An Improvement of an IP Fast Reroute Method Using Multiple Routing Tables

Yuya Tarutani[1*], Masashi Ishigai[2], Naoto Numata[2], Yukinobu Fukushima[3], Tokumi Yokohira[1]

[1] Faculty of Interdisciplinary Science and Engineering in Health Systems, Okayama University, Japan
[2] Graduate School of Interdisciplinary Science and Engineering in Health Systems, Okayama University, Japan
[3] Faculty of Natural Science and Technology, Okayama University, Japan
y-tarutn@okayama-u.ac.jp, ishigai00net@s.okayama-u.ac.jp, numata00net@s.okayama-u.ac.jp,
fukushima@okayama-u.ac.jp, yokohira@okayama-u.ac.jp

Abstract

The demand for high Internet availability has increased. As a routing method to recover fast from a single network component (link or node) failure, a method using multiple routing tables has been proposed. In the method, when a component failure occurs, packets which are supposed to pass through the component in the failure free state are rerouted using a backup routing table (backup table: BT) which does not use the component. Thus, it is important to decrease the increased routing cost, and because the method uses multiple BTs, it is also important to decrease the numbers of entries in BTs. In this paper, we propose a new routing method to decrease the increased routing cost. In the conventional method, there are many components that do not fail but are not used for packet forwarding in a BT. In our proposed method, we use such components as much as possible to decrease the increased routing cost. About the decreasing of the numbers of entries in BTs, we simulate packet forwarding for every node-pair and find unnecessary entries which are not used in packet forwarding in any node-pair. Numerical examples show that our proposed method is superior to the conventional method.

Keywords: Fast network recovery, IP fast reroute, Multiple routing tables, OSPF

1 Introduction

Our daily life deeply depends on various network services such as E-Mail, Web shopping and viewing videos. Since a lot of network services utilize Internet as their infrastructure networks, the halt of network services due to failures of links or nodes in Internet is a big problem. Thus Internet is required to have high availability.

OSPF (Open Shortest Path First) is widely used as an intra-AS (Autonomous System) routing scheme and networks based on OSPF attain high performance in the failure-free situation. However, because failure information has to be sent to every node, OSPF's recover speed from network failures is not so high, and consequently networks go into an unstable state in a failure situation and their performance is poor.

In order to recover fast from network failures, many methods called IP fast reroute (IPFRR) methods have been proposed. In the methods, packets encountering failures are rerouted to other routes which do not pass through the failure components, and the reroutes can be attained locally without sending failure information to every node.

As IPFRR methods focusing on the single network component failure model where a single link or node can fail and when a failure occurs, no nodes can know the kind of failure (whether the failure is a link failure or node failure), the following methods have been proposed.

Loop-Free Alternates (LFA) method [1] based on the concept of Equal Cost Multiple Path (ECMP [2]) is a famous IPFRR method and is implemented in many routers. LFA method reroutes packets encountering network failures to loop-free alternate reroutes which satisfy some conditions. For example, a candidate reroute path of a packet encountering failure is a path which has the same number of hops as the primary path of the packet. Although LFA method is good in that it is easy to be implemented, it can not generally attain complete failure coverage, that is, there is a possibility that it can not find any reroute against some failures. U-turn method [3] has been proposed as an extension of LFA method. Although LFA method can not allow that packets return back to a node through which they have passed, U-turn method allows such return and improves failure coverage. However, it can not still attain complete failure coverage.

In ESCAP (Efficient SCan for Alternate Paths [4]), FIFR (Failure Inferencing Fast Reroute [5-7]) and DisPath (Disjoint Paths Recovery [8]) methods, when a packet arrives at a node from an unusual interface, the methods consider that a failure occurs, and reroute the packet to a route which is considered not to pass through the failure component. Although they can attain complete failure coverage, they have the following drawback. If a failure is permanent, reconvergence of the normal routing table starts and packets are routed based on the new normal routing table. Such packets coexist with packets which are rerouted due to a failure, and the two kinds of packets can not be discriminated in the methods. Thus, it is hard to use a mechanism [9] which avoid micro-loops during the reconvergence of routing tables.

In Not-Via method [10], a failure detecting node creates a tunnel to a node (exit node) and sends packets to the exit node using the tunnel. In the exit node, packets are decapsulated and are sent to their destination nodes. The tunnel is created so that the failure is bypassed. As a similar approach to Not-Via method, TI-LFA (Topology Independent LFA [11]) method has been currently proposed. In TI-LFA, a failure detecting node establishes a protection path like a tunnel which can

---

*Corresponding Author: Yuya Tarutani; E-mail: y-tarutn@okayama-u.ac.jp

DOI: 10.53106/160792642022112306014

bypass the failure component using segment routing. Although these methods can also attain complete failure coverage, the processing of the tunnel (protection path) can be a heavy burden for routers.

Kvalbein et al. have proposed a reroute method [12-13] to use two kinds of routing tables, one normal routing table and multiple backup routing tables (BTs simply). The former is used in the failure-free state and each of the latter is used in a failure state. Each BT is directly created from one respective network topology called a backup (routing) configuration (we call it BC simply). In each BC, some links and nodes in a given topology are assumed to fail and consequently not to be used. When a failure occurs, the method uses the BT created from the BC where the failure component is assumed to fail. Because each node can know whether a packet is rerouted due to a failure or is routed based on the new normal routing table, we can use the micro-loop avoidance method described above. We call the method MRC (Multiple Routing Configurations) method.

In MRC method, reroute paths are not generally the shortest paths of rerouted packets, and consequently the routing cost (for example, the number of hops used by a rerouted packet if every link cost is one) of each rerouted packet generally increases compared to that in the failure-free state. In addition, since MRC method uses multiple BTs, it is very important to reduce the number of entries in each BT.

In this paper, we propose an improved version of MRC method which we call GLA (Global Look Ahead) method. In MRC method, it is assumed that only one component fails (single component failure model). Thus, when a node detects a failure link, although the adjacent node which connects to the detecting node by the failure link and the adjacent node's links may fail, we can consider that the other components do not fail. MRC method does not use components even if they can be considered not to fail. GLA method tries to use such components as much as possible to decrease the increased costs in reroute packets. With respect to BT entry reduction, we first clarity that some entries in a BT are not used even if any single component fails in both of MRC and GLA methods, and then we clarity the number of such entries using simulation runs.

Table 1 summarizes the characteristics of each IPFRR method described above. From Table 1, we consider that MRC and GLA methods are better methods for the single component failure model, and we consider that GLA method is better than MRC method because the increased routing cost of GLA method is smaller and BT entry reduction rate of GLA method is much higher as described later.

The rest of the paper is organized as follows. Section 2 describes the concept of BC, packet reroute procedures in MRC method and describes its problems described above in detail. Section 3 describes GLA method. Section 4 BT entry reduction in both of MRC and GLA methods. Section 5 gives some numerical examples. Section 6 concludes the paper.

Table 1. Comparison of IPFRR methods against single network component failure

	Complete failure coverage	Avoidance of micro-loops	tunneling
LFA U-turn	No	No	No
ESCAP FIFR DisPath	Yes	No	No
Not-Via TI-LFA	Yes	Yes	Yes
MRC GLA	Yes	Yes	No

2 Conventional Packet Reroute Method

In this paper, we assume that every link cost is one for the simplicity of the description. Thus, the routing cost of each packet is equal to the number of hops passed through by the packet. We can easily relax the assumption so that an arbitrary link cost is taken into account.

2.1 Backup Configurations

Each BC has two types of nodes (normal node and isolated node) and three types of links (normal link, isolated link and restricted link). In order to recover from one component (link or node) failure, a set of BCs has to satisfy the following properties.

(P-1) In every BC, every isolated node has restricted and isolated links only, and the number of the restricted links is at least one and the number of the isolated links is larger than or equal to zero.

(P-2) In every BC, suppose the sub-topology where all isolated and restricted links are removed and all isolated nodes are also removed from the BC. If the sub-topology is a connected graph, which has a path (route) between arbitrary two nodes, then we call it *a backbone*. Every BC has to contain the backbone and every isolated node in the BC has to connect to the backbone using at least one restricted link.

(P-3) Every link and every node have to be isolated in at least one BC.

Figure 1 shows an example of BCs. Figure 1(a) is an original network topology, and Figure 1(b) to Figure 1(d) are BCs which are created from Figure 1(a). Figure 1(e) to Figure 1(g) show the sub-topologies in property (P-2) which are obtained from Figure 1(b) to Figure 1(d), respectively. We can easily confirm that BC_1 -BC_3 in Figure 1 satisfy conditions (P1)-(P3).

Papers [12-13] have proposed a BC creation method which creates such BCs that every link and every node are isolated in exactly one BC for property (P-3). Figure 1 is created by the method. We omit the description of the procedure in the method due to space limitation. Hereafter, we represent the link between nodes v_1 and v_2 by link v_1-v_2, and represents the BC where link v_1-v_2 (node v_1) is isolated by BC(v_1-v_2) (BC(v_1)).

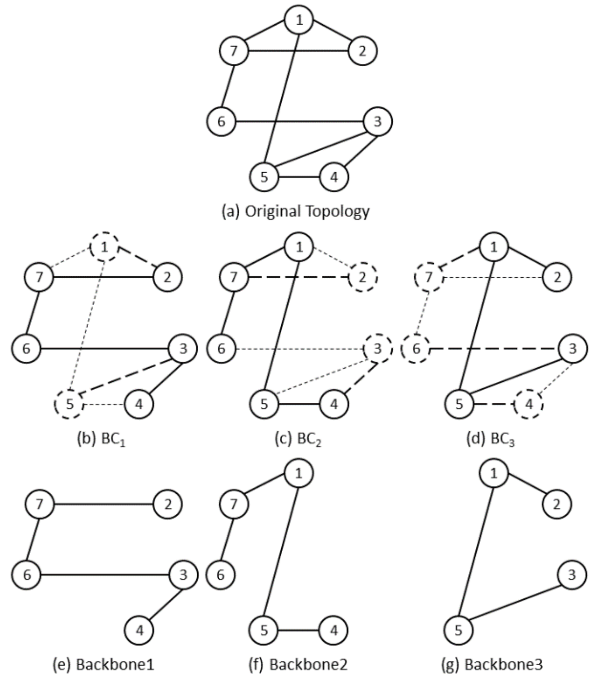


Figure 1. Example of an original topology, BCs and backbones

2.2 Packet Rerouting in MRC Method

One BT is directly obtained from each BC by calculating the shortest path between every node-pair under the condition that (i) a weight (routing cost) of each isolated link is infinite and (ii) a weight of each restricted link is a sufficiently large value. By imposing condition (i), no isolated links are included in any shortest path, and by imposing condition (ii), every restricted link can be included as the first or last hop only in any shortest path. All the created BTs are stored in all nodes in the network. For example, Table 2(a) to Table 2(c) show BTs of node 1 in Figure 1 which are obtained from BCs in Figure 1(b) to Figure 1(d), respectively. Hereafter we represent the BT which is created from BC(v_1-v_2) (BC(v_1)) by BT(v_1-v_2) (BT(v_1)) and the BT of node j which is created from BC$_i$ by BT_i^j.

Table 2. Backup tables of node 1 in MRC method

(a) BT_1^1		(b) BT_2^1		(c) BT_3^1	
Destination	Next node	Destination	Next node	Destination	Next node
2	2	2	7	2	2
3	2	3	5	3	5
4	2	4	5	4	5
5	2	5	5	5	5
6	2	6	7	6	5
7	2	7	7	7	7

Although packets are forwarded based on the normal routing table in the failure-free state, a BT is used for forwarding of packets encountering a component failure. In MRC method, the ID of a routing table used for a packet is assumed to be memorized in the packet's header. That is, the ID of the normal routing table is memorized at the source node of a packet and the ID of a BT is memorized at a failure detecting node. Thus, each node forwards each received packet using the routing table whose ID is equal to the memorized one in the header of the packet. In this paper, we assume that the ID of the normal routing table is 0 and that of BT$_i$ is its suffix i.

In a link failure case, MRC method tries to deliver packets of every node-pair. On the other hand, in a node failure case, MRC method assumes that all the links of the failure node also fail and no packets are generated from it, and delivers packets of every node-pair which does not include the failure node and drops packets destined for the failure node inside the network. Such forwarding is attained by the following procedure.

[Packet Forwarding Procedure in MRC method]

Assume that node u receives a packet with destination node d (destination d means the egress or the final destination of the packet in the routing domain) and whose next hop node and link are v and link u-v, respectively, according to the routing table which is designated by the ID in the packet header.

Step 1: If link u-v does not fail, forward the packet to v and terminate the procedure. Otherwise go to Step 2.

Step 2: If the ID of the packet is larger than 0 (that is, the packet has already encountered another failure and experienced a reroute), then drop the packet and terminate the procedure. Otherwise (that is, the packet encounters a failure for the first time) go to Step 3.

Step 3: Execute the following Step 3-1 or Step 3-2 and go to Step 4.

Step 3-1: If v is not identical to d, then node u selects BT(v).

Step 3-2: If v is identical to d, node u selects BT(u-v).

Step 4: Memorize the ID of the selected BT in Step 3 in the packet header and forward the packet according to the selected BT.

[Example 1: Packet Forwarding Procedure]

For example, in Figure 1, the shortest path from node 1 to node 3 is 1 → 5 → 3 (we denote the path which passes through nodes $v_1, v_2, \dots, v_r$ by $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_r$) in the failure-free state. Assume that link 1-5 fails. A packet generated at node 1 encounters link 1-5 failure and BT(5) (that is, the routing table obtained from BC(5), which is BC$_1$) is selected in Step 3-1. Then the packet is forwarded using BC$_1$, and consequently the packet's reroute route is 1 → 2 → 7 → 6 → 3, which is increased by 2 hops compared to the failure-free state.

The selection of BT in Step 3 is done using the table shown in Table 3 in the network operation stage. We call such table *decision table*. Each entry means that if the destination node of a rerouted packet is the destination node in the entry, the ID of the selected BT is ID in the entry. For example, a packet destined for node 3 uses BT_1^1 when link 1-5 fails as described above. So 1 is stored as ID in the entry of destination node 3 of the decision table.

Table 3. Decision table of node 1 in MRC method

Destination	ID
2	2
3	1
4	1
5	1
6	3
7	1

In MRC method, changing of the ID from 0 to i ($i > 0$) in each packet occurs at most one time. Packets experiencing the second failure are dropped in Step 2. Such dropping can occur when single component failure assumption is not satisfied, that is, multiple component failures occur. On the other hands, note that packets which are supposed not to pass through the failure component in the failure-free state are forwarded on the same path even in a failure state. Thus, the numbers of hops in paths of such packets remain unchanged even in a failure state.

3 Proposed Reroute Method

3.1 Idea

In MRC method, a failure detecting node determines the BC which bypass the failure considering local failure situation, and even if a link of the failure detecting node does not fail, the link is not used for rerouting if the link is isolated in the BC. For example, in Example 1, link 1-7 does not fail because we assume single component failure model and link 1-5 fails. Thus, we can use link 1-7. However, because link 1-7 is isolated in BC_1, MRC method does not use link 1-7.

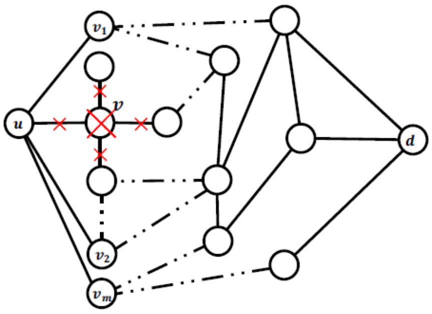


Figure 2. An example network which shows the relation between a failure detecting node and doubtful failures

Our proposed method (GLA method) tries to decrease the number of hops of a rerouted packet using the alive link described above. Generally speaking, when there is an alive link of the failure detecting node and the packet is assumed to be forwarded to its opposite node, if we reroute the packet from the opposite node to the destination node using another BC or original topology (for the simplicity of the description, we refer the original topology as BC_0), there is a possibility that the total number of hops becomes smaller. For example, in Figure 2, assume that a packet destined for node d arrives at node u and the packet should be forwarded to node v in the failure-free state, and assume that node u detects a failure of one (link u-v) of its links (Note that adjacent node v may also fail) and it selects BC_i for packet rerouting in MRC method. When another link u-v_1 is alive and the packet goes to node v_1, the packet may be able to get to its destination node d with shorter hops by using another BC (say BC_j: $j \neq i$). For example, in Example 1, because link 1-7 does not fail, we can forward the packet to node 7. And from node 7, if we forward the packet using the original topology, the packet can get to node 7 via path 7 → 6 → 3 without using the failed link 1-5. Thus, overall reroute path can become 1 → 7 → 6 → 3 and consequently the number of hops decreases by one compared to MRC method.

Based on the idea described above, we describe GLA method in detail in the succeeding section.

3.2 Packet Rerouting in Proposed Method

In Figure 2, assume that the shortest path from node u to node d in BC_0 (that is, the original network topology) includes node v as the next hop node from u. And node u has m links, link u-v_i ($1 \leq i \leq m$) in addition to link u-v. And assume that we have n BCs, BC_j ($1 \leq j \leq n$), and the original topology BC_0. Let $P_j^d(v_i)$ be the shortest path from u to d in BC_j ($0 \leq j \leq n$) when we first transmit a packet from u to v_i and we forward the packet from v_i to d using BC_j. Table 4 shows all $P_j^d(v_i)$s ($1 \leq i \leq m;\ 0 \leq j \leq n$). For example, in $P_0^d(v_1)$, a packet is sent to v_1 and then the packet is sent to d according to BC_0.

Table 4. Concept of $P_j^d(v_i)s$

$P_0^d(v_1)$	$u \rightarrow v_1 \overset{BC_0}{\longrightarrow} d$
$P_1^d(v_1)$	$u \rightarrow v_1 \overset{BC_1}{\longrightarrow} d$
⋮	⋮
$P_j^d(v_1)$	$u \rightarrow v_1 \overset{BC_j}{\longrightarrow} d$
⋮	⋮
$P_n^d(v_1)$	$u \rightarrow v_1 \overset{BC_n}{\longrightarrow} d$
⋮	⋮
$P_0^d(v_i)$	$u \rightarrow v_i \overset{BC_0}{\longrightarrow} d$
$P_1^d(v_i)$	$u \rightarrow v_i \overset{BC_1}{\longrightarrow} d$
⋮	⋮
$P_j^d(v_i)$	$u \rightarrow v_i \overset{BC_j}{\longrightarrow} d$
⋮	⋮
$P_n^d(v_i)$	$u \rightarrow v_i \overset{BC_n}{\longrightarrow} d$
⋮	⋮
$P_0^d(v_m)$	$u \rightarrow v_m \overset{BC_0}{\longrightarrow} d$
$P_1^d(v_m)$	$u \rightarrow v_m \overset{BC_1}{\longrightarrow} d$
⋮	⋮
$P_j^d(v_m)$	$u \rightarrow v_m \overset{BC_j}{\longrightarrow} d$
⋮	⋮
$P_n^d(v_m)$	$u \rightarrow v_m \overset{BC_n}{\longrightarrow} d$

In the network design stage, first we derive $P_j^d(v_i)$s and then we select the path (say $P_{j^*}^d(v_{i^*})$) with the smallest number of hops among all $P_j^d(v_i)$s (If several paths have the smallest number of hops, we randomly select one path). If $P_j^d(v_i)$ includes v (that is, v is an intermediate node of $P_j^d(v_i)$ and v exists between v_i and d), we consider that the number of hops of $P_j^d(v_i)$ is infinite because node v may fail. If $v = d$ and $P_j^d(v_i)$ includes link u-v (that is, link u-v is the last hop of $P_j^d(v_i)$), the number of hops of $P_j^d(v_i)$ is infinite because link u-v fails. It is trivial that the path selected in MRC method is included among all $P_j^d(v_i)$s, and consequently the number of hops of $P_{j^*}^d(v_{i^*})$ is less than or equal to that of the path selected in MRC method.

Table 5 shows all $P_j^d(v_i)$s when a packet destined for node 3 encounters link 1-5 failure at node 1. For example, when we assume that the packet is sent according to $P_0^3(2)$, the path of the packet is 1→2→7→6→3 and the number of hops is 4. More interesting case is $P_2^3(2)$. According to $P_2^3(2)$, the path of the packet is 1→2→7→1→5→4→3. The packet is return to node 1. The number of hops is ∞ because the path includes node 5 (node v) as an intermediate node. $P_1^3(2)$ is the path obtained by MRC method. In Table 5, $P_0^3(7)$ and $P_1^3(7)$ have the smallest number of hops. As described before, we randomly select one path when several paths have the smallest number of hops. If we select $P_0^3(7)$, j^* and v_{i^*} are 0 and node 7, respectively.

Table 5. $P_j^d(v_i)s$ when node u is node 1 in Figure 1

$P_j^3(v_i)$	$u \rightarrow v_i \rightarrow d$	The number of hops
$P_0^3(2)$	1 → 2 → 7 → 6 → 3	4
$P_1^3(2)$	1 → 2 → 7 → 6 → 3	4
$P_2^3(2)$	1 → 2 → 7 → 1 → 5 → 4 →3	∞
$P_3^3(2)$	1 → 2 → 1 → 5 → 3	∞
$P_0^3(7)$	1 → 7 → 6 → 3	3
$P_1^3(7)$	1 → 7 → 6 → 3	3
$P_2^3(7)$	1 → 7 → 1 → 5 → 4 → 3	∞
$P_3^3(7)$	1 → 7 → 1 → 5 → 3	∞

For each j ($0 \leq j \leq n$), all $P_j^d(v_i)$s ($1 \leq i \leq m$; $1 \leq d \leq N$; N is the number of nodes in the original topology) can be obtained using the famous Floyd's shortest path algorithm and the time complexity is O(N^3). Therefore, the overall time complexity to calculate all $P_j^d(v_i)$s is O(nN^3), which is enough small to calculate them in the network design stage.

In the network operational stage, when node u receives a packet destined for d and detects the failure of link u-v, node u uses $P_{j^*}^d(v_{i^*})$, that is, node u forwards the packet to v_{i^*} and the packet is forwarded from v_{i^*} using BC_{j^*}.

In GLA method, even after a packet experiences a failure, it may be rerouted according to BC_0. On the other hand, we would like to drop the packet like MRC method when it encounters another failure due to multiple component failures. Thus, we introduce new ID $n + 1$ for BC_0 and we put $n + 1$ in the packet header when the failure detecting node determines that it reroute the packet using BC_0. Therefore, when a node receives a packet with ID $n + 1$ and the packet's next hop fails, the node drops the packet.

Packet forwarding procedure in GLA method is similar to that in MRC method except that Step 3 is executed using a decision table shown in Table 6 where v_{i^*} and j^* are stored for each d and Step 4 is changed as follows.

Step 4: Memorize the ID of the selected BT in Step 3 in the packet header and forward the packet to node v_{i^*}.

In Table 6, since five IDs of six entries are 4s, many packets experiencing failures at node 1 are forwarded using the original topology BC_0 (=BC_4). Such situation can occur more frequently when destination node d is farther from detecting node u, because node v and its links may fail, other components do not fail and consequently components around d can be alive with a higher probability. Thus, GLA method can attain much smaller numbers of hops of reroute packets than MRC method because MRC method does not use the original topology for packet rerouting.

Table 6. Decision table of node 1 in GLA method

Destination	Next node	ID
2	7	4
3	7	4
4	7	4
5	7	1
6	5	4
7	2	4

4 Decreasing of The Numbers of Entries in BTs

There can be some entries in BTs of MRC and GLA methods which are not used in the network operational stage. For example, we focus on a packet destined for node 6 in Figure 1 in MRC method. If the packet is generated or arrived at either of nodes 1, 2 (4, 5) and detects the failure of its next hop link, the ID of the packet is determined 3 (2) according to Step 3-1 because the next hop node in the original topology is not identical to destination node 6 and destination node 6 is isolated in BC_3 (BC_2). On the other hand, if the packet destined for node 6 is generated or arrived at node 3 (node 7) and the packet detects the failure of its next hop link, the ID of the packet is determined 2 (3) according to Step 3-2 because the next hop node is identical to destination node 6 and its next hop link is isolated BC_2 (BC_3). Thus, the ID of 1 is not memorized in the header of any packet destined for node 6. Therefore the entry of destination node 6 in BT_1^i in any node i is not used.

We derived unused entries using such an exhaustive search that we generate a packet between every node-pair, forward the packet using the packet forwarding procedure in MRC method for every single component failure in the original topology in Figure 1 and obtain the entries which are not used for forwarding. In the same way, we derived unused entries in GLA method. Table 7 and Table 8 show the results. Entries "-" mean unused entries. We can see that the number of unused entries in GLA method is larger than that in MRC method. This is because GLA method has higher probability of the use of BC_4 than MRC method as described before.

Unused entries described above can be removed from BTs used in the network operation stage. Thus, the numbers of entries in BTs in GLA method are expected to be much smaller than those in MRC method.

Table 7. Unused entries in BTs of node 1 in MRC method

(a) BT $_1^1$

Destination	Next node
2	-
3	2
4	2
5	2
6	-
7	2

(b) BT $_1^2$

Destination	Next node
2	7
3	5
4	5
5	5
6	7
7	-

(c) BT $_1^3$

Destination	Next node
2	2
3	5
4	5
5	-
6	5
7	7

Table 8. Unused entries in BTs of node 1 in GLA method

(a) BT $_1^1$

Destination	Next node
2	-
3	-
4	-
5	-
6	-
7	-

(b) BT $_1^2$

Destination	Next node
2	-
3	-
4	5
5	-
6	7
7	-

(c) BT $_1^3$

Destination	Next node
2	-
3	5
4	-
5	-
6	5
7	7

5 Numerical Results

We generated ten sample network topologies of Waxman and Barabasi-Albert (BA) models using BRITE topology generator. One topology with N nodes (N = 10, 20, 40, 60, 80) and the minimum node degree of two is generated in each of Waxman and BA models. The number (L) of links in the generated topology with N nodes for Waxman model is 2N, and that for BA is 2N-3. We denote Waxman and BA topologies with N nodes by Wax$N$ and BAN, respectively. Then, assuming that only one link or node in each topology fails, we simulated OSPF, MRC and GLA methods.

In this section, using the numerical data obtained the simulation described above, we first discuss increasing in the number of hops due to link or node failure and then we discuss decreasing in the number of entries in backup tables.

5.1 The Numbers of Hops

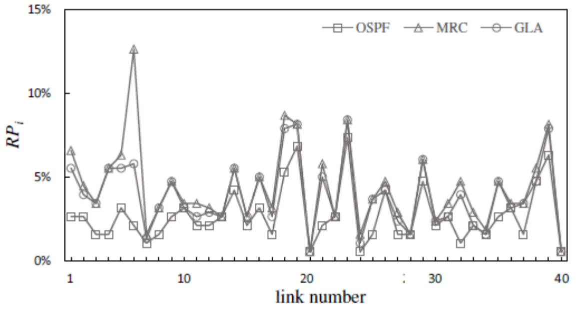
5.1.1 One Link Failure Case

Let P_{sd} be the number of hops of the shortest path from node s to node d in failure-free state and let P_{sd}^i be the number of hops of the path from node s to node d when link i fails in OSPF, MRC and GLA methods.

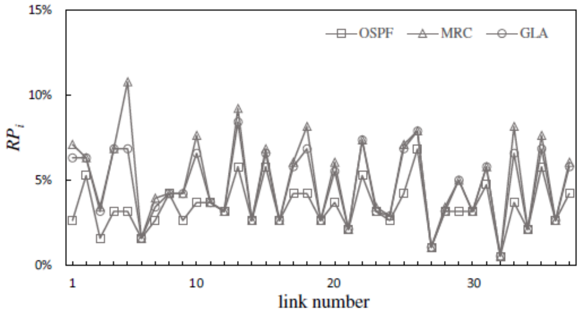
Figure 3(a) shows the ratio of node-pairs (s-d pairs) whose numbers of hops increase when link i fails in the Wax20. Specifically, the ratio (say RP_i) is defined as follows ($1 \leq i \leq L$; $L = 40$):

$$RP_i = \frac{NP_i}{{}_N C_2},$$

where NP_i is the number of node-pairs (s-d) such that $P_{sd}^i - P_{sd} > 0$. In each method, at least a few percentage of node-pairs increase their numbers of hops and at most about 10 percentage of node-pairs increase their numbers of hops. As described later the average ratio of OSPF over all links is smaller than those of MRC and GLA, and that of GLA is slightly smaller than that of MRC. Similarly Figure 3(b) shows RP_i of BA20 (L = 37).



(a) RP_i of Wax20



(b) RP_i of BA20

Figure 3. RP_i of 20-node topology

Figure 4(a) shows the average number of (Say AIH_i) of increased hops of node-pairs whose numbers of hops increase due to the failure of link i in Wax20. Specifically, AIH_i is defined as follows:

$$AIH_i = \frac{1}{NP_i} \sum_{\substack{\text{all } s-d \text{ pairs} \\ \text{such that} \\ P_{sd}^i - P_{sd} > 0}} \left(P_{sd}^i - P_{sd}\right)$$

Similarly, Figure 4(b) shows AIH_i in BA20.

From Figure 4(a) and Figure 4(b), AIH_i of GLA method is close to that of OSPF and much lower than that of MRC method for most values of link number i.

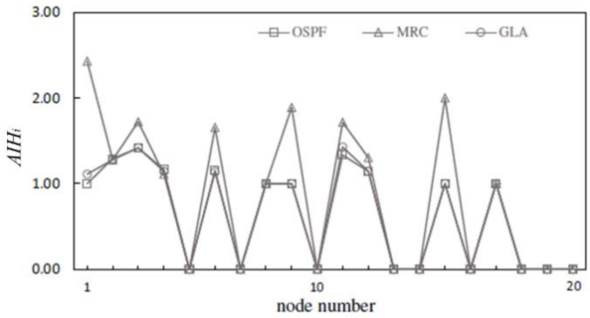
Figure 5(a) shows the average numbers (Say RP_{LF} and AIH_{LF}) of RP_is and AIH_is over all links for each of Waxman topologies. Specifically, RP and AIH are defined as follows:

$$RP_{LF} = \frac{1}{L}\sum_{i=1}^{L} RP_i\,, \quad AIH_{LF} = \frac{1}{L}\sum_{i=1}^{L} AIH_i$$

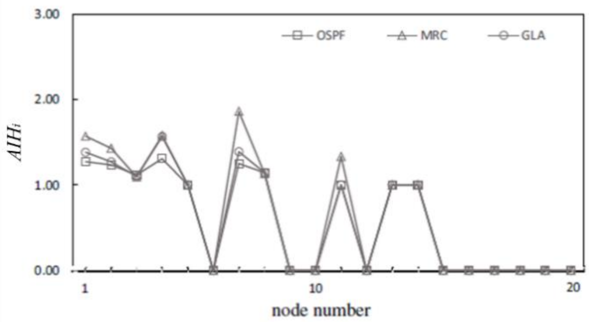
Similarly, Figure 5(b) shows RP_{LF} and AIH_{LF} for each of BA topologies.

In Figure 5, RP_{LF} of each method becomes smaller for larger value of N. This result can be described as follows. As described before, L=2N (L=2N-3) in the Waxman (BA) topologies used in the paper, which is O(N). On the other hand, the total number of node-pairs is N(N-1)/2, which is O(N^2).

Thus, generally speaking, although the number (say TNP_i) of node-pairs whose routes pass through each link i increases as N increases, the proportion of TNP_i to total number of node-pairs (N(N-1)/2) may decrease as N increases. Thus, the proportions of node-pairs which are influenced by a single link failure decreases, and consequently RP_{LF} decreases as N increases.

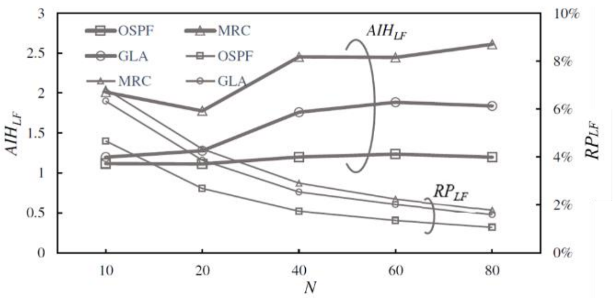


(a) AIHi of Wax20

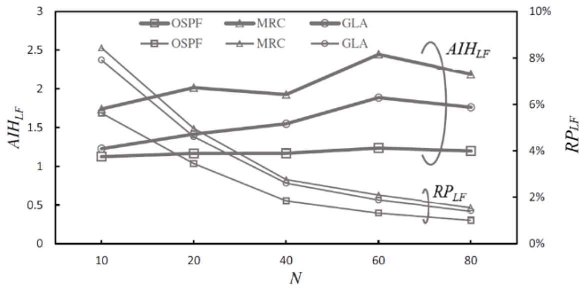


(b) AIHi of BA20

Figure 4. AIH_i of 20-node topology



(a) Waxman topologies



(b) BA topologies

Figure 5. RP_{LF} and AIH_{LF}

In Figure 5, AIH$_{LF}$ roughly increases as N increases. The reason is as follows. The density (which is defined as L/(N(N-1)/2)) of each topology used in the paper becomes smaller for larger value of N. That is, the topology becomes more sparse as N increases. Thus, generally speaking, the number of increased number of hops becomes larger because the number of links which can be used to bypass a link failure becomes smaller.

From Figure 5(a) and Figure 5(b), while RP_{LF} of GLA method is larger than that of OSPF, it is slightly smaller than that of MRC method for each of Waxman and BA topologies. While AIH_{LF} of GLA method is larger than that of OSPF by about 50% at most, it is smaller than that of MRC method by about 50% at most.

5.1.2 One Node Failure Case

Let P_{sd}^{j} be the number of hops of node-pairs (s-d pairs) when node j fails, where we do not consider such a node- pair s, d that node s or d is identical to the failed node j because we assume that the failed node do not generate packets and packets destined for the failed node are removed from the network. Then we define the ratio (say $PR_j;\ 1 \le j \le N$) and the average number (say AIH_j) of the node-pairs (s-d pairs) such that $P_{sd}^{j} - P_{sd} > 0$ as follows:

$$RP_j = \frac{NP_j}{{}_N P_2 - 2(N-1)}$$

where NP_j is the number of node-pairs (s-d pairs) such that $P_{sd}^{j} - P_{sd} > 0$ and $2(\mathrm{N}-1)$ is the number of node-pairs which include the failed node j.

$$AIH_j = \frac{1}{NP_j} \sum_{\substack{\text{all } s-d \text{ pairs} \\ \text{such that} \\ P_{sd}^{j} - P_{sd} > 0}} (P_{sd}^{j} - P_{sd})$$

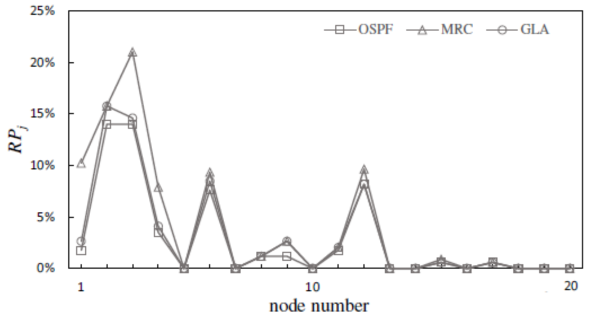
Figure 6 and Figure 7 show RP_j and AIH_j. The points where the values of RP_j and AIH_j are equal to zero for some node numbers mean that NP_j = 0 for node number j. RP_j and AIH_j of GLA method is almost equal to those of OSPF and smaller than those of MRC method.

Figure 8 shows the average values (RP_{NF} and AIH_{NF}) of RP_js and AIH_js over all node numbers which are specifically defined as follows:

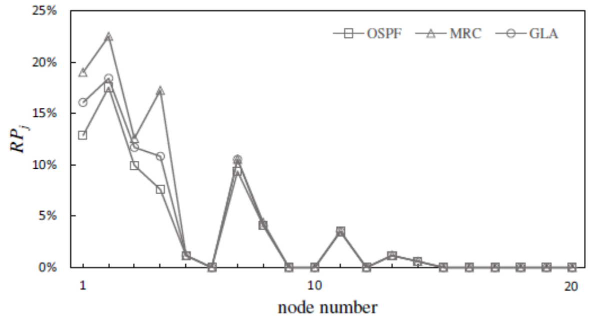
Decreasing of RP_{NF} and increasing of AIH_{NF} for larger values of N are the same as a single link failure case.

From Figure 8, while RP_{NF} of GLA method is close to that of OSPF, it is smaller than that of MRC method. While AIH_{NF} of GLA method is very close to that of OSPF, it is much smaller than that of MRC method.

$$RP_{NF} = \frac{1}{N}\sum_{j=1}^{N} PR_j\,, \qquad AIH_{NF} = \frac{1}{N}\sum_{j=1}^{N} AIH_j$$

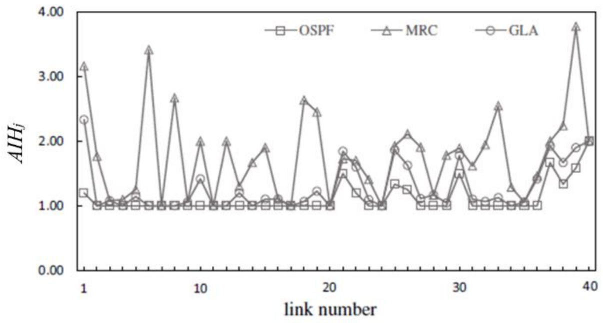


(a) RP_j of Wax20

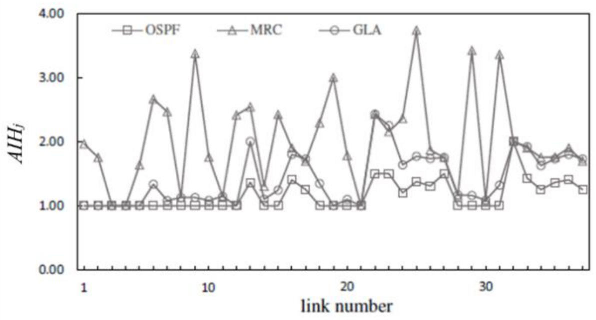


(b) RP_j of BA20

Figure 6. RP_j of 20-node topology

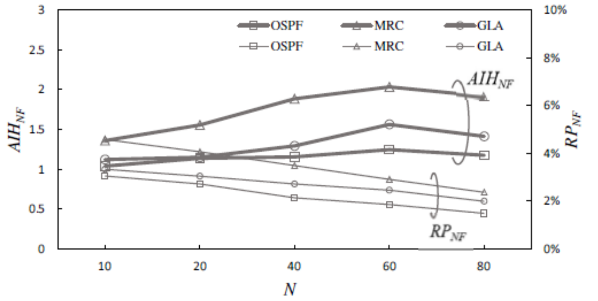


(a) AIH_j of Wax20

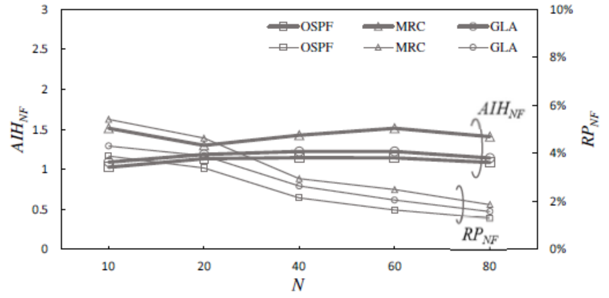


(b) AIH_j of WAX20

Figure 7. AIH_j of 20-node topology



(a) Waxman topologies



(b) BA topologies

Figure 8. RP_{NF} and AIH_{NF}

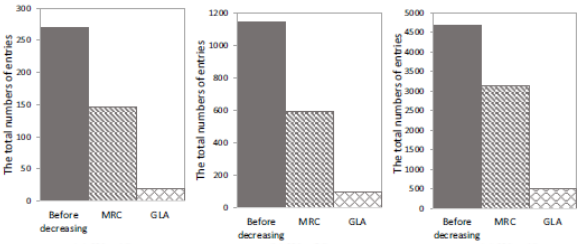
5.2 The Numbers of Entries

We executed the exhaustive search described in Section 4 in order to find unnecessary entries in backup tables for each of Waxman and BA topologies. Figure 9 shows the numbers of entries in backup tables. $\#BT$ means the number of backup tables generated by the algorithm in papers [12-13]. In a topology with N nodes, when the number of entries is not decreased, because each node has $(N-1)$ entries for $(N-1)$ destination nodes in each backup table, it has $(N-1)\times\#BT$ entries, and consequently the number of entries in a topology with N nodes is $(N-1)\times\#BT\times N$. For example, Wax10 has 270 (= $9\times 3\times 10$) entries when we do not decrease the number of entries. On the other hand, the numbers of entries of MRC and GLA methods are decreased to 147 and 20, respectively, and while the ratio of decreasing in MRC method is around 50%, that in GLA method is much higher and is around 10%.

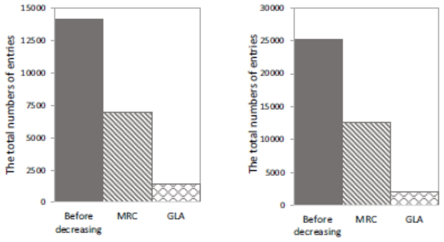
6 Conclusion

We have proposed an IP fast reroute method (GLA method) to bypass a failure component and compared it with a previous method (MRC method) and OSPF. Numerical examples show as follows. In one link failure case, while the average numbers (AIH_{LF}) of increased hops of node-pairs whose numbers of hops increase in GLA method are larger than those in OSPF by about 50% at most, they are smaller than those in MRC method by about 50% at most. In one node failure case, while the average numbers (AIH_{NF}) of increased hops of node-pairs whose numbers of hops increase in GLA method are very close to those in OSPF, they are smaller than those in MRC method. About decreasing in the number of entries in backup tables, while the numbers of entries in MRC method can be decreased by around 50%, those in GLA method can be decreased by around 90%.

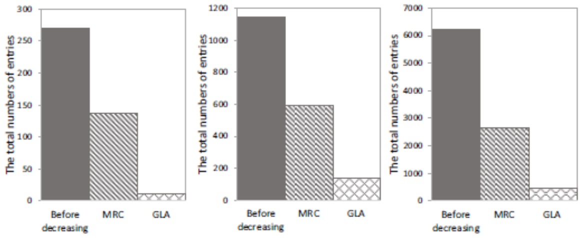
Our future work is to consider IP fast reroute methods to bypass multiple network component failures. One of approaches to bypassing against double component failures is to use MRC method recursively. That is, considering the backbone of each BC (say first level BC) as the original topology, we obtain several BCs (say second level BCs) from the backbone in network design stage. In network operation stage, when a packet encounters a failure, it is rerouted using a first level BC. Then, when the packet encounters another failure, it is rerouted using a second level BC. We will investigate such approach.



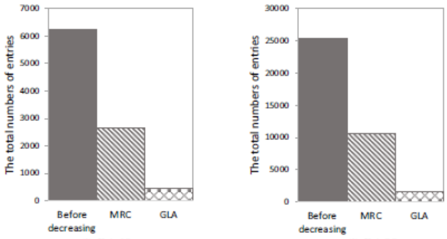
(a) Wax10 #BT = 3 (b) Wax20 #BT = 3 (c) Wax40 #BT = 3



(d) Wax60 #BT = 4 (e) Wax80 #BT = 4



(f) BA10 #BT = 3 (g) BA20 #BT = 3 (h) BA40 #BT = 4



(i) BA60 #BT = 4 (j) BA80 #BT = 4

Figure 9. The total numbers of entries

References

[1] A. Atlas, A. Zinin, Basic Specification for IP Fast Reroute: Loop-free alternates, *IETF RFC 5286*, September, 2008.

[2] D. Thaler, C. Hopps, Multipath Issues in Unicast and Multicast Next-hop Selection, *IETF RFC 2991*, November, 2000.

[3] A. Atlas, U-turn Alternates for IP/LDP Fast-Reroute, *IETF Internet-Draft*, February, 2006.

[4] K. Xi, H. Chao, ESCAP: Efficient SCan for Alternate Paths to Achieve IP Fast Rerouting, *IEEE GLOBECOM 2007*, Washington, DC, USA, 2007, pp. 1860-1865.

[5] S. Nelakuditi, S. Lee, Y. Yu, Z.-L. Zhang, C.-N. Chuah, Fast Local Rerouting for Handling Transient Link Failures, *IEEE/ACM Transactions on Networking*, Vol. 15, No. 2, pp. 359-372, April, 2007.

[6] Z. Zhong, Z. Nelakuditi, Y. Yu, S. Lee, J. Wang, C.-N. Chuah, Failure Inferencing Based Fast Rerouting for Handling Transient Link and Node Failures, *IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, Florida, USA, 2005, pp. 2959-2863.

[7] J. Wang, S. Nelakuditi, IP Fast Reroute with Failure Inferencing, *ACM SIGCOMM 2007*, Kyoto, Japan, 2007, pp. 268-273.

[8] S. Antonakopoulos, Y. Bejerano, P. Koppol, Full Protection Made Easy: The DisPath IP Fast Reroute Scheme, *IEEE/ACM Transactions on Networking*, Vol. 23, No. 4, pp. 1229-1242, August, 2015.

[9] P. Francois, M. Shand, O. Bonaventure, Disruption Free Topology Reconfiguration in OSPF Networks, *IEEE INFOCOM 2007*, Anchorage, Alaska, USA, 2007, pp. 89-97.

[10] S. Bryant, S. Previdi, M. Shand, A Framework for IP and MPLS Fast Reroute Using Not-Via Addresses, *IETF RFC 6981*, August, 2013.

[11] S. Litkowski, A. Bashandy, C. Filsfils, P. Francois, B. Decraene, D. Voyer, Topology Independent Fast Reroute using Segment Routing, *IETF Internet-Draft*, February, 2021.

[12] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, O. Lysne, Fast IP Network Recovery using Multiple Routing Configurations, *IEEE INFOCOM 2006*, Barcelona, Spain, 2006, pp. 1-11.

[13] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing O. Lysne, Multiple Routing Configurations for Fast IP Network Recovery, *IEEE/ACM Transactions on Networking*, Vol. 17, No. 2, pp. 473-486, April, 2009.

Biographies

Yuya Tarutani received an B.E., M.E. and Ph.D. degree in Information Science and Technology from Osaka University, Japan. He is currently an assistant professor in Graduate School of Interdisciplinary Science and Engineering in Health Systems at Okayama University. His research interest includes communication network and network security in IoT network.

Masashi Ishigai received an B.E. and M.E. degree in Engineering from Okayama University in 2018 and 2021. His research interest includes reinforcement learning.

Naoto Numata received an B.E. and M.E. degree in Engineering from Okayama University in 2018 and 2021. His research interest includes reinforcement learning.

Yukinobu Fukushima received the B.E., M.E. and Ph.D. degrees from Osaka University, Japan. He is currently an associate professor of the Graduate School of Natural Science and Technology, Okayama University. His research interest includes knowledge-defined networking and network virtualization.

Tokumi Yokohira received the B.E., M.E. and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan. He has been a professor of Graduate School of Interdisciplinary Science and Engineering in Health Systems of Okayama University. His present research interests include design of virtual networks, technologies to upgrade the speed of the Internet and technologies to increase the fault tolerance of the Internet.