, this request is treated by N network topology correspondingly. To do this, we are finding the shortest paths in terms of the sum weight of switches and links with a delay constraint. We assume that $P'_{(i+j)}(k)$ is the shortest path between u_k and v_k at time slice t_{i+j}, then the sum weight of $P'_{(i+j)}(k)$ is determined as (4) so that (5) is satisfied.

$$\sum\nolimits_{e',v'\in P'}(k) = [\omega_{e'}(i+j) + \omega_{v'}(i+j)] \qquad (4)$$

$$\sum\nolimits_{e'\in P'}\partial_{e'}(i+j) \le D_k \qquad (5)$$

where $j \in [0, k]$ and $t_k \equiv t_i$

The network topology of request RQ_k at each time slice consists of every network path between two arbitrary switches at corresponding time slice. This topology is generated and represented by a number of forwarding rules by the network controller server. Then, the controller imposes these rules to all the switches which belong to the topology of the request.

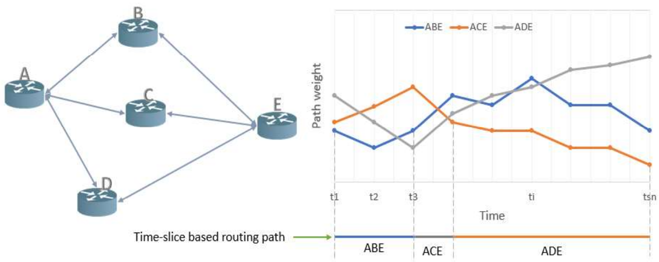


Fig. 3. An example of a time-slice based routing method

We describe the proposed method at each time slice within the duration consisting of following main steps:

- Firstly, we capture the current state of network resources.
- Secondly, we check the bandwidth capacity and delay of every link and the flow table size in every switch. Then, we eliminate the links which do not meet the user QoS requirements from consideration for further network planning.
- Thirdly, we employ the LARAC algorithm [2] for finding the shortest path between every pair of switches.
- Next, we temporarily update the weight of links and switches which are accepted for the request and repeat all the above steps for the next time slice.

Finally, we determine that a request is accepted officially by the system if we can find every network topology at every time slice. Until now, the weight of links and switches are also updated officially. Note that, a network topology for a request is a combination of all network paths between every pair of switches. A request is rejected if we can not find any network topology at any time slice and our algorithm stops immediately.

V. EVALUATION

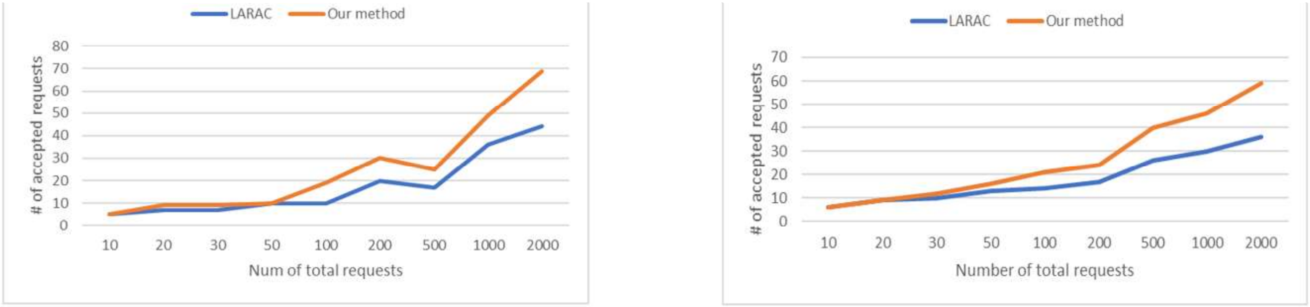
To evaluate the proposed method, we conducted several experiments with various network settings and various user request settings by using mininet. Then, we measure the number of accepted requests and accumulated bandwidth.

A. Evaluation settings

In these experiments, we vary switch network size in 30, 50, 100, 150, and 200, and each network link has a bandwidth capacity in the range of 100 ~ 1000 Mbps and 2~5ms delays. We assume that a switch can hold up to 1000 flow entries. For a spatio-temporal QoS request, we randomly pick 20% of switches in a request and randomly choose start time and duration of usage. The QoS parameter of a request includes bandwidth constraint (1~10Mbps) and delay constraint in range (40ms ~ 200ms).

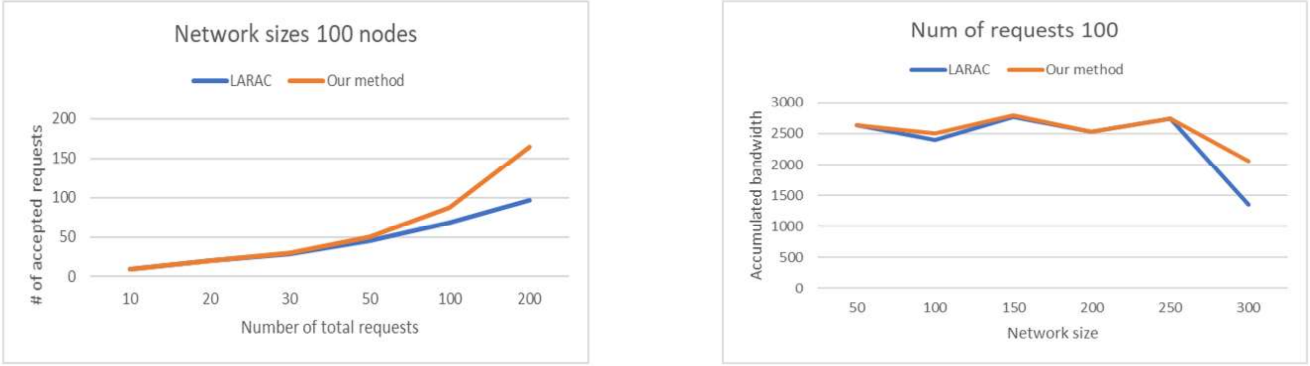
In order to evaluate the performance of our proposed method, we here propose a heuristic algorithm that also employs LARAC but the network topology is conserved during the requested duration.

Fig. 4. The performance of proposed method against origin LARAC algorithm



a) The number of accepted requests by with network sizes 30 nodes

b) The number of accepted requests by with network sizes 50 nodes



c) The number of accepted requests by with network sizes 100 nodes

d) The accumulated bandwidth by 100 requests

B. Results and Analysis

We evaluate the proposed method by varying network size from 30 to 100 and the number of requests increases up to 2000 while other parameters are fixed. Fig. 4 plots the performance curves of two different methods from which it can be seen that the proposed algorithm and origin LARAC algorithm. Our proposed method outperforms LARAC in all cases. Specifically, at the very first of simulation time, when the number of requests is quite small (less than 50), the number of accepted requests seems to be the same between the two methods. However, if the number of requests continues increasing, the number of requests accepted by our proposed method is more than the other marginally up to 1.5 times at the end of the simulation. In other words, the more requests come in the system, our proposed model accepts the more requests than LARAC as Fig. 4(a, b, c). It is because, in LARAC, when the network resource serves a request, it cannot be used for other requests until its duration finishes. However, with our proposed method, the network resource can be shared by several requests in a specific common time-slice of them.

In addition, Fig. 4 (d) indicates that our proposed method delivers more accumulated bandwidth than the LARAC method, especially under larger networks. We observe that the accumulated bandwidth of the proposed method goes down less than the LARAC method as the network size increases from 250 to 300. The reason is that the proposed method can accept more requests whose QoS requirements are small while the LARAC method resource is saturating.

VI. CONCLUSION

In this paper, we implement a web-based application for capturing STQ requests from the user. Next, we maintain a server for request handling. This server helps in implementing our proposed method in order to collect the network resource, query the network path for the user request at each time-slice and generate a network topology matching with the requirements of the request. Currently, we try to enhance the performance of network provisioning in the aspect of the number of accepted requests and QoS constraints. Especially, we will extend this work with machine learning techniques such as reinforcement learning.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1A2B4012559).

REFERENCES

[1] S. Agarwal, M. Kodialam, and T. V. Lakshman. “Traffic engineering in software-defined networks,” in Proceeding of IEEE INFOCOM, 2013.

[2] A.Juttner, B.Szviatovszky, I.Mecs, Z.Rajko, “Lagrange relaxation-based method for the QoS routing problem,” in Proceeding of IEEE INFOCOM, vol. 2, April 2001, pp. 859-868.

[3] Van-Quyet Nguyen, Sinh-Ngoc Nguyen, Deokjai Choi, Kyungbaek Kim, “Location-aware Network Provisioning,” in Proceeding of APNOMS 18th Conference, Korea, 2017, pp. 239-242.

[4] Huu-Duy Nguyen et, al “Implementation of a Framework for Location-aware Dynamic Network Provisioning”, in Proceeding of KIPS Conference, Korea 2018, pp. 133-135.

[5] M.Huang et al, “Dynamic routing for network throughput Maximization in Software-defined networks,” in Proceeding of IEEE INFOCOM, USA, 2016, pp. 1-9.

[6] Mike Jia, W.Liang, M.Huang, Z.Xu and Yu Ma, “Routing Cost Minimization and Throughput Maximization of NFV-enabled Unicasting in Software-defined networks,” IEEE Transaction and Network service Management, vol. 15, no. 2, pp. 732-745, June 2018.