Service-centric Segment Routing Mechanism using Reinforcement Learning for Encrypted Traffic

Van TONG*, Sami SOUIHI*, Hai Anh TRAN† and Abdelhamid MELLOUK*
*LISSI-TincNET Research Team, UPEC, France
Email: van.tong@univ-paris-est.fr, sami.souihi@u-pec.fr, mellouk@u-pec.fr
† SOICT, HUST, Vietnam
Email: anhth@soict.hust.edu.vn

Abstract—**For the past decade, IP (Internet Protocol) routing approaches utilize TCAM (Ternary Content Addressable Memory) for the rule matching in the switches. These approaches are expensive and require more power consumption. Fortunately, the emerging of segment routing can resolve this drawback by encoding a routing path into the packet header to forward the packets to a destination. However, the standard segment routing algorithm has encountered a main problem. Using the shortest path to forward the packets can lead to a high traffic load on these paths and a performance reduction. It results in a decrease in user's perception and some negative economic impacts for ISPs (Internet Service Providers). Therefore, in this paper, we propose a novel service-centric segment routing mechanism using reinforcement learning in the context of encrypted traffic. Our proposal aims to help ISPs to decrease the influence of the network problems and meet the strict user's requirement related to QoE (Quality of Experience). The obtained results under the considered conditions demonstrate that our approach outperforms the standard segment routing algorithm and requires reasonable computational cost.**

Index Terms—**Network troubleshooting, Segment routing, Traffic classification, SDN**

I. Introduction

In 2020, Covid-19 pandemic is happening all over the world, which causes many negative impacts on various fields (e.g. economy, education, tourism, etc.). People are advised to stay at home to protect themselves during the confinement. For that reason, many online activities are occurring via the Internet such as education, remote working, entertainment and so on. As a result, some ISPs and network operators [**?**] have to suffer the high pressure from the significant amount of Internet traffic due to a huge number of network devices. In the past, ISPs can increase the capacity of their network infrastructure by implementing new network devices. However, this solution is no longer available during the confinement of Covid-19 pandemic. The amount of Internet traffic is increasing exponentially while the capacity of network infrastructure is linearly growing. This results in the congestion problem in the network. Therefore, traffic engineering as well as routing approach are the potential solutions to optimize the network and reduce the influence of the network problems.

In much existing research work [**?**], [**?**], routing approaches are implemented to reduce affections of problems in the network. These approaches use TCAM for rule matching in the switches, so it can retrieve the content in a single clock cycle. However, these approaches are expensive and require more power consumption. Fortunately, segment routing is a potential promise to solve these disadvantages. Segment routing is a kind of label switching approach, which encodes the routing paths into the packet's headers. The core idea of SR architecture is based on the notion of source routing and tunneling to guaranty the scalability property in decreasing the amount of state information to be processed in the core network. Besides, the main benefit of SR is to fix scalability issues and limitations of Multi-Protocol Label Switching (MPLS) approach. Concretely, MPLS requests IP network to maintain explicit state at all nodes along an MPLS path and that may bring a scalability problem in both control plane and data plane. Also, MPLS cannot benefit from the load balancing given by Equal Cost MultiPath (ECMP) routing. On the contrary, SR does not require any state maintenance in all network nodes and it takes efficiently advantage of ECMP routing.

However, the standard segment routing algorithm contains some drawbacks related to high traffic load and performance reduction because it considers the shortest paths to forward the packets. Moreover, implementing the common routing mechanism for various kinds of services is not effective. Nowadays, the network traffic is encrypted to protect the user's privacy and data during the transmission, so the information about the class of service will be hidden. Therefore, in this paper, **we propose a novel service-centric SR (Segment Routing) mechanism using RL (Reinforcement Learning) in the context of encrypted traffic**. Our proposal is to help ISPs to reduce affections of network problems by selecting appropriate paths corresponding to different kinds of services and meet the strict user's requirements related to QoE. We consider QUIC (Quick UDP Internet Connection) protocol [**?**] which is a transport layer network protocol designed by Google from 2012. We mention here only a use-case (QUIC protocol), but it can be generalized to other encryption protocols.

The remainder of the paper is structured as follows. Section II introduces some related work in segment routing. In Section III, the paper presents the proposed service-centric segment routing mechanism. Section IV describes the experimental results of the proposed mechanism. Finally, the paper concludes with Section V which highlights some future works.

II. Related work

A. Related work

Davoli et al. [?] proposed architecture for SR-based TE (Traffic Engineering). The ideal is to improve the TE using SR which allows to select the paths within a network domain via operating at the border of the network.

Barakabitze et al. [?] presented QoEMuSoRo, a QoE-based multipath source routing algorithm, to optimize the user's perception by forwarding MPTCP (Multi-path TCP) flows using SR over SDN. Concretely, QoEMuSoRo selects the shortest path which meets the strict requirements related to packet loss and delay.

Lee et al. [?] proposed a segment routing algorithm that can meet the bandwidth requirements. Besides, the proposed routing algorithm also considers the balance of traffic load using the link criticality and the reduction of the extra cost of the packet header size.

Barakabitze et al. [?] proposed a QoE-aware SDN-based MPTCP/SR approach to improve the QoE for multimedia services over the 5G networks. This approach controls the network flows using SDN controller and performs the source routing using SR.

Huang et al. [?] studied segment routing with the maximum segment list depth (SLD) constraint to propose an approach for resolving the problems related to the large label stack and the long packet header. Besides, a novel path encoding schema is presented to minimize the SLD under the specific constraint when they also consider multiple kinds of overhead. The proposed approach proved that it can resolve the problems of segment routing when there are the constraints on the maximum SLD.

The problem of the existing research work on segment routing is that they consider the same solutions for a variety of services, which is not effective with the rapid growth of many multimedia services. In this paper, **the proposed service-centric segment routing algorithm is able to implement a specific routing strategy for a kind of service and meet the strict requirements of end-users in term of QoE**.

III. Proposed Service-centric SR Mechanism for Encrypted Traffic

A. Overview of Service-centric SR Framework

Our proposal is a multi-modular system which is depicted in Fig. 1. The major components in this framework are as follows:

1) *Network monitoring* contains two main modules including parameter measurement and traffic classification modules. The parameter measurement module is to monitor and collect some network parameters while the traffic classification module is to classify the encrypted traffic into different kinds of services.
2) *Service-centric Detection* considers the class of service and some network parameters from network monitoring to identify the abnormal symptoms of problems in the network.
3) *Service-centric Remediation* aims to reduce affection of problems by implementing different routing strategies corresponding to various services which meets the strict user's requirements in term of QoE.

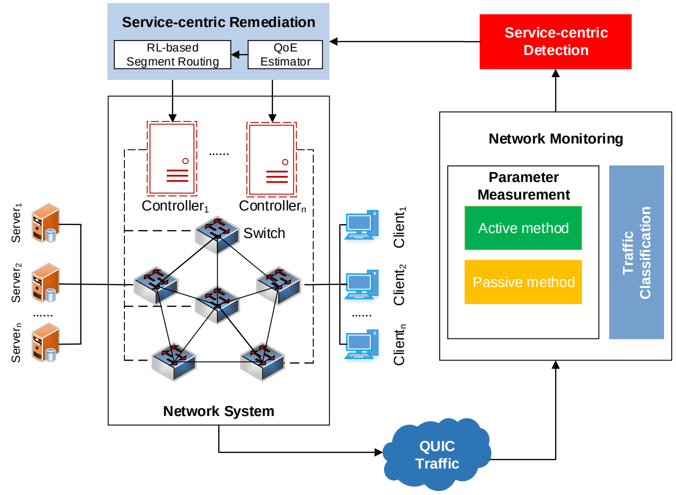

Fig. 1: The service-centric SR framework for encrypted traffic.

In this paper, we present the overview of service-centric SR framework for encrypted traffic and concentrate on a novel service-centric segment routing mechanism using reinforcement learning in the service-centric remediation module. In this framework, we utilize SDN (Software-Defined Networking) to facilitate in the experiment. The reason for it is that SDN offers some native monitoring approaches (e.g. link layer discovery protocol, etc.), and the opportunity to change and update the network configuration in the switches remotely via the SDN controller.

B. Network Monitoring

1) Traffic Classification: Nowadays, network traffic is encrypted to protect data and user's privacy during transmission. Therefore, we present a novel traffic classification approach to identify different kinds of services including video streaming, file transfer and VoIP. After investigating the characteristic of some services, we present a classification approach using deep learning and hybrid features (handcrafted features and implicit features). First, network traffic is classified into mice flows (VoIP) or elephant flows (video streaming and file transfer) using the random forest algorithm and some handcrafted features (flow-based features). Then the elephant flows will be classified into video streaming or file transfer using convolutional neural network and implicit features (packet-based features). The detail of this approach is described in detail in [?].

2) Parameter Measurement: Similar to the traffic classification module, we implement a parameter measurement module to estimate several network parameters on each link in the network. Many service level agreements (SLAs) of service providers rely on some performance metrics containing packet loss and latency [?]. Consequently, we consider here several parameters including latency, packet loss and link utilization.

Concerning the latency, it refers to the time interval of data transmission through a link in the network. There are much existing work calculating the link latency using LLDP (Link-Layer Discovery Protocol) [?], so we implement this approach with some modifications to estimate link latency. Concretely, the latency of link s_1-s_2 refers to the duration when the controller sends a Packet-out message to switch s_1 until the controller receives a Packet-in message from switch s_2.

Regarding the packet loss, it refers to the loss of the LLDP packet on each link. It is measured as the ratio between the number of packets lost and packet sent.

As for the link utilization, it is the amount of traffic traversing a link divided the link capacity.

C. Service-centric Detection

When the abnormal symptoms of problems happen, it reflects that a network problem can occur in the network soon. Much research work detect the network problems by identifying the abnormal symptoms of problems in the network. Therefore, in this section, we present a novel service-centric detection approach to identify abnormal symptoms. We consider here the class of service (CoS) in the detection approach because various services have particular behaviors that lead to a particular range of network parameters. This can help to improve the accuracy of abnormal symptom detection. The detail of this approach is described in Fig. 2.

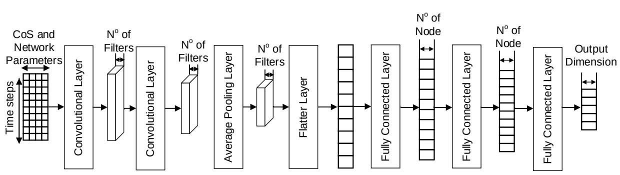

Fig. 2: The service-centric symptom detection approach.

The service-centric detection approach takes into account the class of services, and some time-series network parameters on network flows such as latency, packet loss and link utilization as the input. Then this information will be analyzed using convolutional neural network (CNN) to detect the abnormal symptoms of problems (e.g. increase of latency, packet loss, etc.). Some of the time-series network parameters are not effective for symptom detection, so 2D (2 dimensional) convolutional neural network is implemented to learn effective representations to enhance the accuracy of symptom detection. Next, this representation will be processed in 2D average pooling layer and flatter layer before fed into the fully connected layer to classify the network states into normal or abnormal symptoms.

D. Service-centric Remediation

Although there are many remediation approaches (e.g. routing approaches, load balancing, etc.), some existing researches [?], [?] implement the routing aspect to reduce the influence of network problems in the network. Therefore, in the service-centric remediation module, a service-centric segment routing mechanism using reinforcement learning is proposed to reduce the affection of problems by resolving it temporarily. The term *'service-centric'* means that we consider different kinds of strategies corresponding to a variety of services to improve the performance of the remediation module.

1) QoE Estimator: Although there is much research work on segment routing, these approaches take into account QoS (Quality of Service) as the input. Nowadays, perceived end-to-end quality becomes one of the main user's requirements that need to be guaranteed by ISPs. Consequently, in the service-centric remediation approach, we consider QoE as the input of the segment routing algorithm.

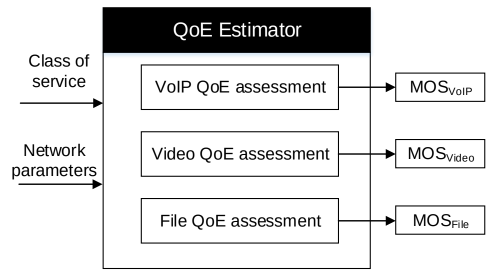

Fig. 3: The QoE estimator.

The rapid growth of Internet leads to a various kinds of multimedia services (e.g. video streaming, chat, etc.). Implementing the same QoE model for different kinds of services is not effective. Consequently, we implement a novel QoE estimator to calculate the QoE [1] of different services in the context of encrypted traffic using the class of service from the previous module (traffic classification). This module is inherited from our previous work [?]. According to the traffic classification module, the QoE estimator can apply the appropriate QoE models corresponding to the specific services (video streaming, file transfer and VoIP) to enhance the accuracy of QoE estimation. The detail of the QoE estimator is described as in Fig. 3.

2) RL-based Segment Routing: The segment routing algorithm is considered here as a reinforcement learning task that takes into account QoE as environment feedback. The reinforcement learning model contains *agent, state, action, reward, and policy*. The detail is described as follows:

Agent: An entity in the network system implement a learning algorithm to execute its tasks. In the routing algorithm, an agent aims to identify the appropriate routing paths to maximize the reward.

State: A snapshot of the network environment returned by an agent.

Action: An action describes how an agent responds to the network environment. In the routing algorithm, an action is

[1]Mean opinion score (MOS) at the end-user' side.

a routing path between a source and a destination in the network.

Policy: A policy is a map between a state and an action in the network environment.

Reward: A reward is feedback which the network environment returns to the agent. In the routing algorithm, the agent observes the network state s and implements an action a from the routing policy. Next, the agent moves to next state s' and receives a reward r. We consider here QoE of the chosen path as the reward which network environment returns to the agent. The QoE is calculated via QoE estimator (section III-D1).

The objective of reinforcement learning task is to optimize the objective function O_f in order to maximize the expected cumulative reward (Eq. 1):

$$O_f = Max\ E[\sum_{t=0}^{\infty} \gamma_t \times r_t]. \tag{1}$$

where $\gamma_t \in$ [0, 1] is the discount factor.

We present the routing information in a Q-table. Each value in the Q-table represents a Q-value of an action (a routing path) in the network environment. According to the centralized architecture of SDN, all routing paths in the network can be extracted in the SDN controller. Our approach is based on reinforcement learning to learn the actions corresponding to the network states and, it uses this action to make control decisions. The agent executes an action and receives the immediate reward r. Using this immediate reward and the long-term reward, it can update the Q-table that influences on the future routing policy. When an action is chosen, the Q-value of this action in the Q-table is updated as in Eq. 2:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')]. \tag{2}$$

where α is the learning rate and γ is the discount factor.

3) Exploration and Exploitation Tradeoff: To obtain the optimal cumulative reward, reinforcement learning need to balance between exploration and exploitation phase. We can not implement systematically exploitation phase which selects the action with maximal Q-value because each routing path need be evaluated many times to obtain the reliable expected reward. Therefore, we consider the tradeoff between exploration and exploitation phase. This tradeoff is formalized as a MAB problem (Multi-Armed Bandit). In this paper, we present some popular algorithms to resolve the MAB problem including *ϵ-greedy, softmax* and *UCB1* (Upper Confidence Bounds) [**?**].

First, *ϵ-greedy* is the simplest policy to resolve the bandit problem. Concretely, the agent selects the action with highest reward with probability of (1-ϵ). Otherwise, the agent selects the actions randomly. The objective is that the agent can learn more about network environment and becomes more confident. The ϵ value will be reduced against the time as in Eq. 3:

$$f(x) = \epsilon * (0.5 + \log_{10}(2 - \arctan(\frac{x}{10} - 2))). \tag{3}$$

where x is the episode number (forwarding time).

Second, the *softmax* algorithm selects action using a probability function of reward. Each action a_i is assigned a probability p_i as in Eq. 4:

$$p_i = \frac{e^{\frac{r_{a_i}}{\tau}}}{\sum_{j=1}^{N} e^{\frac{r_{a_j}}{\tau}}} \tag{4}$$

where τ is the temperature parameter.

The more we reduce temperature parameter τ, the more we exploit the system. Therefore, *softmax* algorithm not only explores the less-used action, but also selects the best action in terms of expectation gains. In that way, we reduce temperature parameter τ each episode (forwarding time) as in Eq. 5:

$$\tau = \tau - \phi. \tag{5}$$

where ϕ ($0 < \phi < 1$) is the weight parameter.

On the other hand, the *UCB1* algorithm is related to an index-based policy. The UCB-index is defined as the sum of the current reward and the size of the one-sided confidence interval for the reward. The UCB-index is described as in Eq. 6:

$$UCB - index_{a_i} = r_{a_i} + \sqrt{\frac{2ln(N)}{n_{a_i}}}. \tag{6}$$

where n_{a_i} is the number of times action a_i is chosen and N is the episode number (forwarding time).

After calculating UCB-index for each action, *UCB1* algorithm selects the action with maximal UCB-index.

IV. Experimental results

A. Experiment Setup

To evaluate the performance of the novel service-centric segment routing mechanism, we set up a testbed with *mininet* v2.2 and *ONOS* controller v2.4. The uniform link capacity is set to 10 Mbps, and we generate a huge amount of traffic between clients and servers to emulate the network problems (congestion). The network topology is depicted as in Fig. 4. In the exploration-exploitation tradeoff, ϵ, τ and ϕ are set to 1, 3 and 0.01, respectively. In the reinforcement learning algorithm, α and γ are set to 0.7 and 1, respectively. These parameters are chosen via the experiments. The source code of the service-centric segment routing framework is available at [**?**].

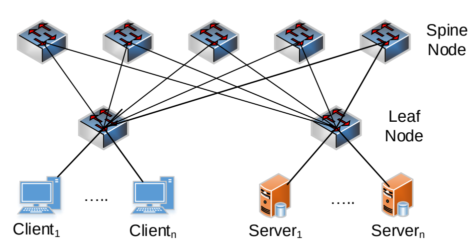

Fig. 4: Network topology.

B. Benchmark

To validate the performance of the proposed segment routing mechanism, we compare our proposal with some benchmark including:

1) *Standard Segment Routing (Standard_SR)*: This algorithm selects the shortest paths between clients and services.
2) *Segment Routing with maximal QoE (Max_QoE)*: This approach identifies all available routing paths between clients and services using SDN controllers. After that, it calculates the QoE of these paths and selects the path with maximal QoE (maximal reward). In the proposed service-centric segment routing mechanism using reinforcement learning (*RL_SR*), we only calculate the QoE of the chosen path which is selected via some selection algorithms such as *ϵ-greedy, softmax* and *UCB1*. This can help to reduce the computational cost of the proposed mechanism.

These approaches are evaluated via some performance metrics as follows:

1) *Cumulative Reward*: This factor is to evaluate the performance of the algorithm over times, and it is calculated every 100 episodes (forwarding time).
2) *CPU Usage*: CPU time is the amount of time (in clock ticks or seconds) for which a central processing unit (CPU) is used for executing the program's instructions. Besides, CPU time is measured as a percentage of the CPU's capacity which is called CPU usage. This parameter is measured via *psutil* library in Python [**?**].

C. Performance Analysis

In this section, we evaluate the performance of several segment routing algorithms including standard segment routing (Standard_SR), segment routing algorithm with maximal QoE (Max_QoE) and service-centric segment routing mechanism using reinforcement learning (RL_SR). We consider here video streaming as a use-case to evaluate the performance of these approaches, and the figure for other services (file transfer and VoIP) will be considered in our future work.

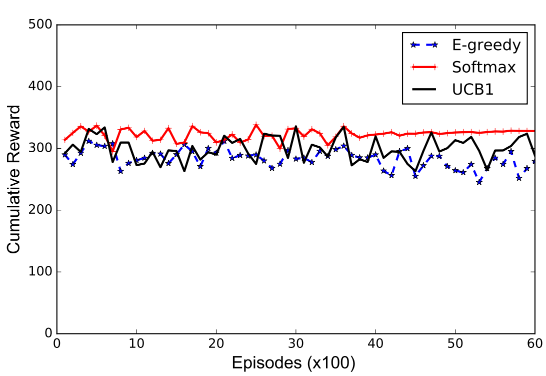

Fig. 5: Cumulative reward of three selection algorithms in proposed segment routing mechanism.

There are some selection algorithms (e.g. UCB1, softmax, etc.) for the MAB problem in reinforcement learning algorithm, so we evaluate the performance of these algorithms to choose the appropriate selection algorithm. Fig. 5 illustrates the cumulative reward of *UCB1, softmax*, and *ϵ-greedy* in proposed segment routing mechanism. The cumulative reward of *softmax* algorithm is better than the others. During the first 4,000 episodes, the cumulative reward of three algorithms fluctuate between 270 and 340. After that, the cumulative reward of *softmax* algorithm converges around 325.6 while the other algorithms continue to fluctuate. *ϵ-greedy* algorithm selects the action with maximal reward in the following stage, so some less-used action will not be chosen frequently. As a result, the figure for *ϵ-greedy* is lowest in three algorithms. *softmax* algorithm selects the action based on the probability function of reward, so the less-used action will be chosen more frequently. Therefore, the cumulative reward of *softmax* algorithm is better than the others, and we use *softmax* algorithm as the selection algorithm in reinforcement learning for the following experiments.

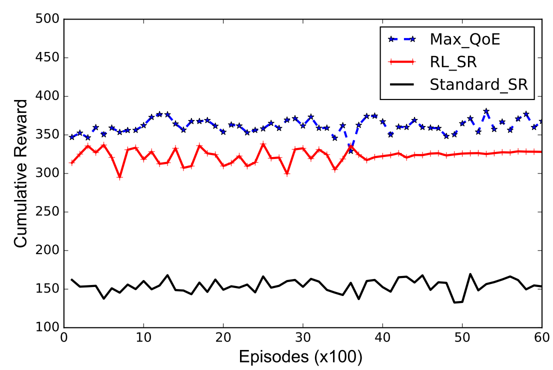

Fig. 6: Cumulative reward of three kinds of segment routing mechanism.

We now analyze the performance of different segment routing algorithms including *Max_QoE, RL_SR*, and *Standard_SR* (Fig. 6). *Standard_SR* algorithm selects the shortest path to forward the packets between clients and servers, so the cumulative reward is lower than the others. *Max_QoE* algorithm systematically selects the paths with maximal reward, so the figure of this algorithm is slightly higher than the figure for our proposal (*RL_SR*). However, *Max_QoE* algorithm incurs a high computational cost, so it is not effective with the large-scale network. The CPU usage of three segment routing algorithms are described in Fig. 7.

The CPU usage of *Max_QoE* algorithm is the highest in three segment routing mechanisms, and it requires up to approximately 74 percent of CPU's capacity. *RL_SR* algorithm requires about 62 percent of CPU's capacity, and the required CPU's capacity usage reduces to 55 percent for *Standard_SR* algorithm. *Max_QoE* algorithm need to measure the network parameters of all links in the network, and then calculate the QoE of all available routing paths. As a result, it requires more

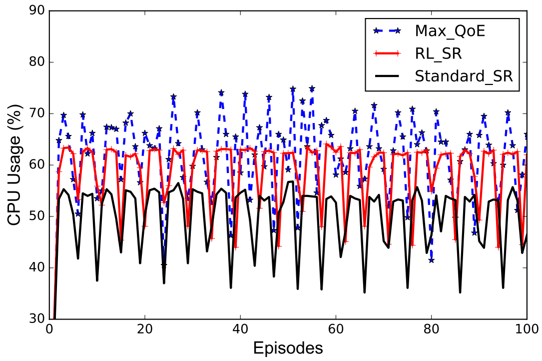

Fig. 7: CPU usage of three kinds of segment routing mechanism.

computational cost. *RL_SR* algorithm only need to measure the network parameters in the links of a chosen routing path and calculate the QoE for this path, so it requires less computational cost in comparison with *Max_QoE* algorithm. A noticeable feature from Fig. 7 is that the CPU usage falls to the bottom periodically. The reason for it is that we set a small sleep duration after a given time of packet generation between clients and servers.

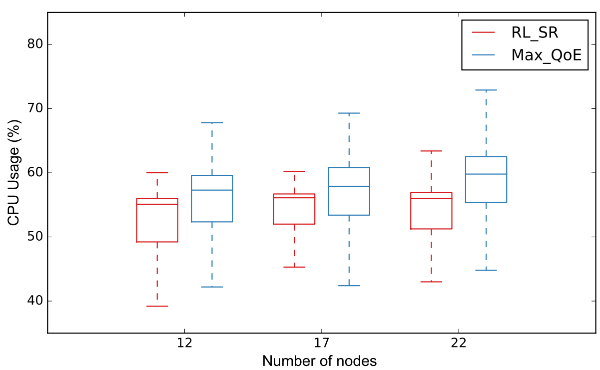

Fig. 8: CPU usage of three kinds of segment routing mechanism against the number of nodes in the network topology.

To evaluate the computational cost of *RL_SR* and *Max_QoE* algorithms thoroughly, we investigate the CPU usage of these algorithms in the increase of a number of nodes in the network topology (Fig. 8). When we increase the number of nodes in the network topology, the CPU usage of *RL_SR* algorithm is more stable than the figure for *Max_QoE* algorithm. The variation of CPU usage in *Max_QoE* algorithm will increase in the growth of a number of nodes in the network topology. Therefore, *Max_QoE* algorithm is not effective in the large-scale network (e.g. a network with few thousands of nodes, etc.).

V. CONCLUSION

In this paper, we proposed a novel service-centric segment routing mechanism for encrypted traffic. It is also considered as a remediation approach in network troubleshooting architecture. The proposal is able to reduce the affection of network problems by selecting the appropriate routing paths that meet strict user's requirements related to QoE. The experimental results show that the proposed service-centric segment routing mechanism using reinforcement learning obtains better cumulative reward in comparison with the standard segment routing algorithm.

In the future, we will consider the proposed routing mechanism in the context of other services including file transfer and VoIP. Besides,concerning the scalability, we plan to investigate the performance of the proposed routing mechanism in the large-scale network to evaluate it completely.

REFERENCES