Smart Media Journal / Vol.11,No.5 / ISSN:2287-1322
https://dx.doi.org/10.30693/SMJ.2022.11.5.82

Link Stability aware Reinforcement Learning based Network Path Planning

Hong-Nam Quach*, Hyeonjun Jo†, Sungwoong Yeom**, Kyungbaek Kim***

Abstract

Along with the growing popularity of 5G technology, providing flexible and personalized network services suitable for requirements of customers has also become a lucrative venture and business key for network service providers. Therefore, dynamic network provisioning is needed to help network service providers. Moreover, increasing user demand for network services meets specific requirements of users, including location, usage duration, and QoS. In this paper, a routing algorithm, which makes routing decisions using Reinforcement Learning (RL) based on the information about link stability, is proposed and called Link Stability aware Reinforcement Learning (LSRL) routing. To evaluate this algorithm, several mininet-based experiments with various network settings were conducted. As a result, it was observed that the proposed method accepts more requests through the evaluation than the past link annotated shorted path algorithm and it was demonstrated that the proposed approach is an appealing solution for dynamic network provisioning routing.

Keywords : SDN | network provisioning | reinforcement learning | QoS demand constraints | link stability

I. INTRODUCTION

Recently, with the rapid development of network infrastructures and services and the increased demand for network QoS, dynamic network provisioning has been made available for users to assist network service providers in providing more flexible and personalized network services to their customers[1]. However, because of the harmful effects of the COVID-19 pandemic, people are unable to leave their homes, increasing the pressure on customers to join their advanced networks, despite the fact that their network resources and infrastructure are limited. As a result, they must address the unique requirements contained in user requests. This fact has prompted research on how to leverage limited network resources more efficiently to support personalized network services and respond to various user-specific requests with varying constraints and a guarantee of QoS while also considering the locations and durations of a user request. Therefore, network service providers have been looking for a framework that can best use their available network resources while accommodating as many specific requirements as possible to address this issue.

* This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(No.2019-0-01343, Regional strategic Industry convergence security core talent training business)

* Master student, Department of Artificial Intelligence Convergence, Chonnam National University.
† Master student, Graduate school of Information Security, Chonnam National University.
** PhD student, Department of Artificial Intelligence Convergence, Chonnam National University.
*** Professor, Department of Artificial Intelligence Convergence, Chonnam National University.

Manuscript : 2022. 03.07
Revised : 2022. 04. 10
Confirmation of Publication: 2022. 04. 22
Corresponding Author : Kyungbaek Kim e-mail : kyungbaekkim@jnu.ac.kr

Additionally, Software-defined Network (SDN) is well-known for its numerous benefits, including decoupling the network control block from the underlying routers, switching to a logically centralized controller, and greater network the consequent ease in programming networks. A Reinforcement Learning-based approach for intelligent routing decisions in the SDN environment is proposed in this paper, dubbed Link Stability aware Reinforcement Learning (LSRL) routing. The result demonstrates that the LSRL algorithm can discover, learn, and utilize potential routing paths efficiently, albeit network traffic changes dynamically.

Ⅱ. RELATED WORK

A framework for location-aware dynamic network provisioning is presented in papers[2-4]. This framework enables the gathering of user requirements such as location and QoS, mapping the requested locations to the network infrastructure, selecting routes that meet the requested QoS, and deploying a prepared network service into the SDN enabled network infrastructure via the network controller. Also, path-finding in a location-aware network has been the subject of numerous studies in recent years.

The authors of [5,6] proposed online algorithms for unicast and multicast requests that incorporate a bandwidth constraint and maximize network throughput. However, these researches do not take into account the location specific information associated with user requests. Also, a method for accepting customer requests (Spatial-Temporal QoS requests) includes the location of customer, use time, and QoS deployed in [7]. Nonetheless, these studies have not addressed how to troubleshoot network problems that occur unexpectedly, such as node congestion, damaged connection devices, broken devices, or natural disasters. Furthermore, the Internet is a complex network whose state is constantly changing. As a result, network providers must quickly calculate and identify alternate routes that do not negatively impact the customer when congestion occurs.

Software-defined Network (SDN) is well-known apart from simplifying policy enforcement, network configuration, and evolution. This separation of the control and data planes enables dynamic control and management of packet forwarding and processing in switches, which is expected to simplify network management and improve network capacity utilization delay-and-loss performance.

Reinforcement learning[8] consists of an agent and an environment, and the agent obtains an optimal policy by interacting with the environment through the Markov Decision Process (MDP) [4]. Reinforcement learning can be classified into policy-based reinforcement learning and value-based reinforcement learning. Policy-based reinforcement learning only makes decisions based on policies. That

is, the policy simplifies the policy itself in the approximator function using the policy gradient and updates the objective function in the direction of increasing. Such policy-based reinforcement learning does not respond to changing values. On the other hand, value-based reinforcement learning makes decisions based on a function that derives a value for a particular state. Value-based reinforcement learning is easy to calculate because it selects a policy based on the value, but only one policy can be selected, and even a small change in the value causes a large change in the policy. However, as networks become more and more complex in recent years, the importance of the value chain-based network is increasing compared to the policy network.

Q-Learning [5] is a value-based reinforcement learning method that focuses on the Q function to find the Q value and selects a policy based on it. Q-learning uses the Bellman Equation to find the value function for the next state through a recursive function. That is, the action that maximizes the sum of the present value and the future value is selected, the value is obtained, and the action that maximizes the value is selected. In this paper, we propose a routing algorithm based on Q-Learning for value chain-based network implementation.

Algorithm 1. LSRL Routing Algorithm

INPUT:
Learning rate: α (α = 0.8)
Episodes parameter: n
S_R = {P_{S_R}} pair of switches in S_R. P_{S_R} = (src, dst)
Network information – link-state
OUTPUT: List of the paths to connect the switches in S_R
Procedure:
Choose any P_{S_R} in S_R and find the route to connect
for each (src, dst) ∈ P_{S_R} **do**
Initialize Q: $Q(S, A) = 0$, $\forall s \in S$, $\forall a \in A$
for *episode* ← 1 **to** n **do**
Start: $St = src \in S$;
while S_{t+1} *is not dst* **do**
Select A_t for S_t with policy π from Q using the ε-greedy method of exploration and exploitation;
$R_{t+1} \leftarrow R(S_t, A_t)$ // Agent gains the reward and observes next state S_{t+1} ;
$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha * \left[R_t + \max_A Q_t(S_{t+1}, A) - Q_t(S_t, A_t)\right]$
$S_t \leftarrow S_{t+1}$ // Go to next state;
end
end
Get Q-table, find the paths for P_{S_R} with state-action pairs that gained the min Q-values
End
Store the set of paths for all node–pairs in the network

III. LSRL Routing Algorithm

In this paper, we propose a Q-learning-based Link Stability aware Reinforcement Learning (LSRL) algorithm for value chain-based network management. The LSRL algorithm uses learning to determine the optimal paths between each pair of nodes in the data plane by considering the network link stability. The details of LSRL algorithm is described in Algorithm 1.

The input parameters of algorithm are the learning rate, the general parameters, the number of episodes to learn, the

source destination nodes pairwise, and information about the connections between them. The output contains a list of paths linking the devices in the highest-rewarding node pairs. In addition, the optimum value is used to generate the Q-table for the state-action pairs of routing algorithm. As a result, the LSRL algorithm determines the shortest path between any two network nodes under any certain network status.

First of all, Q-table is initialized with zeros. Then, LSRL algorithm run "n" episodes to complete Q-table by conducting following procedures. After initializing the Q-table, the LSRL algorithm starts with the initial state of node "src". Then, the agent chooses the next node in the "Action Space" and the discovery and exploitation process to choose a neighbor node from the neighbors of current node as the next hop by using ε -greedy. Next, the agent manipulates the link state and the state "S_t" to determine the specific goal with the action "A_t" and then recognizes the following state "S_{t+1}".

Next, equation (1) is used to compute the reward. The link stability calculations were made using three parameters collected from customer requirements, including bandwidth (*brq*), delay (*drq*), and packet-loss (*lrq*), and then expressed as equation (1), where *brq*, *drq*, *lrq* are the normalization values for bandwidth, delay, and packet loss, respectively.

$$R = \theta_1 * brq + \theta_2 * (1 - drq) + \theta_3 * (1 - lrq) \quad (1)$$

The reward is proportional to the amount of bandwidth requested, while the penalty is inversely proportional to the delay and packet drop required. Besides, these parameters serve as metrics for link stability. This could explain why the link stability metric in the link state is positive [9]. The values θ_1, θ_2 and θ_3 are parameters representing weights for the compensation matrix. The weights are denoted by equation (2).

$$\theta_1+\theta_2+\theta_3 = 1,\ \theta_1, \theta_2, \theta_3 \in [0, 1] \quad (2)$$

In equation (1), the requirements for bandwidth, latency, and packet loss are normalized. The Min-Max method normalizes each feature in the learning process of agent, as each parameter is composed of distinct units[10]. The equation below illustrates an example of normalization (3).

$$\bar{x_i} = \frac{x_i - min(x)}{max(x) - min(x)}, x_i \in X \quad (3)$$

Each normalized value is calculated using equation (3). (requests of bandwidth, delay, and packet loss). In equation (3), X is represented by the collection of values used to normalize, and x_i is the normalized value Following that, the agent considers the learning rate metrics, the reward, and the new state and then revises the Q-function using equation (4).

$$Q_{t+1}(S_t,A_t) = Q_t(S_t,A_t) + \alpha*[R_t + max\ Q_t(S_{t+1},A_t) - Q_t(S_t,A_t)] \quad (4)$$

It then transitions to the next state, one episode concludes, and the next episode begins. Finally, after the agent completes a conversion, it uses the result as a Q-table based on the state-action tuples with the highest Q values to determine the optimal route between the "src" and "dst". After determining the optimal path for each source-destination node pair, the agent stores the results in a routing table; the controller SDN then retrieves and forwards these optimal paths to the routing tables of the devices.

Ⅳ. Evaluation

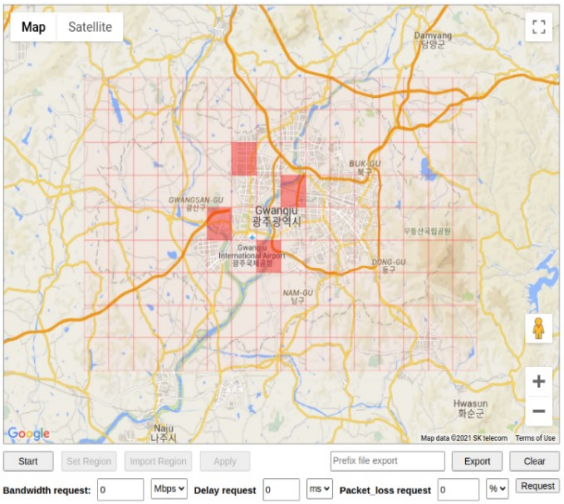


Fig. 1. Web-UI for taking user requests

This section describes the evaluation of the proposed LSRL routing algorithm. For evaluation a series of experiments with various network configurations using Mininet were conducted while also deploying the Web UI to support obtaining the requirements of users. Users can select locations, and network parameters such as bandwidth, latency, and packet loss ratio in order to decide necessary QoS level.

4.1 Evaluation setting

Table 1. Configuration of experimental environment

Configuration	List
Operating System	Ubuntu 20.04.2.0 LTS
Memory	RAM 8GB
Programming language	Python3.8, NodeJS, Socket.io, HTML
Libary	Python3.8, MySQL server, Mininet, Numpy, Pandas
Database	MySQL server
Simulator	Mininet 2.2.3
protocol	OpenFlow1.3, OpenvSwitch2.3.1
Controller	Mininet API Contronller

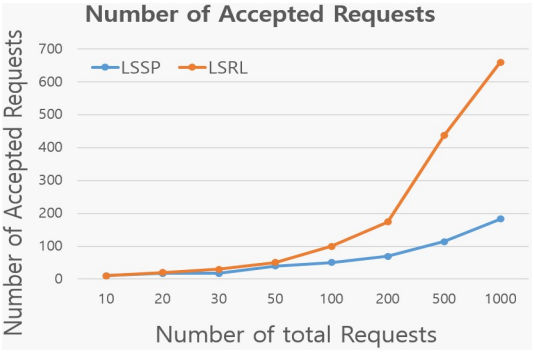
It is assumed that the network resource has 23 switches. The topology was implemented by using Mininet which has 23 nodes and 37 links by using a Python script. In experiments, setting up the link capacity topology on each network link has a bandwidth capacity in the range of 100~1000 Mbps, 2~5ms delays, and a packet-loss ratio arranged from 0~10%. Also, 50% of connections were set up with a capacity bandwidth of 1000 Mbps, and the packet loss is 0%. 40% of links have bandwidth 500 Mbps with packet loss of 5%, and 10% connection of topology with 100 Mbps and 10%. Experiments randomly pick 20% of devices in a request and randomly choose the start time and usage duration. The QoS parameter includes bandwidth constraint (1 ~ 10Mbps) and delays in range (40ms ~ 200ms). Table 1 shows configuration of experimental environment.

4.2 Traffic Creation

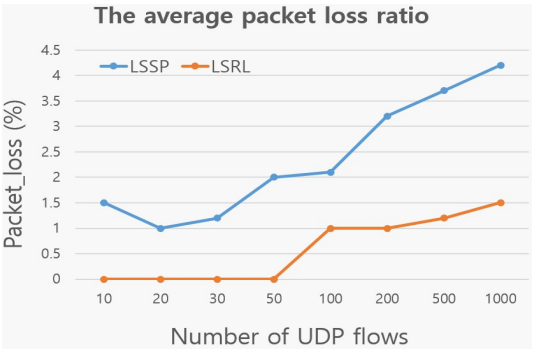
To generate traffic in the Mininet-based emulation, the tool iperf3 [11] was used, scripts were developed to run iperf3 clients and servers on the hosts. Since iperf3 allows for the specification of a target transmission rate per connection, UDP traffic was generated. The experiment included the complete execution of iperf3 scripts that generated traffic according to a traffic matrix. In addition, traffic was generated between pairs of nodes using sixteen publicly available intra-domain traffic matrices.

4.3 Results and Analysis

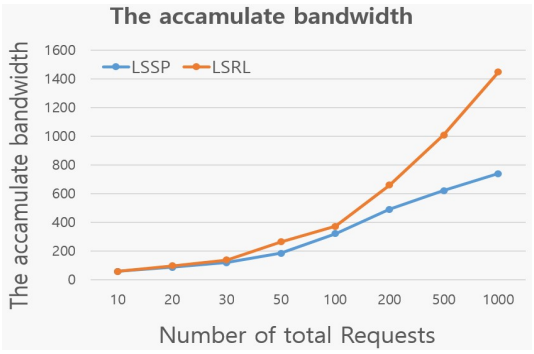
Fig. 2.(a) compares the performance curves of the Link Stability aware Reinforcement Learning (LSRL) and Link Stability aware Shortest Path (LSSP) algorithm, demonstrating that when the number of requests increases, the LSRL approach consistently outperforms LSSP. Here, LSSP routing algorithm means Link Stability aware Shortest Path routing algorithm, which is a kind of deterministic routing algorithm with annotated link stability information. To be more precise, during the initial period of the test, the total number of accepted requirements appears identical for the two methods when the number of solicitations is relatively small (less than 50). However, as the number of requests increases, the LSRL accepts slightly more than LSSP and the acceptance of request becomes twice at the conclusion of simulation. when network resource is assigned to one request, it cannot be used for other



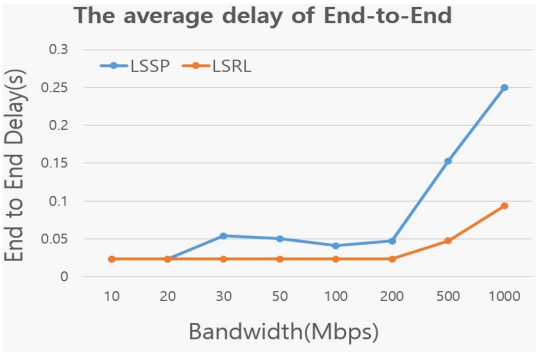
(a)



(b)



(c)



(d)

Fig. 2. The result of comparison between the proposed method: LSRL and LSSP (a) : The number of accepted requests, (b) : The accumulated bandwidth, (c) : The average delay of End-to-End, (d) : The average Packet loss ratio

requests until the duration of request expires. However, the LSRL algorithm can increase the number of requests accepted by the network by identifying alternate paths identical to the shortest path.

Fig. 2.(b) illustrates proposed method providing a greater total cumulative bandwidth than the LSSP method, despite the difference being relatively small. It was observed that as the no of as the number of requests increases frequently, the result difference widens significantly indicated that. Until the simulation is completed, the cumulative bandwidth of the two methods is nearly equal. This is because the resource is limited. Since the resource begins to saturate, the proposed method can accept more requests with constant QoS requirements but constant cumulative bandwidth.

We use the iperf tool to evaluate the end-to-end delay and the packet loss ratio of the two different routes of LSRL and LSSP. Frist, we generate the user request with the source and destination. Then We deploy both LSRL and LSSP to routing from src to dst. LSSP is based on the cost of finding the shortest path, a hop count lower than LSRL is selected for the shortest path in LSSP. However, low-capacity routes, result in congestion and traffic concentration on certain routes. On the other hand, based on link stability LSRL may have more hop counts than LSSP. but it is possible to reduce the end-to-end delay and packet loss ratio compared to LSSP, as shown in Fig. 2.(c) and (d).

As a result, when the number of requests, flows, and requested bandwidth increase, it can be seen that the network management performance deteriorates due to the policy that does not avoid the section in which the congestion level increases in the LSSP. On the other hand, in the proposed value chain-based LSRL, network management efficiency can be improved by securing a probabilistic possibility to fully utilize the capabilities of limited resources.

V. Conclusion

The LSRL algorithm, an RL-based solution with link stability, was proposed in this paper for intelligent and efficient routing in dynamic network provisioning. The experimental results comparing the proposed LSRL to LSSP algorithm indicate that LSRL outperforms LSSP algorithm. LSRL generated more shorter paths than other algorithms this allows for the acceptance of numerous user requests while maintaining various levels of QoS constraints, whereas LSSP algorithm generates only one shortest route. The reward minimization technique enables the agent to discover, learn, and exploit the most efficient routes for accepting requests, maximizing resource utilization.

Due to the fact that both the proposed method and the existing LSSP routing scheme are based on the cost function for routing and making routing decisions, neither method takes temporal complexity into account. However, as a

matter of fact, the complexity of time is a critical point. As a result, it could explain why the cost of activating a new service can be reduced if the time complexity is reduced. In the future, It is intended to investigate a technique for augmenting LSRL with Deep Reinforcement Learning (DRL) to enhance LSRL based routing decision-making capabilities.

REFERENCES

[1] Wellman, Barry, "Physical place and cyberplace: The rise of personalized networking," *International journal of urban and regional research*, Vol. 25, No. 2, pp. 227-252, Jun. 2021.

[2] Nguyen, Van-Quyet, et al., "Location-aware dynamic network provisioning," *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE*, Seoul, Korea, Sep. 2017.

[3] Gde, Dharma N., et al., "Design of service abstraction model for enhancing network provision in future network," *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE*, Kanazawa, Japan, Oct. 2016.

[4] Quach, Hong-Nam, Chulwoong Choi, and Kyungbaek Kim, "Dynamic Network Provisioning with AI-enabled Path Planning," *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE*, Daegu, Korea, Sep. 2020.

[5] Huang, Meitian, et al., "Dynamic routing for network throughput maximization in software-defined networks," *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE*, San Francisco, USA, Apr. 2016.

[6] Jia, Mike, et al., "Routing cost minimization and throughput maximization of NFV-enabled unicasting in software-defined networks," *IEEE Transactions on Network and Service Management*, Vol. 15, No. 2, pp. 732-745, Jun. 2018.

[7] Nguyen, Huu-Duy, et al., "Handling Spatio-Temporal QoS Requests for Dynamic Network Provisioning," *Proceedings of the 2019 KICS Korean-Vietnam International Joint Workshop on Communications and Information Sciences*, Hanoi, Vietnam, Nov. 2019.

[8] Quach, Hong-Nam, Sungwoong Yeom, and Kyungbaek Kim, "Survey on Reinforcement Learning based Efficient Routing in SDN," *The 9th International Conference on Smart Media and Applications*, Jeju, Korea, pp. 196-200, 2020.

[9] Jabraeil Jamali, Mohammad Ali. "A multipath QoS multicast routing protocol based on link stability and route reliability in mobile ad-hoc networks." Journal of Ambient Intelligence and Humanized Computing, Vol. 10, No. 1, pp. 107-123, Jan. 2019.

[10] Al Shalabi, Luai, and Zyad Shaaban. "Normalization as a preprocessing engine for data mining and the approach of preference matrix," *2006 International conference on dependability of computer systems, IEEE*, Szklarska Poreba, Poland, May. 2006.

[11] Radoglou-Grammatikis, Panagiotis, et al. "DIDEROT: An intrusion detection and prevention system for DNP3-based SCADA systems," *Proceedings of the 15th International Conference on Availability, Reliability and Security*, New York, USA, pp. 1-8, Aug. 2020.

Authors

Hong-Nam Quach

He received the MS degree in the Artificial Intelligence Convergence at the Chonnam National University, South Korea. His research interests are the implementations of flow control, routing, and Software Defined Networking with deep learning.

Hyeonjun Jo

He received the BS degree with double major in Information E-commerce and Digital Forensics from Wonkwang University, South Korea, in 2020. He is currently a MS student in Interdisciplinary Program of Information Security, Chonnam National University, South Korea. His research interests are in the design and implementation of routing algorithm in software defined networking with reinforcement learning, ransomware analysis among malicious codes.

Sungwoong Yeom

He received the BS, MS in the Department of Electronics and Computer Engineering from Chonnam National University, South Korea, in 2019 and 2021, respectively. He is currently an PhD in the Artificial Intelligence Convergence at the Chonnam National University, South Korea. His research interests are the implementations of network anomaly detection, flow control, routing, GRID/Cloud system and Software Defined Networking with deep learning.

Kyungbaek Kim

He received the BS, MS, and PhD degrees in electrical engineering and computer science from Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1999, 2001, and 2007, respectively. He is currently a Professor in the department of Artificial Intelligence Convergence at Chonnam National University. Previously, he was a postdoctoral researcher in the department of Computer Sciences, University of California, Irvine, CA, USA. His research interests are Intelligent Distributed System, Software Defined Network/Infrastructure, Bigdata Platform, GRID/Cloud system, Social networking system, AI applied Cyber Physical System, BlockChain and other issues of distributed systems. He is a member of ACM, IEEE, IEICE, KIISE, KIPS, KICS, KIISC, KISM and so on.