Intelligent Routing with deep reinforcement learning

Noe M. Yungaicela-Naula[a,*]

[a] Tecnologico de Monterrey, School of Engineering and Sciences, Monterrey 64849, Mexico

ARTICLE INFO

Keywords:
Reinforcement learning

ABSTRACT

This document presents a design of an intelligent routing mechanism by using deep reinforcement learning.

1. Definition

The routing mechanism is based on the ideas presented in [1, 2, 3]. The network graph is defined as follows.

- $\mathbf{G} = (\mathbf{V}, \mathbf{E})$.
- $\mathbf{V}$: nodes.
- $\mathbf{E}$: links.
- N: number of nodes.
- $u^{i,j}$: utilization rate of $e^{i,j}$.
- $d^{i,j}$: delay of $e^{i,j}$.
- $p_{src,dst}$: path between src and dst.
- $u(p_{src,dst}) = \min_{i,j \in p_{src,dst}} u^{i,j}$: path utilization rate.
- $d(p_{src,dst}) = \frac{d^{i,j}}{|e^{i,j}|_{i,j \in p_{src,dst}}}$: path delay.

1.1. Network Performance

$$n_p = \frac{\rho}{|E|} \sum_{i,j \in V} u^{i,j} + \frac{(1-\rho)}{|E|} \sum_{i,j \in V} d^{i,j} \tag{1}$$

where $\rho < 1$.

1.2. Markov Decision Process

1.2.1. State

We define the state by $s_t \in \mathbb{R}^{3N+E}$.

$$s = (f_1^\top, f_2^\top, f_3^\top, f_4^\top)^\top \tag{2}$$

where f_1, f_2, f_3, and f_4 are indicator vectors described as follows. $f_1 \in \mathbb{R}^N$ denotes the source node. $f_2 \in \mathbb{R}^N$ represents the destination node. $f_3 \in \mathbb{R}^{\mathbb{N}}$ represents the current node. Finally, $f_4 \in \mathbb{R}^E$ indicates the current path that connects the source the current node.

For the network in Figure 1, the indicator vectors are:

$$\begin{aligned}
f_1 &= (0_1, 1_2, 0_3, 0_4, 0_5, 0_6)^\top && \texttt{// Src node} \\
f_2 &= (0_1, 0_2, 0_3, 1_4, 0_5, 0_6)^\top && \texttt{// Dst node} \\
f_4 &= (0_1, 0_2, 0_3, 0_4, 0_5, 1_6)^\top && \texttt{// Current node} \\
f_3 &= (0_1, 0_2, 0_3, 1_4, 0_5, 0_6, 1_7, 0_8, 0_9)^\top && \texttt{// Current path}
\end{aligned}$$

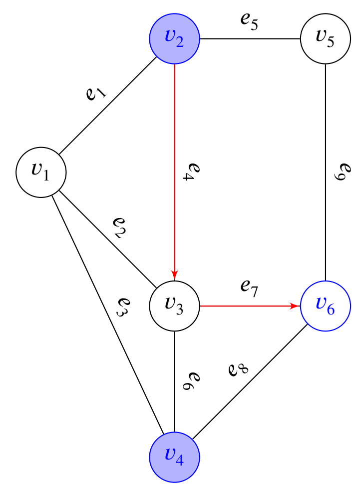

Figure 1: State of routing in an example of network. v_2 is the source, v_4 is the destination, and v_6 is the current node.

1.2.2. Actions

$$A = \{a_1, a_2, a_3, \ldots, a_i, \ldots, a_{|V|}\}. \tag{3}$$

where the action a_i indicates a hop to a neighbor node i.

1.3. Reward

$$r = \begin{cases} -0.02 & \text{if } a_i \text{ is not possible} \quad \texttt{// i isn't a neighbor} \\ 0 & \text{if } a_i \text{ is possible AND } n_{step} < max_{steps} \\ -0.001 & \text{if } n_{step} = max_{steps} \text{ AND } n_i \neq n_{dst} \quad \texttt{// failed} \\ r_0 & \text{if } n_i = n_{dst} \end{cases} \tag{4}$$

where

$$\begin{aligned} r_0 = 0.1 + \sigma \cdot (\beta \cdot u(p_{src,dst}) \\ + (1-\beta) \cdot d(p_{src,dst})) + (1-\sigma) \cdot n_p. \end{aligned} \tag{5}$$

2. Implementation

Matlab.

*Corresponding author
ORCID(s):

References

[1] G. Kim, Y. Kim, and H. Lim, “Deep reinforcement learning-based routing on software-defined networks,” *IEEE Access*, vol. 10, pp. 18 121–18 133, 2022.

[2] A. Rezapour and W.-G. Tzeng, “Rl-shield: Mitigating target link-flooding attacks using sdn and deep reinforcement learning routing algorithm,” *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021.

[3] D. M. Casas-Velasco, O. M. C. Rendon, and N. L. S. da Fonseca, “Drsir: A deep reinforcement learning approach for routing in software-defined networking,” *IEEE Transactions on Network and Service Management*, pp. 1–1, 2021.