사용자 요청을 통한 강화 학습 기반 네트워크 프로비저닝

홍남 곽, 염성웅, 김경백
전남대학교 전자컴퓨터공학부
email: quachhongnam1995@gmail.com, yeomsw0421@gmail.com,
kyungbaekkim@jnu.ac.kr

Reinforcement Learning based Network Provisioning with User Requests

Hong-Nam Quach, Sungwoong Yeom, Kyungbaek Kim
Dept of Electronics and Computer Engineering,
Chonnam National University
email: quachhongnam1995@gmail.com, yeomsw0421@gmail.com,
kyungbaekkim@jnu.ac.kr

요 약

In this paper, we propose an approach for finding the shortest paths based on reinforcement learning. Traditional dynamic routing algorithms used in the network layer, link-state routing, and distance-vector routing, utilize Dijkstra or Bellman-Ford's algorithm, respectively. These network routing algorithms are susceptible to overloading popular links present in multiple shortest-path trees, making these links severe bottlenecks for network performance. As a result, Q-routing, a reinforcement learning algorithm, can be applied to alter a packet's route on-the-fly using estimates of the delay along each path. This implemented in networks through acknowledgments that back-propagate minimum router-to-destination packet delays.

1. Introduction

Nowadays, along with the steady development of the 5G network, the demand for quality personalizing network services is increasing because of this reason, providing flexible and personalized network services, which are become an important issue and an exciting topic for network service providers. However, the needs of users for an advanced service of the network increased regularly. Not only do users care about the quality of the service, but they also concern about the location and time to use it comfortably. This fact raises the question of how to leverage the limited network resources to provide personalized network services and respond to a variety of user requests with various constraints, for example, QoS guarantee, selecting locations, and duration. Besides, the Software-defined Networking (SDN) known with many merits by decoupling the network control block from the underlying routers and switching to a logically centralized controller and make the programmability of the network.

Additionally to simplifying policy enforcement, network configuration, and evolution, such a 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 expected to ease network management and improve network capacity utilization as well as delay-and-loss performance [1]. In addition to it, machine learning is gaining much attention from academia and industry due to its outstanding accomplishment of large-scale data processing, organization, and smart decision-making. Some studies use it to solve the deadlock problem in network operation and management [2], [3]. Some works use its intelligent algorithm to achieve smart, customizable, and detailed routing management for SDN.

Moreover, Reinforcement learning (RL) is a technique inspired by behavioral psychology that acquires knowledge by exploring the interaction with the local environment without the need for external supervision [4]. Besides, RL can continuously explore the surrounding environment through agents without prior knowledge, familiar with the entire environment after several training cycles, and make the right choice finally. Thus, it is suitable to address the routing challenge in the network. In this paper, we propose a method to provide the shortest paths for selecting switches based on Reinforcement learning to handle user requests on the location-aware dynamic network provisioning platform.

2. Related Work

In previous studies [5], [6] proposed an SDN-based framework that handles user requests, including locations, QoS levels, and implements a proper network service dynamically. Through a map-based web interface, the requirements such as locations and QoS asked by users. After obtaining the user request, our framework will map out the corresponding network switches in the resource, then generate network paths between these network switches and guarantee the given QoS level. After that, the network controller implements the flow rules to the switches above for deploying a prepared network service into SDN (Software-defined Networking) enabled network infrastructure. However, how to calculate and provide a reasonable route between selected switches becomes a notable issue. So, researchers need to answer the question of how network providers can make decisions on whether the requests are acceptable to their limited resources. Therefore, this paper prosed the method to finding paths based on reinforcement learning.

Besides, Reinforcement learning(RL) is known to be a powerful tool for solving complex problems. Q-learning is an intelligent algorithm that is applied to SDN routing to optimize SDN routing intelligently. Additionally, the field of Reinforcement learning has grown dramatically over several years. The RL has used in many application majors such as game playing, networks, and telecommunications, for building autonomous systems that improve themselves with experience. As a result, RL is considered an efficient solution to solve the routing network with complex conditions. The algorithm of RL to address routing problems started in 1994 with the seminal work of Boyan and Littman [7]. Q-Routing is a flexible routing algorithm based on the Q-learning principle to improve packet routing in communication networks that was proposed by Boyan and Littman [7]. Q-Learning is a model-free algorithm. It was relying on Q-value iteration to achieve balance with exploration and utilization, which is one of the simplest methods to take in intensive learning [8]. Q-learning is applied to the routing issues in the Q-routing, with the routing table in the distance-vector algorithm [8] recovered by the table of estimations (Q values) based on the link delay. In the Q-routing, the routing policies are used as in the distance vector routing algorithm. Q-function used to update routing table entries in Q-routing [8]. However, the Q-routing has mentioned that neural networks can be used for approximating the Q-function by combining differing parameters of the system, such as queue lengths, time delays[7]. The algorithm described in [8], [9]:

$$Q'(d, n_1) = \min_{Q(d, n_2)} + Transmission\ Time(R\ \ to\ \ n_1) + Waiting\ Time(R)$$

$$diff = Q'(d, n_1) - Q(d, n_1)$$

$$Q(d, n_1) \mathrel{+}= \alpha^* diff$$

Q(d, n): current estimate for delay from router n to device d. New assessments integrated into our old view for Q(d, n) through exponential averaging. When selecting a route to the destination device, the router observes the Q values of neighboring routers and selects the neighbor with the minimum estimated delay. These estimates initialized to all be 0. Still, once the packet reaches the desired destination, we will have a more accurate estimate for the delay to that device, and this information will propagate backward. As more packages can complete different paths in the initial exploration phase, we can use the improved estimates to find the minimum delay paths between devices.

3. Comparision of the Q-routing and the shortest path algorithm (Dijkstra):

This paper is going to consider and compare two algorithms that are the Dijkstra algorithm and Q-routing algorithm (the routing based on Q-learning algorithm) to calculate and find the shortest path. The first algorithm, Dijkstra, is also known as one of the most popular deployed algorithms for finding the shortest route in today's internet routing. In our previous research, the Dijkstra algorithm is used to calculate the paths based on cost metrics, which are the network's parameters such as delay, bandwidth, and costs. This approach, the packets have arrived at a node, take the same shortest path to the destination on the network. The main disadvantage of this algorithm is the changes in the network's traffic dynamically are not taken into consideration. For example, if there is congestion in the intermediate node's queue, packets will be delayed before reaching the destination. In these cases, choosing alternate paths that may not be the shortest path, but delivery time may be the fastest, is the optimal solution to solve congestion. This routing algorithm is being used without intelligence; it needs human supervision and interpretation to adapt to failures and changes dynamic. For the Dijkstra algorithm, it is quick to give the shortest path, but it will provide only one not all of the best routes. If the forwarding of all packets only used the Dijkstra algorithm, It can cause a severe problem that all data flow combined by selecting the same the shortest path, which leads to network the bottleneck and dramatically reduces the use of network links. Therefore, we need a better solution to manage the routing of the SDN network. With today's development and advancement of machine learning techniques, the routing algorithm based on reinforcement learning (Q-learning) is considered a new approach and have great potential to optimize routing in Software-defined Networking. In Q-learning, we have action, rewards, states, and discount rates. Q-learning can be used to find an optimal policy for selecting steps for any Markov decision process. The difference between Q-routing and Q-Learning is that Q-routing does not have a discount rate, and for each state. Besides, Q-routing will choose the minimum future cost instead of maximum future reward. Q-routing is an adaptive algorithm capable of routing without prior knowledge of the operating environment through process exploration and exploitation; it sends the packages based on neighbors' routing information. In the process of self-learning and exploring the environment, to build routing tables based on distribution time (value Q), the Q-routing will send packets to every node in the network. These delivery times will be continuously updated every time a node sends a packet to a specific destination and adjustment depending on traffic at a given time to the destination. Therefore, a node may decide to select alternate routes when it found that the queues appear congested or predicted of congestion in the intermediate nodes based on this information. That leads to avoid the bottleneck and faster delivery, unlike the Dijkstra algorithm.

4. Conclusion

This paper presented the advantages and disadvantages of approaches and algorithms for routing packets on dynamic networks with the shortest path algorithm and the reinforcing learning perspective. Q-routing offers a solution to improve routing performance in congested networks, as opposed to merely using the shortest path algorithms.

In our future work, to evaluate the proposed method, we conducted mininet-based experiments with various network settings. The test will be taken in a network with 30, 50, 100, 150, 200 nodes, respectively. Each other link has a bandwidth capacity in the range of 100 - 1000 Mbps, the delay is between 2 - 5 ms. The test assumes that a switch can hold up to 1000 flow entries. The QoS parameter of a request includes bandwidth constraint (1~10Mbps) and delay constraint (40ms ~ 200ms). With this setting, we can measure the number of accepted requests and the accumulated bandwidth for our proposed new method and other earlier methods using the Dijkstra algorithm. Besides, trying to improve the Q-routing algorithm is very important. Deep reinforcement learning is known as the optimal solution to enhance the performance of the Q-routing algorithm. Additionally, we can see that routing based on machine learning techniques has a vast potential to achieve routing optimization with various operation and maintenance strategies.

Acknowledgments

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). This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2016-0-00314) supervised by the IITP Institute for Information & communications Technology Planning & Evaluation).

References

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

[2] R. Boutaba et al., “A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities,” J. Internet Services Appl., vol.9, p.16, Dec. 2018.

[3] Z. M. Fadlullah et al., “State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems,” IEEE Commun. Surveys Tuts., vol.19, no.4, pp.2432 - 2455, 4th Quart., 2017.

[4] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction.
2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[5] 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.

[6] 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.

[7] J. A. Boyan and M. L. Littman, “Packet routing in dynamically changing networks: A reinforcement learning approach,” Advances in neural information

processing systems, 1994, pp.671-678
[8] R. S. Sutton and A. G. Barto, “Reinforcement learning: an introduction”, MIT Press, Chp.1-3, 1994.
[9] Littman, Michael, and Justin Boyan. "A distributed reinforcement learning scheme for network routing." Proceedings of the international workshop on applications of neural networks to telecommunications. Psychology Press, 1993.