International Journal of Advanced Smart Convergence Vol.12 No.3 13-18 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.3.13

IJASC 23-9-2

On the Data Features for Neighbor Path Selection in Computer Network with Regional Failure

Yong-Jin Lee

Professor, Dept. of Technology Education, Korea National University of Education, Korea
lyj@knue.ac.kr

Abstract

This paper aims to investigate data features for neighbor path selection (NPS) in computer network with regional failures. It is necessary to find an available alternate communication path in advance when regional failures due to earthquakes or forest fires occur simultaneously. We describe previous general heuristics and simulation heuristic to solve the NPS problem in the regional fault network. The data features of general heuristics using proximity and sharing factor and the data features of simulation heuristic using machine learning are explained through examples. Simulation heuristic may be better than general heuristics in terms of communication success. However, additional data features are necessary in order to apply the simulation heuristic to the real environment. We propose novel data features for NPS in computer network with regional failures and Keras modeling for computing the communication success probability of candidate neighbor path.

Keywords: *Neighbor Path Selection, Data Features, Heuristic Algorithm, Simulation*

1. Introduction

Regional failures in computer network caused by earthquake or forest fire have geographically correlated characteristics. Correct neighbor path selection (NPS) for communication becomes an important issue because simultaneous regional failures may make it impossible to transfer important data between the source node and the destination node [1]. To prevent the communication loss, for example, a resilient overlay network (RON) uses an alternative path to bypass network failure [2]. Feamster and other researchers shows that if the primary path fails, RON can find an alternative communication path [3].

Several general heuristics for NPS have been proposed. Proximity-aware heuristic finds the neighbor path by using Euclidean distance between all physical nodes [4, 5]. Sharing-aware heuristic considers sharing node. Proximity-sharing-aware heuristic uses both proximity and sharing to find the best neighbor path [5, 6].

Manuscript Received: August, 10, 2023 / Revised: September, 13, 2023 / Accepted: September, 17, 2023
Corresponding Author: lyj@knue.ac.kr
Tel: +82-43-230-3774, Fax: +82-43-230-3610
Professor, Dept. of Technology Education, Korea National University of Education, Korea

However, since above heuristics cannot utilize past failure information, they can determine only local optimal neighbor paths. In other words, to bypass a regional failure, there must be a lack of algorithms for successful neighbor path prediction. These defects can be addressed preventing algorithmic deficits and providing performance guarantees through machine learning [7].

Simulation (Communication-aware) heuristic causes regional failures after obtaining the shortest paths as many as an arbitrary number (K) in a given network [5]. Then, after computing the communication success probability for each path, the path with the maximum value among them is selected as the neighboring path. Communication success probability is obtained using machine learning.

Simulation heuristic showed better communication success probabilities than general heuristics [5]. However, further considerations for NPS are needed to achieve better results required for the real regional fault network. Therefore, in this study, the dataset of simulation heuristic is revised and supplemented to reflect the real environment.

2. Data Features of Previous NPS Heuristics in Regional Fault Network

2.1. Related Heuristics for NPS

In order to obtain the neighbor path, we can use three general heuristic rules- proximity, sharing factor and proximity-sharing factor. Figure 1 depicts example network.

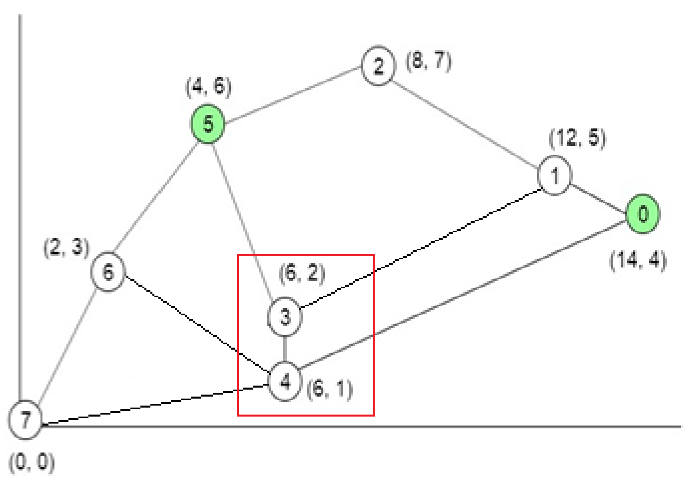

Figure 1. Example network for neighbor path selection

The number of nodes is 8 from node 0 to node 7. The coordinate is given in the form of (x, y). Thus, the coordinate of node 0 is represented as (14, 4). Source node is 0 and destination node is 5. The primary (shortest) path is 0-1-2-5 and total distance is 13.4. If we compute necessary information to apply proximity-aware heuristic, sharing-aware heuristic, and proximity-sharing aware heuristic to the example network in Figure 1, we obtain Table 1. Distance threshold is set to 3 and the number of candidate neighbor paths (K) is set to 4, respectively.

Proximity-aware heuristic utilizes the proximity which indicates the closeness between shortest (primary) path (S) and candidate neighbor path (N). In Figure 1, proximity-aware heuristic selects N = (0, 4, 3, 5) as the neighbor path since the proximity of all three paths are the same minimum (3), but the distance is minimal (14.0).

Sharing-aware heuristic uses the sharing factor which shows that two paths share a common router on each path. It finds the path with the lowest sharing factor between candidate neighbor paths.

In Figure 1, since there are three paths with the least sharing factor (0), sharing-aware heuristic finds the path with the minimum distance (14.0), thus it selects N = (0, 4, 3, 5) as the neighbor path. When there are several nodes having the same least proximity and sharing factor, distance is used as a tie breaker.

Table 1. Information applying the related heuristics to example network of Figure 1

Path	Euclidean distances between S and N	Proximity	Sharing factor	Distance of N
S=(0,1,2,5) N=(0,1,3,5)	(0, 2.2, 8.2, 10.2) (2.2, 0, 6.7, 8.1) (6.7, 4.5, 5.4, 4.1) (10.2, 8.1, 4.5, 0)	5	1	13.4
S=(0,1,2,5) N=(0,4,3,5)	(0, 8.5, 8.2, 10.2) (2.2, 7.2, 6.7, 8.1) (6.7, 6.3, 5.4, 4.1) (10.2, 5.4, 4.5, 0)	3	0	14.0
S=(0,1,2,5) N=(0,4,6,5)	(0, 8.5, 12.0, 10.2) (2.2, 7.2, 10.2, 8.1) (6.7, 6.3, 7.2, 4.1) (10.2, 5.4, 3.6, 0)	3	0	16.6
S=(0,1,2,5) N=(0,4,7,6,5)	(0, 8.5, 14.6, 12.0, 10.2) (2.2, 7.2, 13.0, 10.2, 8.1) (6.7, 6.3, 10.6, 7.2, 4.1) (10.2, 5.4, 7.2, 3.6, 0)	3	0	21.7

Proximity-sharing-aware heuristic combines proximity and sharing factor. It first check proximity and then check sharing factor. Because there are one more path with the least proximity and the least sharing factor, respectively, Proximity-sharing-aware heuristic also selects N = (0, 4, 3, 5) with the least distance as the neighbor path in Figure 1.

Simulation heuristic extends the general heuristic considering only one neighbor path [5]. Instead, it enumerates K possible paths and generates random failures on each path many times and creates the datasets. By using machine learning, simulation heuristic computes communication success probability of each candidate path, N = (0, 4, 3, 5), N = (0, 4, 6, 5), and N = (0, 4, 7, 6, 5) in Figure 1. It obtains the best path with the largest success probability.

If regional failures occur on the node 3 and 4 of example network simultaneously, all neighbor paths by proximity-aware, sharing-aware, proximity-sharing-aware, and simulation heuristic cannot succeed in communication.

2.2. Data Features of Simulation Heuristic for NPS

Now, we consider the data features in the simulation heuristic for NPS. In a given network, the distance between two farthest nodes is calculated and set to diameter. For selecting the neighbor path, we have to compute the proximity and the sharing factor.

Proximity is obtained by comparing all Euclidean distance between all the nodes on the primary path and the candidates of neighbor paths with the given threshold. This threshold is affected by the diameter. If we set the threshold as diameter, then proximity becomes large between primary path and neighbor path. If we divide the diameter by given distance factor, the proximity becomes small between primary path and neighbor path. For example, we set the distance factor to 1, then the threshold is equal to the diameter of network. On the other hand, sharing factor can be easily obtained by checking common router on the primary path and the candidates of neighbor path.

The size of failure region is also affected by the diameter. We assume that first error occurs at any node. Then, the region of subsequent errors is related with the region of the first error because the failure is regional. To confine the failure region, we introduce the random factor. If we divide the diameter by larger random factor, the failure region size becomes smaller. If we set the random factor to 1, then the failure region becomes the whole network area. That is, the use of random factor enables us to deal with the regional failures.

Figure 2 shows the distance threshold and the related failure region when the distance factor is 3 and the random factor is 2. In Figure 2, source node is 0 and destination is 5. Primary path is 0-1-2-5 and neighbor path is 0-4-3-5. Assuming that two failures occur at node 1 and 2, example shows that neighbor path succeed in the communication.

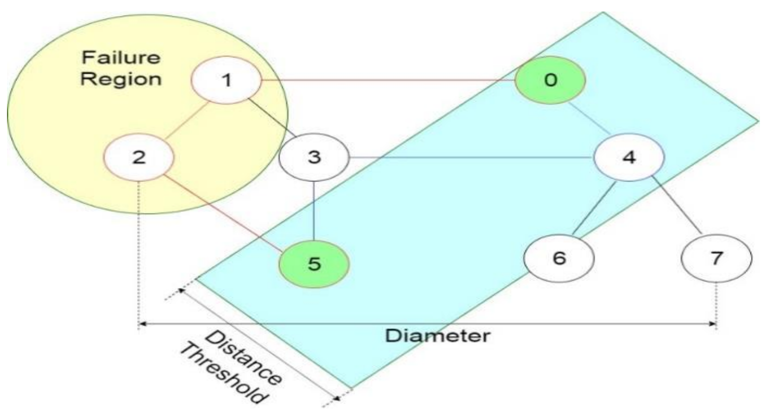

Figure 2. Distance threshold and failure region

Since simulation heuristic deals with the regional failures, it confines the failure region by using random factor shown in Figure 2. Simulation heuristic first generates the number of failures. Then first failure node is generated on the whole network. Dividing diameter by random factor, simulation heuristic can obtain the failure region including the first failure node. Subsequent failures after the first failure occur only in the failure region. Thus, simulation heuristic can simulate the regional failures. We check whether the neighbor path fails. If no failure occurs, communication is successful.

3. Novel Data Features for NPS in Regional Fault Network

The results of simulation heuristic is superior to those of general heuristics such as proximity-aware, sharing-aware, and proximity-sharing aware heuristics. However, several things need to be considered in order to apply simulation heuristic to the real environment.

We consider the dataset used by simulation heuristic [5]. (1) The simulation heuristic used Erdos-Renyi model [8]. ER(n, p) model specifies the number of vertices (n) and the probability of having an edge(p) between each two vertices. But it is realistic to use the topology of the communication network that is actually in use, G = (N, E). N and E means node and edge, respectively. (2) The simulation heuristic used the distance between source and destination, but it is more realistic to use the delay. (3) In order to simulate the regional failures, the simulation heuristic used proximity, diameter, distance factor, random factor, distance threshold and failure region size. They are generated from random number according to Uniform distribution. However, it is realistic to use probability distribution from past data. (4) Although the number of failures was randomly generated in the dataset of simulation heuristic, it is desirable to use mean real number of failures from data history. From the above description, a dataset of simulation heuristic may be modified as shown in Table 2.

Table 2. Novel data features and path information of simulation heuristic for NPS

Data feature	Path information
General	
• the real network topology, G = (N, E) • the node index of source • the node index of destination • primary path delay • neighbor path delay • the number of nodes on the primary path • the number of nodes on the neighbor path • the number of Euclidean distances under threshold • the number of common routers	• source node • destination node • the node index of primary path • delay of primary path • the node index of neighbor path • delay of neighbor path • the number of total failures • the node index of failure nodes • the fault nodes on the primary path • the fault nodes on the neighbor path
Regional failure	
• maximum distance between nodes in network • random number for setting distance threshold • distance threshold (= diameter/distance factor) • random number for setting failure region size • failure region size (= diameter/random factor) • the node index of failure • the failure probability distribution	

Simulation heuristic should enumerate K possible neighbor paths from source to destination. It also creates the dataset on the real topology. Simulation heuristic may use machine learning to compute communication success probability of each candidate path, and to find the best neighbor path with the highest communication success probability. Simulation heuristic generates regional random failures on the primary path and neighbor path by some iterations. This dataset does not include detailed path information. Table 2 shows the detailed path information obtained by simulation heuristic.

The application procedure of simulation heuristic for regional fault network is as follows: Since dataset has one binary output, we can use the machine learning by binary classification model and compute the communication success probability for each candidate path. In order to enhance the accuracy rate, we can use the deep multi-layer perceptron model [9].

For implementation, we can use Keras modeling [10,11]. As an activation function, the Relu function may be used because it is easy to propagate backward. The output is either success (1) or failure (0), so the Sigmoid function may be used. An example of leaning model using Python and Keras can be described as follows.

```
model = Sequential()
model.add(Dense(32,input_dim=16, activation='relu'))
model.add(Dense(64,activation='relu'))
model.add(Dense(64,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
```

We can use the binary_crossentropy and adam optimizer for the model compilation. In actual implementation, a more sophisticated model may be obtained by adjusting the number of neurons or layers.

4. Conclusions

In this paper, we deal with the data features required for the neighbor path selection when the simultaneous regional failures due to earthquakes and forest fires occur in computer network. To this end, we first describe previous general heuristics and simulation heuristic based on the machine learning through example. Communication success probability of simulation heuristic may be better than that of general heuristics. However, additional data features are necessary when applying simulation heuristic to the real regional fault network. We propose novel dataset to improve the simulation heuristic and deep learning modeling to compute the communication success probability for candidate neighbor path. In the future, research are expected to evaluate performance of simulation heuristic using novel dataset in the real environment.

Acknowledgement

This work was supported by the 2023 Sabbatical Leave Research Grant funded by Korea National University of Education.

References

[1] Y. –J. Lee, “A path selection model considering path latency in the communication network with geographically correlated failures,” *ICIC Express Letters*, Vol. 13, No. 9, pp. 789-796, 2019. DOI: http://doi.org/10.24507/icicel.13.09.789

[2] S. Tarkoma, *Overlay Networks*, CRC Press, 2000.

[3] N. Feamster, D. Anderson, H. Balakrishnan, and M. F. Kaasheok, “Measuring the effects of Internet path faults on reactive routing,” *SIGMETRICS Performance Review*, Vol. 31, No. 1, pp. 126-137, 2003. DOI: http://doi.org/10.1145/885651.781043

[4] K. Kim and N.Venkatasubramanian, ”Assessing the impact of geographically correlated failures on overlay-based data dissemination,” in Proc. of IEEE Global Telecommunications Conference, Miami, USA, pp. 1-5, 2010. DOI: http://doi.org/10.1109/GLOCOM.2010.5685229

[5] Y. –J. Lee, “Machine learning based neighbor path selection model in a communication network,” *International Journal of Advanced Smart Convergence*, Vol. 10, No. 1, pp. 56-61, 2021. DOI:http://doi.org/10.7236/IJASC.2021.10.1.56

[6] Y. –J. Lee, “Neighbor selection considering path latency in overlay network,” *Journal of Natural Science*, KNUE, Vol. 2, pp. 25-32, 2013.

[7] O. Simeone, “A very brief introduction to machine learning with applications to communication systems,” *IEEE Trans. on Cognitive Communication and Networking*, Vol. 4, No. 4, pp. 648-664, 2018. DOI: http://doi.org/10.1109/tccn.2018.2881442

[8] P. Erdos and A. Renyi, *On the evolution of random graphs*, Princeton University Press, 2006. DOI: https://doi.org/10.1515/9781400841356.38

[9] N. Buduma, *Fundamentals of Deep Learning*, O’reilly, 2017.

[10] A. S. Williams, *Deep Leaning with Keras*, CSPIA. 2017.

[11] J. Moolayil, *An Introduction to Deep Learning and Keras*, pp. 1-16, Apress, Berkely, 2019. DOI:https://doi.org/10.1007/978-1-4842-4240-7_1