Implementation of a fast failure recovery method considering load distribution for network slicing

Takeru Misugi*, Kouji Hirata*, and Takuji Tachibana‡
* Graduate School of Science and Engineering, Kansai University, Osaka, Japan
E-mail: {k407776, hirata}@kansai-u.ac.jp
† Graduate School of Engineering, University of Fukui, Fukui, Japan
E-mail:takuji-t@u-fukui.ac.jp

***Abstract*—In recent years, network slicing technology that can provide virtual networks according to various service requirements has been attracted much attention. In this paper, we propose a load-balanced fast failure recovery method based on Multiple Routing Configurations (MRC) in communications using network slicing. MRC ensures path availability in the case of a link/node failure by preparing multiple backup routing configurations in advance. In this paper, we implement MRC by considering the load of each node based on the amount of traffic that passes through it, which prevents unbalanced load after switching the routing configuration. We confirm that the load of nodes is equalized for each backup configuration by the proposed method. In addition, by evaluating the implementation using Mininet, we show that routing paths are ensured by adopting a backup configuration in the case of a failure.**

I. Introduction

In recent years, due to the diversification and sophistication of Internet services, various strict requirements have been required for communication networks. In the 5G mobile communication system, it is expected that the requirements for networks will be more different for each use case than ever before, and network slicing technology [3] will have an important role in realizing these requirements. Network slicing is a technology that virtually divides a communication network into multiple networks (called slices), each of which is independent of the other. It ensures the communication quality for each slice and provides communication functions that meet various requirements.

In communications using the network slicing technology, in order to provide high-quality services to users, it is necessary to continue the services of each slice in the case of a failure. In traditional IP networks, when a modification such as a network failure occurs, the routing table of each router is dynamically changed using a routing protocol such as Open Shortest Path First (OSPF) [8]. In such routing protocols, the routing table of each router is changed by exchanging information among neighboring nodes when a failure occurs. Therefore, the convergence time during which the routing table is unstable is long, and thus there is a risk that many packets forwarded during this period are dropped. To solve this problem, the Multiple Routing Configurations (MRC) method has been proposed in the past [6]. MRC ensures fast path availability in the case of a link/node failure by preparing multiple backup routing configurations in advance.

In this paper, we propose a load-balanced fast failure recovery method based on MRC in communications using network slicing. The proposed method makes backup routing configurations so as to balance the load of nodes assumed to be failed for each backup routing configuration. This will prevent unbalanced load among nodes after switching configurations. In addition, we implement our proposed method with the use of P4 (Programming Protocol-Independent Packet Processors) [4] in software-defined networking (SDN) environments. P4 is a programming language for SDN and it allows us to flexibly define the functions of network devices. By confirming the operation using Mininet [1], we show that routing paths are ensured by switching routing configurations in the case of a failure.

II. Multiple routing configurations (MRC)

MRC is a technology for fast recovery from a single link/node failure in a network. MRC prepares multiple backup routing configurations according to failure points in advance. When a node or link fails on a normal routing configuration, MRC immediately switches the routing configuration to a backup routing configuration corresponding to the failure point. By doing so, data transmission can continue without packet losses.

On each backup routing configuration, nodes are classified into normal nodes and isolated nodes. Also, links are classified into normal links, isolated links, and restricted links. The normal nodes and the normal links can be used for packet transmission on the backup routing configuration, while the isolated nodes and the isolated links do not carry any traffic. The restricted links are used only for the first hop or the last hop of packet transmission whose source or destination is an isolated node in such a way that the packets can reach an isolated node on the backup configuration.

MRC guarantees tolerance for a single link/node failure by isolating each link and each node on at least one backup routing configuration. On each backup routing configuration, all the node pairs must be connected by a path that does not pass through an isolated node or an isolated link. Therefore, each configuration must satisfy the following constraints.

- Respective nodes and links become isolated nodes and isolated links on at least one backup routing configuration.

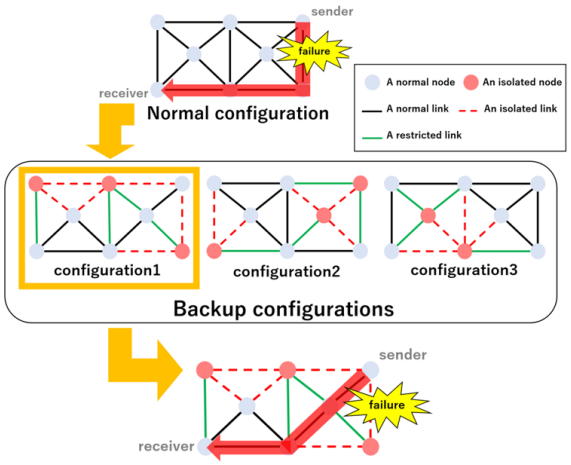

Fig. 1. Switching the routing configuration in the case of a failure.

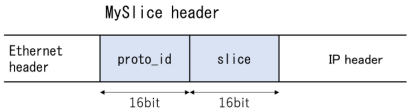

Fig. 2. Structure of the MySlice header.

- On each backup routing configuration, there exists a connected graph composed of all the normal nodes and the normal links.
- Isolated nodes are connected with isolated or restricted links, and they are not connected with normal links.
- At least one of the links connected to an isolated node is a restricted link.
- At least one of the nodes connected to an isolated link is an isolated node.

Fig. 1 shows an example of a normal configuration and backup configurations. When a failure occurs in a link, a node connected to the link detects the failure. In the case of a node failure, a neighboring node detects it. Then, data transmission can be continued by selecting a backup routing configuration where the failed link (node) is isolated and switching to it.

III. Proposed Method

In the proposed method, we implement network slicing and MRC with P4, taking into account a load of backup routing configurations. P4 is a language to program the behavior of data planes (i.e., functions used for data transmission in network devices) [7]. We can define and add the functions required for data transfer in software by using P4, which enables us to more flexibly control SDN environments [5].

A. Implementation of network slicing and MRC

In P4, packets can be forwarded using user-defined headers. To identify slices, we define a header named MySlice between the Ethernet header and the IP header, as shown in Fig. 2. The proto_id field is used to identify whether the IP header follows the MySlice header. The slice field is used to identify slices in which the packet is being sent. In our implementation, an individual routing table for each slice is created in each SDN switch (node). By processing incoming packets according to the MySlice header and the routing table, it is possible to forward the packets to appropriate output ports.

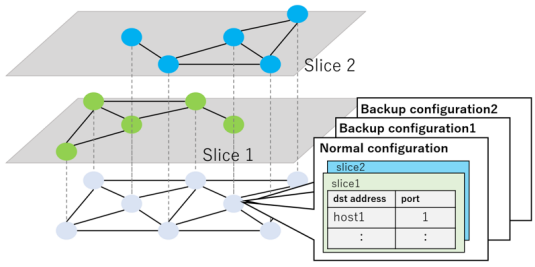

Fig. 3. Creation of routing tables.

TABLE I
List of Symbols.

Symbol	Definition
$\mathcal{N}$	Set of nodes
$\mathcal{N}(u)$	Set of adjacent nodes of node u
$\mathcal{S}$	Set of slices
(u, v)	Unidirectional link from node u to node v
$\mathcal{D}_i$	Set of isolated nodes in backup routing configuration i
n_C	Number of backup routing configurations
$l_a(u, v)$	Load on link (u, v) in slice a
$\gamma(u)$	Total potential of node u
$\gamma_a(u)$	Potential of node u in slice a

Furthermore, in our implementation, routing tables for backup routing configurations in addition to the normal routing configuration are created for each slice as shown in Fig. 3. After switching to a backup routing configuration in a case of failure, each SDN switch needs to identify which configuration is selected in order to forwards incoming packets to appropriate output ports. To do so, we use the Type Of Service (TOS) field in the IP header of the incoming packets. In the TOS field, the configuration ID that indicates the backup routing configuration to be used is recorded. Each switch forwards the incoming packets according to the routing table for the backup routing configuration corresponding to the configuration ID.

B. Creating backup routing configurations

In MRC, isolated nodes and isolated links in each backup routing configuration are not used for packet transmission. Therefore, some problems such as a sudden increase in the number of hops of routing paths and unbalanced traffic load may occur after switching to a backup routing configuration. One of the causes of these problems is that multiple nodes through which a large amount of traffic pass in the normal configuration are selected as isolated nodes on the same backup configuration. In order to overcome this problem, in [6], a selection method of isolated nodes has been proposed. In this method, the sum of loads of the links (i.e., traffic

volume passing through the links) connected to each node is defined as the potential of the node. The potential indicates the load of the node in this case. This method creates backup routing configurations such that the sums of the potentials of the isolated nodes on the backup routing configurations become even. By doing so, it can distribute the effect of isolated links and isolated nodes on each backup routing configuration because the potential of each node represents how much traffic the node should process. It has shown that the selection method performs better load balancing than a conventional routing protocol.

Our method proposed in this paper extends this selection method of isolated nodes, assuming the use of MRC in network slicing environments. Table I shows the symbols used in this paper. We assume that a network and slices are given. On each slice, flows between given sources and destinations are transmitted. Each flow has a certain amount of traffic demand. For each slice $a \in \mathcal{S}$, the proposed method first calculates the potential $\gamma_a(u)$ of each node $u \in \mathcal{N}$ on the normal routing configuration as follows:

$$\gamma_a(u) = \sum_{v \in N(u)} (l_a(u,v) + l_a(v,u)), \quad (1)$$

where link load $l_a(u,v)$ of each slice is given by the sum of the amount of traffic demand of flows that pass through the link on the slice. Then, the proposed method calculates the total potential $\gamma(u) = \sum_{a \in \mathcal{S}} \gamma_a(u)$ of each node $u \in \mathcal{N}$. The proposed method creates backup routing configurations according to the total potentials of each node. Specifically, nodes having small total potentials are isolated in the same backup routing configuration. On the other hand, nodes having large total potentials are not isolated together in the same configuration. By doing so, the proposed method creates backup routing configurations such that the sums of the total potentials of the isolated nodes on the configurations become even as much as possible.

The specific procedure of creating backup routing configurations in the proposed method is as follows.

1) Sorts all the nodes in an ascending order of their total potentials. Let $\mathcal{L} = \{u_1, u_2, \ldots, u_{|\mathcal{N}|}\}$ denote the sorted node list.
2) $i \leftarrow 1$ and $j \leftarrow 1$.
3) Find k given by

$$\min\{k \mid \sum_{l=j}^{k} \gamma(u_l) > \gamma + \delta, k = j, j+1, \ldots, |\mathcal{N}|\}, \quad (2)$$

where $\gamma = \sum_{u \in \mathcal{N}} \gamma(u)/n_C$ and δ is a parameter.
4) Assign nodes $u_j, u_{j+1}, \ldots, u_k$ to tentative backup routing configuration i. Then, $i \leftarrow i+1$, $j \leftarrow k+1$, and go to step 3).
5) If k is not found in (2), assign the remaining nodes to tentative backup routing configuration i. Then, apply the MRC algorithm proposed in [6] to determine actual backup routing configurations.

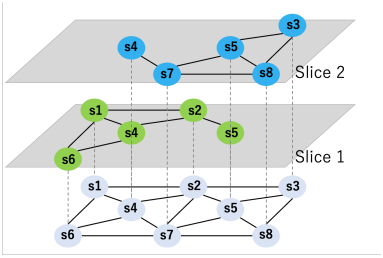

Fig. 4. Network model.

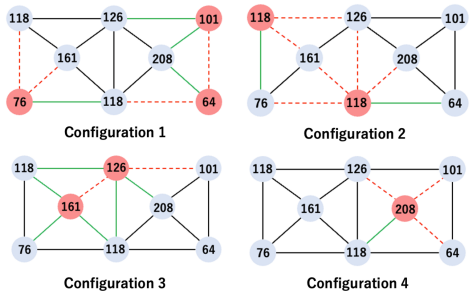

Fig. 5. Backup routing configurations constructed by the proposed method (red nodes/links represent isolated nodes/links).

In steps 3) and 4), the proposed method aims to divide the nodes into n_C groups where the sum γ_i of the total potentials of the isolated nodes on each backup routing configuration i (i.e., $\gamma_i = \sum_{u \in \mathcal{D}_i} \gamma(u)$) is approximately equal to $\gamma = \sum_{u \in \mathcal{N}} \gamma(u)/n_C$.

IV. Demonstration Experiments

A. Confirmation of load balancing

In order to confirm the behavior of the proposed method, we conduct demonstration experiments. We use the network shown in Fig. 4, where there exist two slices. Each node fills the role of a sender, a receiver, and an intermediate node. The traffic demand of each sender-receiver pair is randomly selected from [1, 20]. We create four backup routing configurations. We here calculate the potential values for each backup configuration to confirm that the proposed method can distribute the potential values to each backup configuration in a network slicing environment.

Fig. 5 shows the configurations constructed by the proposed method where the number on each node represents its total potential $\gamma(u)$. As we can see from Fig. 5, the nodes having the small potentials are collectively isolated in configuration 1 while the node having the large potential is isolated alone in configuration 4. Furthermore, Table II shows the sum γ_i of the total potentials of the isolated nodes on each configuration i and their coefficients of variation (CV). For the sake of comparison, we show the result of a first-fit method where nodes are isolated in numerical sequence shown in Fig. 4. We observe that the proposed method reduces CV of the total potential of isolated nodes. Therefore, the proposed method

TABLE II
POTENTIAL VALUES.

	Conf. 1	Conf. 2	Conf. 3	Conf. 4	CV
First-Fit	326	202	219	225	0.200
Proposal	241	236	287	208	0.117

Fig. 6. Header information of packets on slice 1.

can create backup configurations considering load distribution in a network slice environment.

B. Implementation of MRC for network slicing

In order to confirm the operation of the proposed method, we evaluate the implementation with P4 on Mininet [1]. We construct the network and slices shown in Fig. 4, where each node represents an SDN switch connected to one host (e.g., host 1 connects to switch 1).

We first confirm the operation of network slicing without failures. We here check the behavior of packet forwarding with the MySlice header defined in our proposed method by analyzing packets using scapy [2]. We assume that packets are transmitted from host 1 to host 5 in slice 1. In this case, host 1 sends packets with the slice field set to 1 in the MySlice header. Fig. 6 shows the header information of transmitted packets at sender host 1 and receiver host 5. From this figure, we observe that the MySlice header is inserted between the Ethernet header and the IP header. Each switch along the routing path recognizes the value of the slice field and processes the packets according to the routing table of slice 1. As a result, the packets can arrive at receiver host 5.

Next, we assume that packets are transmitted from host 1 to host 5 in slice 2. To do so, the slice field in the MySlice header of the packets is set to 2. Switch 1 connected to host 1 recognizes that the slice field of the incoming packets is 2 and refers to the routing table of slice 2. However, there is no action in the routing table for the packets whose destination address is host 5 on slice 2 because slice 2 does not include switch 1. Therefore, the packets are dropped at switch 1. From these results, we confirmed that the switches adequately process incoming packets based on the routing tables configured for slices.

Finally, we confirm that the operation of the MRC with the backup routing configurations shown in Fig. 5. We assume that a failure on the link (s1, s2) is occurred by using the "link s1 s2 down" command of Mininet. Under this assumption, we send packets from host 1 to host 5 on slice 1. On the normal routing configuration, the routing path is [s1, s2, s5] as shown in Fig. 7. The packet cannot reach host 5 because it contains a failed link. On the other hand, by using backup routing configuration 2 where the link (s1, s2) is isolated, the packets whose TOS is set to 2 arrive at host 5 as shown in Fig. 8. From these results, we can see that the proposed method ensures routing paths in the case of a failure.

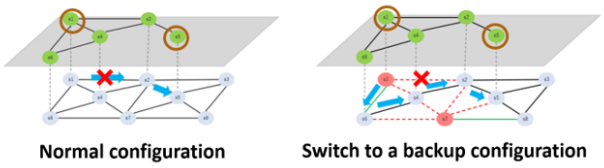

Fig. 7. Ensuring routes by MRC.

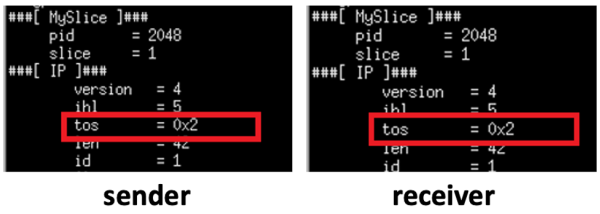

Fig. 8. Packets after switching the routing configuration.

V. CONCLUSION

In this paper, we proposed a load-balanced fast failure recovery method based on MRC in communications using network slicing. We implemented the proposed method with P4, and confirmed its behavior.

Acknowledgment This research was supported by SCOPE of the Ministry of Internal Affairs and Communications, Japan, under Grant No. 191605004.

REFERENCES

[1] Mininet, http://mininet.org/
[2] scapy, https://scapy.net/
[3] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.
[4] P. Bossharty, D. Daly, M. Izzard, N. McKeown, J. Rexford, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "Programming Protocol-Independent Packet Processors," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87–95, Dec. 2013.
[5] H. Harkous, M. Jarschel, M. He, R. Pries, and W. Kellerer, "Towards Understanding the Performance of P4 Programmable Hardware," in *Proc. Symposium on Architectures for Networking and Communications Systems 2nd EuroP4 Workshop*, Cambridge, UK, September 2019.
[6] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Multiple Routing Configurations for Fast IP Network Recovery," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 473–486, Apr. 2009.
[7] D. Scholz, H. Stubbe, S. Gallenmuller, and G. Carle, "Key Properties of Programmable Data Plane Targets," in *Proc. 32th International Teletraffic Congress (ITC 32)*, Osaka, Japan, 2020.
[8] V. Vetriselvan, P. R. Patil, and M. Mahendran, "Survey on the RIP, OSPF, EIGRP Routing Protocols," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 1058–1065, 2014.