On network traffic concentration and updating interval for proactive recovery method against large-scale network failures

Takuro Horie*, Go Hasegawa†, Satoshi Kamei‡, Masayuki Murata*
*Graduate School of Information Science and Technology, Osaka University
Yamadaoka 1-5, Suita, 565-0871 Japan, E-mail: {t-horie,murata}@ist.osaka-u.ac.jp
†Cybermedia Center, Osaka University
Machikaneyama 1-32, Toyonaka, Osaka, 560-0043 Japan, E-mail: hasegawa@cmc.osaka-u.ac.jp
‡NTT Service Integration Laboratories, NTT Corporation
Midori-cho 3-9-11, Musashino, Tokyo, 560-0043 Japan, E-mail: kamei.satoshi@lab.ntt.co.jp

***Abstract*—Proactive recovery methods from network failures, based on multiple routing configurations, are effective for quick failure recovery, as compared with reactive recovery methods. However, there are two major problems, especially when we consider recovering from large-scale network failures: updating interval for recalculation of routing configurations against network growth, and network traffic concentration on specific nodes and links after recovering failures. In this paper, we first propose a light-weight and distributed algorithm for updating routing configurations when new nodes and links join the network, which does not need overall recalculation. We then evaluate the proactive recovery method for large-scale network failures, with proposed algorithm from the viewpoint of above two problems. Through numerical evaluation results, we find that to maintain the recovery performance, we should recalculate the routing configurations when the network grows by roughly 5-10%. We also present that the degree of network traffic concentration decreases when we employ a hop-by-hop selection mechanism of routing configuration.**

***Index Terms*—Routing, large-scale network failures, proactive failure recovery, network growth**

I. Introduction

Computer networks have already been regarded as an essential infrastructure, much like water and gas utilities. Therefore, recovering from network failures and ensuring network connectivity are an important challenge. In general, network recovery methods are categorized into two types: reactive and proactive [1]. In reactive recovery methods [2, 3], when network nodes detect network failures, they recalculate their routing configurations and propagate them throughout the network to converge the routing. They can accommodate various kinds of network failures flexibly without failure prediction by utilizing dynamic mechanisms in calculating and propagating alternative paths after detecting the failures. One of the main shortcomings of reactive recovery methods is that they require considerable time for routing convergence after the failures since new routing configuration is generally propagated in a hop-by-hop manner [4].

In contrast, in proactive recovery methods [5-7], recovery settings (e.g., routing configurations) by assuming possible failures are calculated and shared by network nodes (routers and switches) beforehand. When a network failure is detected, the recovery method immediately selects one of the settings to correspond to detected failure. Therefore, when the failure is covered by calculated settings, proactive recovery methods do not require routing convergence time after the failure. However, when the failure has not been considered in the calculation, it cannot be recovered completely. So, in proactive methods, we should carefully select the network failures assumed to occur in calculating recovery settings.

In [8], we proposed a recovery method from large-scale network failures based on multiple routing configurations [9]. In the proposed method, we assume various kinds of simultaneous network failures and construct network topologies from the original topology, so that we avoid using failed network equipment. Through numerical evaluation results, we confirmed that our method can improve network reachability while keeping the average path length sufficiently small.

However, we have left two open issues for proposed method, as for existing proactive methods based on multiple routing configurations. One is network traffic concentration after recovering failures. Since the main objective of multiple routing configurations is to maintain the reachability of the network after recovering failures, network traffic may concentrate on specific nodes and links after route changes by applying one of routing configurations. This problem becomes serious especially when we consider recovering from larger-scale network failures since the routing configurations for accommodating multiple-node/link failures would have less available links in the selected network topology.

Another problem is the updating interval of routing configurations against the growth of the network. Ideally, the routing configurations should be recalculated every time when a new node or link is added to the network. Otherwise, the recovery performance would degrade especially at the area around newly-added nodes and links in the network and that is more serious when considering the recover from large-scale network failures. We also want to keep the frequency of recalculation to be small since the recalculation requires the distribution of new routing configurations throughout the network. So, we should clarify the appropriate updating intervals for recalculating routing configurations.

In this paper, we present evaluation results to confirm the performance of the recovery method proposed in [8],

focusing on the network traffic concentration after recovery and updating intervals of routing configurations. Especially, we investigate the effect of the selection mechanisms of routing configurations in routing decision. Also, we propose an algorithm to accommodate newly-joining network nodes and links in routing configurations without overall recalculation, and give guidelines for determining the intervals for recalculating routing configurations.

The remainder of this paper is organized as follows. In Section II, we briefly explain the proposed method in our earlier study [8]. Section III gives detailed descriptions of the problems with network growth and proposes an adding algorithm of new nodes and links to routing configurations without overall recalculation. In Section IV, we discuss the network traffic concentration problem after recovering failure. In Section V, we evaluate the proposed method from the viewpoint of the problems explained in Sections III and IV. Finally, Section VI summarizes the conclusions of the present study and discusses areas of future consideration.

II. Proposed Method

A. Resilient Routing Layers (RRL) [9]

We first introduce Resilient Routing Layers (RRL) [9], which is the basis of our method. As shown in Fig. 1, RRL calculates multiple network topologies and routing tables from the original network topology, which are called as Routing Layers (RLs). In each RL, RRL assumes failure of the network node(s) and isolates them and their adjacent links from the network, so that the isolated nodes are not used for constructing the route between other node pairs. We refer such isolated nodes as *safe nodes*. The set of multiple RLs shared by network nodes is called as Routing Layer Set (RLSet), and each RL in RLSet has some safe nodes in its topology. When failure of network nodes is detected, RRL searches an RL from RLSet in which all failed nodes are isolated, and utilizes such RL for recovering the failure. By this method, any single-node failure can be recovered completely when all network nodes are isolated in at least one RL in RLSet. Moreover, RRL can recover multiple-node failure when all failed nodes are safe in the same RL. However, to the best of our knowledge, there is no previous works, except our paper [8], on the detailed methods for recovering from multiple-node failure based on RRL.

B. RLSet construction for recovering large-scale failures

For constructing effective RLSet to accommodate multiple-node failure, it is important to consider carefully how to choose safe nodes in each RL in RLSet. In [8], we proposed two construction algorithms of RLSet against patterns of simultaneous multiple-node failures.

1) Hub-based algorithm: The hub-based construction algorithm (HUB) assumes failures of high-degree nodes (hub nodes) and their adjacent nodes. It constructs an RLSet that a hub node and as many of its adjacent nodes as possible are isolated in a single RL.

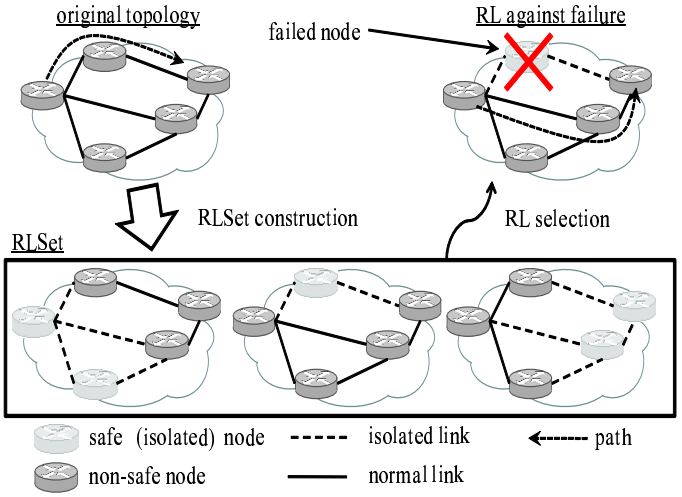


Fig. 1. Resilient Routing Layers (RRL)

2) Attribute-based algorithm: The attribute-based construction algorithm (ATR) pays attention to the attribute of network nodes. This is based on the assumption that the nodes which have same attribute such as location, vendor name, version of node OS, and topological information, fail simultaneously. So, we construct an RLSet so that the nodes with the identical attribute are isolated in a single RL.

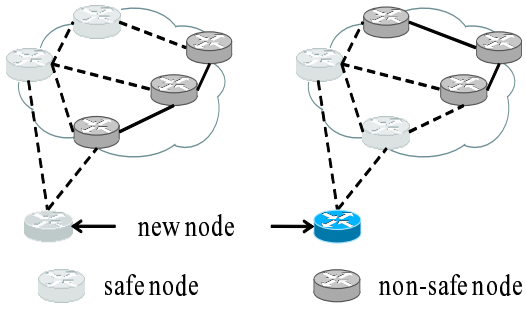
For both algorithms, we consider the overlapping feature, where a single node in the network is isolated in multiple RLs in RLSet. This feature increases the recovering performance of the proposed method since such RLSet would cover larger number of failure patterns. However, since the number of available links in the network decreases when the number of isolated nodes in each RL increases, the path length of each node pair in the network tends to increase. We consider HUB_o and ATR_o which are HUB and ATR with the overlapping feature, respectively.

C. RL selection mechanisms

In [8], we considered the selection mechanisms of RL in routing decision, since they largely affect the recovering performance of the proposed method. We summarize the two algorithms as follows.

1) Static RL selection: In the static RL selection, the source node selects the RL to be used during its packet forwarding. The node tries to find the RL from RLSet in which all failed nodes are isolated. If it is found, the data transmission can be done without any problem. When such RL does not exist, the source node selects the RL which has the largest number of failed nodes as safe. In this case, our method cannot achieve complete reachability of all node pairs.

2) Dynamic RL selection: In the dynamic RL selection, RLs are selected in a hop-by-hop manner. That is, the source and intermediate nodes can select the RLs to be used. The operation of the source node is identical to the static selection. Differently from the static RL selection, the intermediate nodes can change the RL to be used when the next-hop node in the current RL fails. While such on-demand selection generally



(a) An RL in which new node is safe (b) An RL in which new node connects only safe nodes

Fig. 2. Problems in joining new nodes and links

creates a routing loop by repeated changes of RLs, our method avoids it by forcing to use an RL which has larger number of safe nodes than the current RL. The dynamic selection has shortcomings in terms of additional overhead at intermediate nodes and the increase of path length. From the results of [8], we found that this selection greatly improves the network reachability while the degree of increase of path length is limited.

III. Adding new network nodes

Generally, the information networks are always growing up, meaning that the number of nodes and links in the network continues increasing. When new nodes or links are added to the network, the routing configurations for proactive recovery methods should be recalculated. Ideally, for our proposed method, the RLSet should be recalculated and distributed to network nodes every time when a single node or link joins the network. However, the frequent recalculation and distribution of RLSet should be avoided due to the viewpoints of calculation overhead and distribution delay. Therefore, in what follows in this section, we propose an algorithm to add new network nodes and links to the existing RLSet without overall recalculation.

When a new node or link is added to the network, each network node behaves as follows. First, the new node or link is added to the network topologies of all RLs in RLSet. At this phase, the newly-added nodes and links are not isolated in any RL, so the RLSet does not support any failure patterns related to those nodes and links. Therefore, we need to isolate newly-added nodes and links in some RLs in RLSet. When a link is added, the new link is isolated in RL(s) in which the link is connected to at least one safe node. When a node is added, we search RLs in RLSet where the node is connected to at least one non-safe node. Among such RLs, we select one RL with minimum number of isolated nodes and isolate the newly-added node in the selected RL. Then, each network node modifies the routing table for each RL in RLSet as follows. When a link is added, only the route between the nodes connected by the newly-added link is recalculated. When a node is added, the routes departing from and arriving at the node are recalculated. Note that the above calculations can be done at each node in a distributed fashion, so we do not require any information exchanges between any nodes. The pseudo code of the algorithm is shown in Algorithm 1.

Algorithm 1 New network element addition

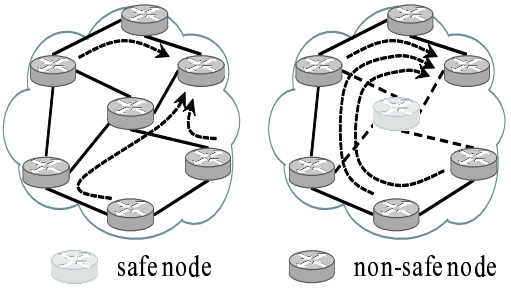
```
1:  Add the new node or link to the network topologies in all
    RLs in RLSet.
2:  if a link is added then
3:      Search RLs in RLSet in which the new link connected
        to at least one safe node.
4:      Isolate the new link in the searched RL(s).
5:  end if
6:  if a node is added then
7:      Search RLs in RLSet where the new node is connected
        to at least one non-safe node.
8:      Isolate the new node in the RL with minimum number
        of isolated nodes among the searched RLs above.
9:  end if
10: for all RLs in RLSet do
11:     if a link is added then
12:         Recalculate the route between nodes connected the
            new link.
13:     end if
14:     if a node is added then
15:         Recalculate the route departing from or arriving at
            the new node.
16:     end if
17: end for
```

However, by utilizing the above algorithm, the recovery performance may degrade in some situations. We explain the problem by using Fig. 2, where we depict the case when a new node connects to one safe node and one non-safe node (Fig. 2(a)), and the case when a new node connects only to two safe nodes (Fig. 2(b)). In the former case, the newly-added node can be isolated without any problems and the recovery performance does not degrade. However, in the latter case, the newly added node cannot be isolated and there is no route to/from the node in this RL. So, when such RL is selected for failure recovery, the reachability of the network degrades slightly.

To solving this problem, we need the overall recalculation of the RLSet to maintain the recovery performance. In Section V, we evaluate the performance degradation caused by this problem and discuss the appropriate interval for overall recalculation of RLSet against network growth.

IV. Network traffic concentration

When a network failure occurs and one RL from RLSet is utilized for recovering the failure, routes between nodes in the network changes largely since some nodes and links are isolated in the RL. Here, while the network reachability can be maintained, the following two problems occur. One is the increase of traffic amount in the network since the decrease of available nodes and links increases the path length between the nodes. Another problem is the bias in the route selection, meaning the concentration of the traffic to specific nodes and links, due to the decrease of available links in the network. These problems cannot be ignored especially when



(a) Traffic routes on the original network

(b) Traffic routes on the RL which isolates a center node

Fig. 3. Network traffic concentration after recovering failure

we consider the recovery from large-scale failures, since RLs for accommodating such serious failures have more isolated nodes, meaning less available links.

The example of the above problem is illustrated in Fig. 3. Fig. 3(a) shows the traffic routes on the original network. Fig. 3(b) shows the traffic routes after the failure of the center node occurs and the RL which isolates the failed node is utilized for recovery. Obviously, the network traffic concentrates the specific links after the failure recovery. In the next section, we evaluate this problem and show that a hop-by-hop RL selection mechanism [8] decreases the degree of the traffic concentration.

V. Evaluation results

A. Evaluation settings

In this paper, we investigate the performance of the proposed method explained in Section II from the following two viewpoints: the concentration of network traffic after recovering failure and updating intervals of RLSet against network growth. For the evaluation of traffic concentration, we utilize the AS-level network topology, constructed only by ASes administrated by Japan Network Information Center (JPNIC), which is obtained from CAIDA [10]. The topology consists of 259 nodes and 1162 links, meaning that the average degree of the network nodes is 4.4. Note that we exclude the nodes which are connected like the chain-topology since they have no alternative route. For performance evaluation with network growth, we utilize the network topology based on BA model [11]. We start with the network topology with 259 nodes and 1030 links, and add a new node with four links to the network in a one-by-one manner until the network has 359 nodes and 1430 links.

For evaluating network traffic concentration, we evaluate the traffic ratio, which is defined as the ratio of the network traffic amount on a link after the recovering failure to that before the failure occurs. The amount of network traffic of a link is defined as the number of node pairs whose route passes through the link. To investigate the performance against network growth, we evaluate the network reachability that represents the ratio of reachable node pairs after recovering failure, to all node pairs in the network except the failed nodes. We then also evaluate the average path length between all reachable node pairs.

We use the network multiple simultaneous failures the following four types:

F_RND is the failure of randomly selected nodes.
F_ADJ is the failure of directly interconnected nodes.
F_ATR is the failure of the nodes with identical attribute.
F_LNK is the failure of the links which are among directly interconnected nodes.

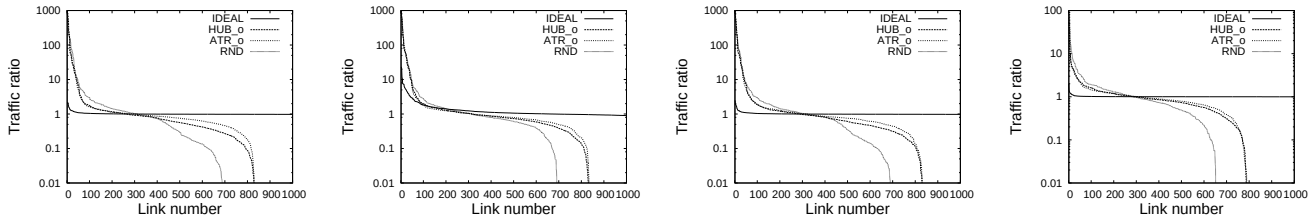
B. Traffic ratio after recovery

Fig. 4 illustrates the distribution of the traffic ratio of each link in the network with static RL selection against two node failures. In this figure, we plot the results of IDEAL, HUB_o, ATR_o, and RND. IDEAL represents the result of the ideal case where we recalculate the routing tables after failure detection. RND means the case when we randomly select nodes to be isolated for each RL, with overlapping feature (Subsection II-B). The number of RLs in RLSets is around 250 for all construction algorithms except RND, and 2000 for RND. From Fig. 4, we can observe a small part of links in the network suffers from the great increase of traffic after recovering failure. This is caused by the network traffic concentration described in Section IV. We can also find that RND and HUB_o have larger values of traffic ratio on specific links, compared with other algorithms. This is because these algorithms isolate more nodes and their adjacent links in each RL, which decrease the number of available links in the selected RL's network topology. The results of IDEAL give the smallest increase of traffic ratio since it recalculates the routing configurations with all non-failed links in the network. As shown in Fig. 4(d), we can observe that the results against F_LNK are smaller than other failures. This is because F_LNK fails smaller number of links compared with other three failure patterns.

Fig. 5, which shows dynamic RL selection, illustrates the results corresponding to Fig. 4. We can observe that the dynamic RL selection gives the smaller degree of the traffic ratio increase compared with the static RL selection. This is because by utilizing hop-by-hop RL selection mechanism, the routes of node pairs would be distributed throughout the network.

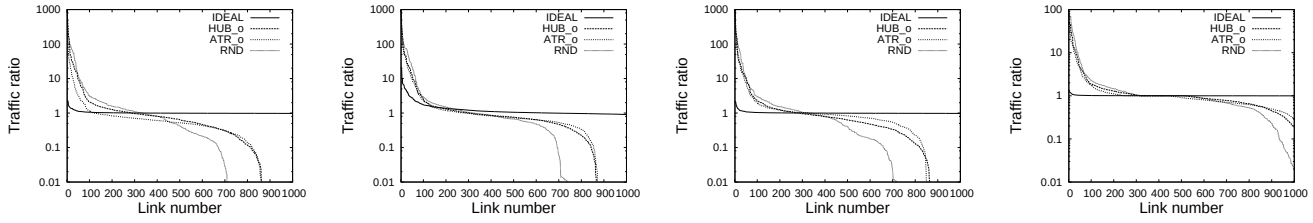
C. Performance with network growth

Fig. 6 represents the network reachability as a function of the number of added nodes with static RL selection against two node failures when we use HUB_o and ATR_o, where they have around 20 RLs in each RLSet. Note that the other construction algorithms have the similar performance. The label *plain* is the case when we do not recalculate the RLSet, the *recal* is the case when we recalculate every time a new node joins the network. From the figure, we can observe that the recalculation of RLSet does not affect the network reachability. This is because the static RL selection has few RLs which isolate all failed nodes, so the problem



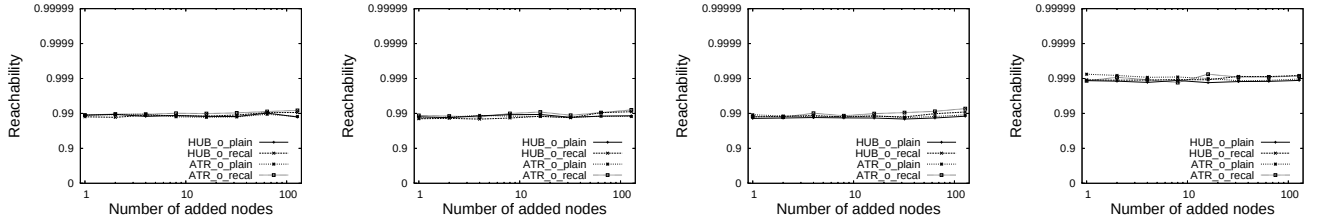
(a) F_RND (b) F_ADJ (c) F_ATR (d) F_LNK

Fig. 4. Traffic ratio with static RL selection against two node failures



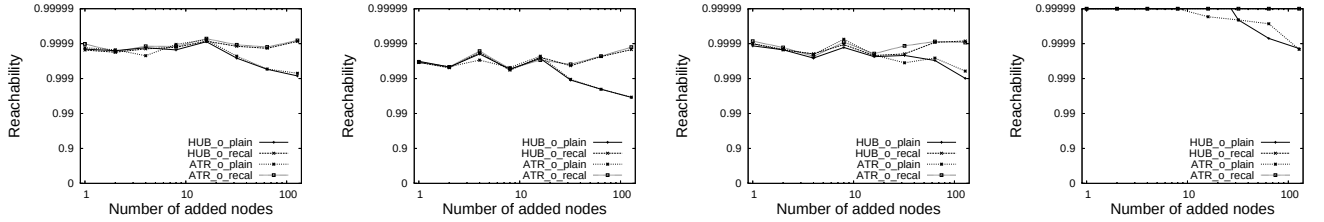
(a) F_RND (b) F_ADJ (c) F_ATR (d) F_LNK

Fig. 5. Traffic ratio with dynamic RL selection against two node failures



(a) F_RND (b) F_ADJ (c) F_ATR (d) F_LNK

Fig. 6. Network reachability with static RL selection against two node failures



(a) F_RND (b) F_ADJ (c) F_ATR (d) F_LNK

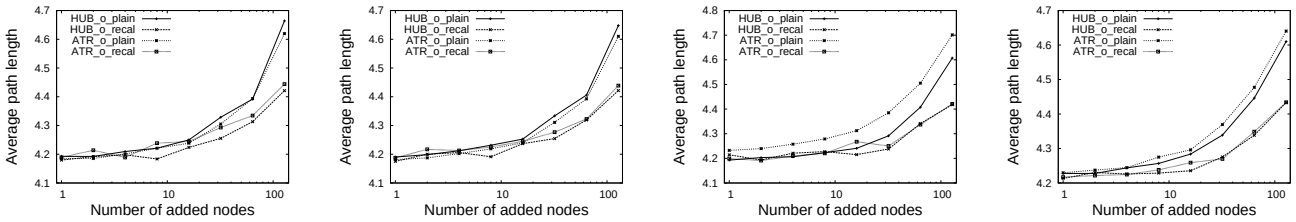
Fig. 7. Network reachability with dynamic RL selection against two node failures

described in Section III does not differentiate the performance of recalculation.

Fig. 7 shows the results corresponding to Fig. 6 when we use the dynamic RL selection. We can find that the network reachability without recalculation would decrease when more than around ten nodes join the network, against any failure pattern. This means that with dynamic RL selection, we should recalculate the RLSet when ten nodes join the network, to maintain the performance of the proposed method.

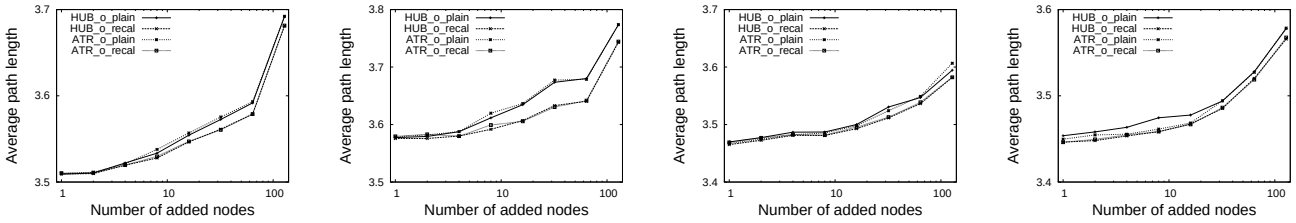
Fig. 8 shows the average path length as a function of the number of added nodes with static RL selection against two node failures. We can see from this figure that the average path lengths of recalculated RLSets are smaller than that of non-recalculated RLSets, especially when the number of added nodes is larger than ten. This means that the recalculation of RLSets affects path lengths significantly. Moreover, we can observe that the average path length increases as the number of added nodes increases in both cases. This is caused by enlargement of the topology diameter by network growth.

Fig. 9 shows the corresponding results to Fig. 8 when we use the dynamic RL selection. In contrast to the static RL selection, the effect of recalculation is negligible. This is because by using dynamic RL selection, the routes for node pairs become quite different from the shortest path.



(a) F_RND (b) F_ADJ (c) F_ATR (d) F_LNK

Fig. 8. Average path length with static RL selection against two node failures



(a) F_RND (b) F_ADJ (c) F_ATR (d) F_LNK

Fig. 9. Average path length with dynamic RL selection against two node failures

Furthermore, we find that the path lengths with the dynamic RL selection are shorter than that with the static RL selection. This is because that when there is no RL in which all failed nodes are safe, the dynamic RL selection utilizes the RL that has many available links for packet forwarding in spite that the static RL selection utilizes the RL with few available links.

From these results, we conclude that we should recalculate the RLSet when approximately the 5-10% nodes of all network nodes join the network to maintain both reachability and path length.

VI. Conclusion

In this paper, we investigated the two major problems of the proactive recovery method from large-scale network failures, which is based on multiple routing configurations. One is network traffic concentration to specific nodes and links after recovering failure. Another problem is related to network growth. For latter problems, we proposed a new algorithm to add nodes and links to routing configurations without overall recalculation. Through numerical evaluations for the two problems, we found that the dynamic selecting mechanism of routing configurations can decrease the degree of traffic concentration after the recovery. We also found that the recalculation of routing configurations every network growth by 5-10% achieves better reachability and shorter path length.

For future work, we will further evaluate proposed method with different networks and evaluation conditions. We also plan to consider the application of the proposed method to routing in overlay networks.

Acknowledgment

This work was partly supported by "Special Coordination Funds for Promoting Science and Technology: *Yuragi Project*," Grant-in-Aid for Scientific Research on Priority Areas 18049050 in Japan, and the National Institute of Information and Communications Technology of Japan (NICT).

References

[1] S. Rai, B. Mukherjee, and O. Deshpande, "IP resilience within an autonomous system: Current approaches, challenges, and future directions," *IEEE Communications Magazine*, vol. 43, pp. 142–149, Oct. 2005.

[2] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 756–767, May 2002.

[3] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, Oct. 2001.

[4] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," in *Proceedings of IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 3, Jun. 2001, pp. 293–306.

[5] O. Klopfenstein, "Robust pre-provisioning of local protection resources in MPLS networks," in *6th International Workshop on Design and Reliable Communication Networks, 2007 (DRCN 2007)*, Oct. 2007, pp. 1–7.

[6] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang, and C.-N. Chuah, "Proactive vs reactive approaches to failure resilient routing," in *Proceedings of the IEEE INFOCOM 2004, 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, Mar. 2004, pp. 176–186.

[7] D. Wang and G. Li, "Efficient distributed bandwidth management for MPLS fast reroute," in *Proceedings of the IEEE/ACM, Transactions on Networking*, vol. 16, Apr. 2008, pp. 486–495.

[8] T. Horie, G. Hasegawa, S. Kamei, and M. Murata, "A new method of proactive recovery mechanism for large-scale network failures," in *Proceedings of the IEEE 23rd international conference on advanced information networking and applications (AINA-09)*, Bradford, May 2009.

[9] A. Hansen, A. Kvalbein, T. Čičić, and S. Gjessing, "Resilient routing layers for network disaster planning," *Lecture notes in computer science*, vol. 3421, pp. 1097–1105, Apr. 2005.

[10] The CAIDA Web Site, available at http://www.caida.org/home/.

[11] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.