Master's Thesis

Title

Proactive recovery from multiple failures utilizing overlay networking technique

Supervisor
Professor Hirotaka Nakano

Author
Takuro Horie

February 15th, 2010

Department of Information Networking
Graduate School of Information Science and Technology
Osaka University

Master's Thesis

Proactive recovery from multiple failures utilizing overlay networking technique

Takuro Horie

Abstract

Recent network applications require high network availability for maintaining continuous connectivity. However, most of existing routing protocols in the current Internet have problems in recovering from multiple simultaneous failures, where they require a long time for routing convergence after detecting such failures since the network equipment has to detect the failures, recalculate routing configurations, and propagate the configurations throughout the network. For example, BGP requires considerable time, which is from a few minutes to several days, to converge routing configurations, especially for large-scale failures or certain types of network topologies. Essentially, the routing convergence time in BGP has no theoretical upper bound, and there are many situations in which the routing convergence time increases significantly, as in the count-to-infinity problem. Accordingly, various methods to improve the routing convergence time in BGP have been proposed, however, most of them require modifications to BGP itself. Therefore, they cannot be applied to the current Internet environment in short time due to difficulties in interoperability and the need for the standardization process.

In this thesis, the author proposes a proactive recovery method against multiple simultaneous failures for large-scale packet switching networks. The proposed method exploits the overlay networking technique to realize fast and effective recovery from failures, since it is implemented at application layer. Specifically, it constructs multiple logical network topologies assuming various failure patterns in advance. When a failure is detected, the proposed method can immediately recover from the failure by utilizing the appropriate topology to the failure, without waiting routing convergence in the underlay network. Furthermore, the proposed method considers the correlation among overlay links in terms of utilizing underlay link to construct the effective topologies for recovering from multiple simultaneous failures. The proposed method also could construct the topologies for failure recovery at each overlay node in a distributed fashion.

Through numerical evaluations in terms of the overlay reachability and the average path length after failure recovery, the author shows that the proposed method improves the overlay network reachability from 51 % to 97 %, while keeping the path length to be enough small, when 25 % underlay links are down simultaneously.

Keywords

Overlay network
Routing
Multiple failures
Proactive failure recovery

Contents

List of Figures

List of Tables

1 Introduction

Recent network applications require high network availability for maintaining continuous connectivity. However, most of existing routing protocols in the current Internet have problems in recovering from multiple simultaneous failures, where they require a long time for routing convergence after detecting such failures since the network equipment has to detect the failures, re-calculate routing configurations, and propagate the configurations throughout the network. For example, Border Gateway Protocol (BGP) [1], which operates inter-Autonomous System (inter-AS) routing in the current Internet, requires considerable time (from a few minutes to several days) to converge routing configurations after detecting network failures, especially for large-scale failures or certain types of network topologies [2-6]. Essentially, routing convergence time in BGP has no theoretical upper bound, and there are many situations in which the routing convergence time increases significantly, as in the count-to-infinity problem [7]. Various methods to improve the routing convergence time in BGP have been proposed [8-10]. However, most of them require modifications to BGP itself, which means that they require standardization processes. Consequently, such modifications cannot be deployed to the current Internet in the near future.

Therefore, the overlay networking technique has been proposed, which can deploy original protocols immediately since it does not require standardization processes. In this thesis, overlay networks are defined as upper-layer networks built on the lower-layer packet switching networks such as IP network. Figure 1 illustrates the definition of underlay and overlay networks in this thesis. These overlay networks provide special-purpose application services such as file sharing, grids, IP-VPN services, and Content Delivery/Distribution Networks (CDNs) [11-14]. In overlay networks, the endhosts and servers that run application programs become overlay nodes that form the upper-layer logical network with logical links among the overlay nodes, and the overlay nodes control the application traffic to satisfy their requirements and policies.

Furthermore, *overlay routing*, which is the overlay networking technique specialized to the traffic routing, has been proposed [15-18]. Since the overlay routing controls application traffic in application layer, the overlay-routed traffic may traverse different routes from BGP routing, and moreover, the links that are limited the usage by BGP can be utilized any routes by the overlay routing. The reason why BGP limits some links is that Internet Service Providers (ISPs) consider monetary cost structures and utilization policies. In IP network, ISPs generally have many links

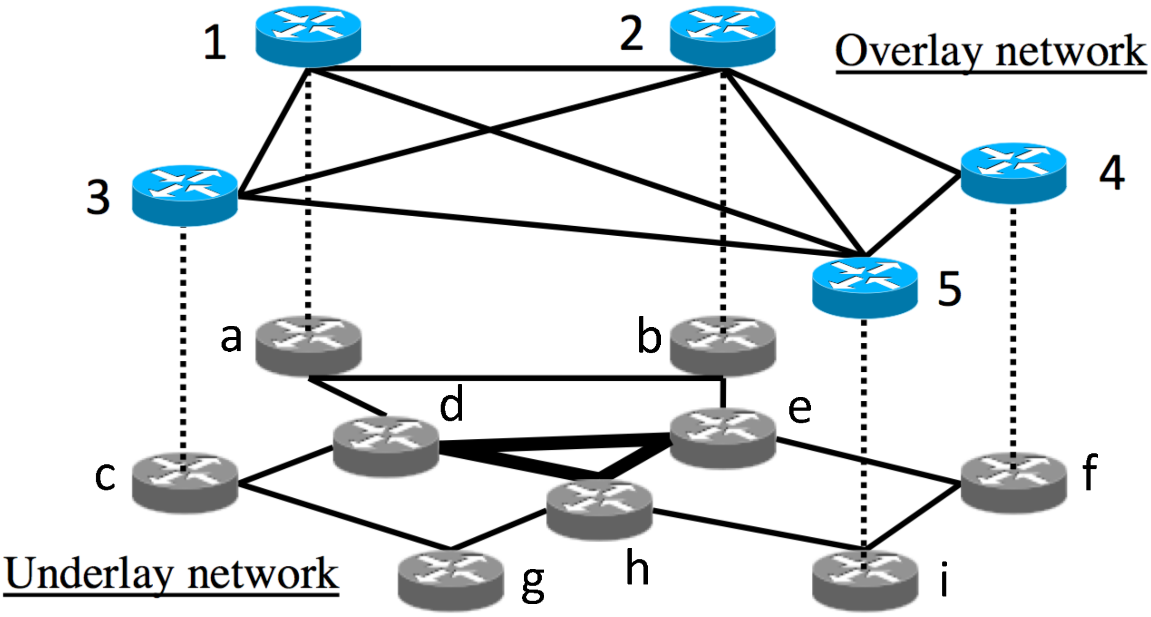


Figure 1: Definition of underlay and overlay networks

interconnecting with other ISPs based on various types of monetary cost structures and utilization policies, such as peering and transit relationships. For example, each peering link can be utilized only by two ASes which are interconnected by the peering link since the maintenance cost of such links are paid by the ASes interconnected by the links [19]. Therefore, ISPs make routing decisions based on the cost structure against their neighboring ISPs [20-23]. On the other hand, since the overlay routing can control application traffic regardless of ISPs' routing policies, the overlay-routed traffic may traverse different routes in the network that the ISPs do not assume in their under-layer routing configurations. One of advantages in overlay routing, which is caused by this mismatch in routing policies, is that the overlay routing can improve the user-perceived network performance such as end-to-end delay and available bandwidth [24-26].

One problem in overlay routing is that a single underlay network failure would cause multiple simultaneous failures in overlay networks. Figure 2 shows an example of such failures. In the figure, the links and paths in the underlay network are denoted as *underlay links* and *underlay paths*, respectively, and the links in the overlay networks are defined as *overlay links*. Each overlay link between two overlay nodes corresponds to an underlay path, which consists of one or more un-

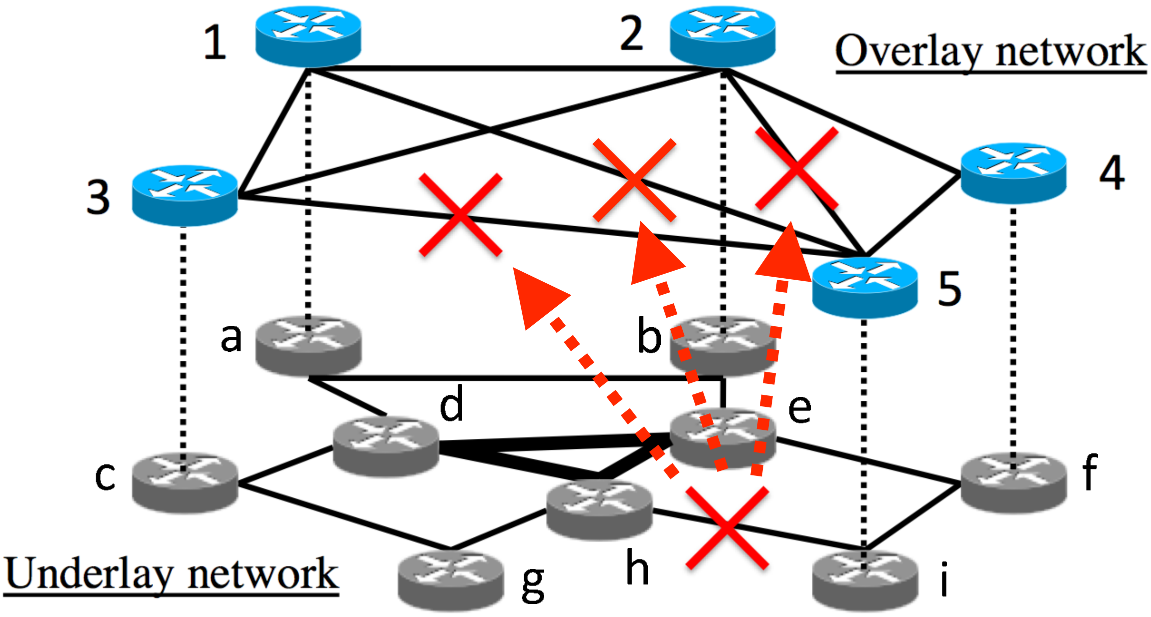


Figure 2: Multiple simultaneous failures in the overlay network

derlay links. Note that it is likely to occur that multiple overlay links share some underlay links in their underlay path. When such underlay links fail, multiple overlay links go down simultaneously. In Figure 2, the overlay links 1-5, 2-5, and 3-5 overlap the underlay link h-i. Therefore, when the underlay link h-i goes down, the overlay links 1-5, 2-5, and 3-5 lose the connectivity simultaneously. Generally, since the overlay network cannot control the underlay routing, the overlapped utilization of underlay links, as described in Figure 2, cannot be explicitly avoided. Therefore, the overlay networks should have an effective recovery method against multiple simultaneous failures.

In general, network recovery methods are categorized into two types, reactive and proactive [27]. In reactive recovery methods, when network nodes detect network failures, they calculate new routing configurations and propagate them throughout the network to converge the routing [15, 28]. The nodes can accommodate various kinds of network failures flexibly without failure prediction. One of the main shortcomings of reactive recovery methods is that the considerable time is required for routing convergence after the failures, since new routing configurations generally propagated in a hop-by-hop manner. In contrast, proactive recovery methods pre-calculate recovery settings (e.g., routing configurations) by assuming possible failures and distribute the settings throughout the network in advance [29-31]. Then, when a network failure is detected, the recovery method immediately selects one of the pre-calculated settings according to the detected failure. When the failure is covered by the pre-calculated settings, proactive recovery does not require routing convergence time after the failure. However, when the failure has not been considered in the pre-calculation, the proactive recovery method cannot completely recover from the failure. Therefore, in the proactive method, we should carefully select the network failures assumed to occur in pre-calculating the recovery settings.

In this thesis, the author proposes a proactive recovery method against multiple simultaneous failures for large-scale packet switching networks. The proposed method exploits the overlay networking technique to realize fast and effective recovery from failures. Specifically, it is based on Resilient Routing Layers (RRL) [32] that constructs multiple logical network topologies assuming various failure patterns in advance. When a failure is detected, the proposed method can immediately recover from the failure by utilizing the appropriate topology to the failure, without waiting routing convergence in the underlay network. Furthermore, the proposed method considers the correlation among overlay links in terms of utilizing underlay links to construct the effective topologies for recovering from multiple simultaneous failures in the overlay network. Another objective for the proposed method is that it should be applied to the existing overlay networks by simple mechanism, for improving the reliability of the existing overlay networks.

The effectiveness of the proposed method is demonstrated by numerical evaluation results using an actual router-level network topology and topologies generated by BRITE [33]. The author exhibits that the proposed method improves overlay network reachability significantly in case of simultaneous multiple underlay link failures. Furthermore, it is shown that the proposed method can sustain the path length after recovery enough small. In addition, the author proposes the partial re-calculation algorithms of the topologies for failure recovery against the overlay network changes.

The remainder of this thesis is organized as follows. Section 2 gives a brief explanation of RRL, which is the basis of the proposed method. In Section 3, the design issues and detailed algorithms of the proposed method are presented. The author confirms the effectiveness of the proposed method using extensive numerical examples in Section 4. Finally, Section 5 summarizes the conclusions of this thesis and discusses areas of future consideration.

2 Resilient Routing Layers (RRL)

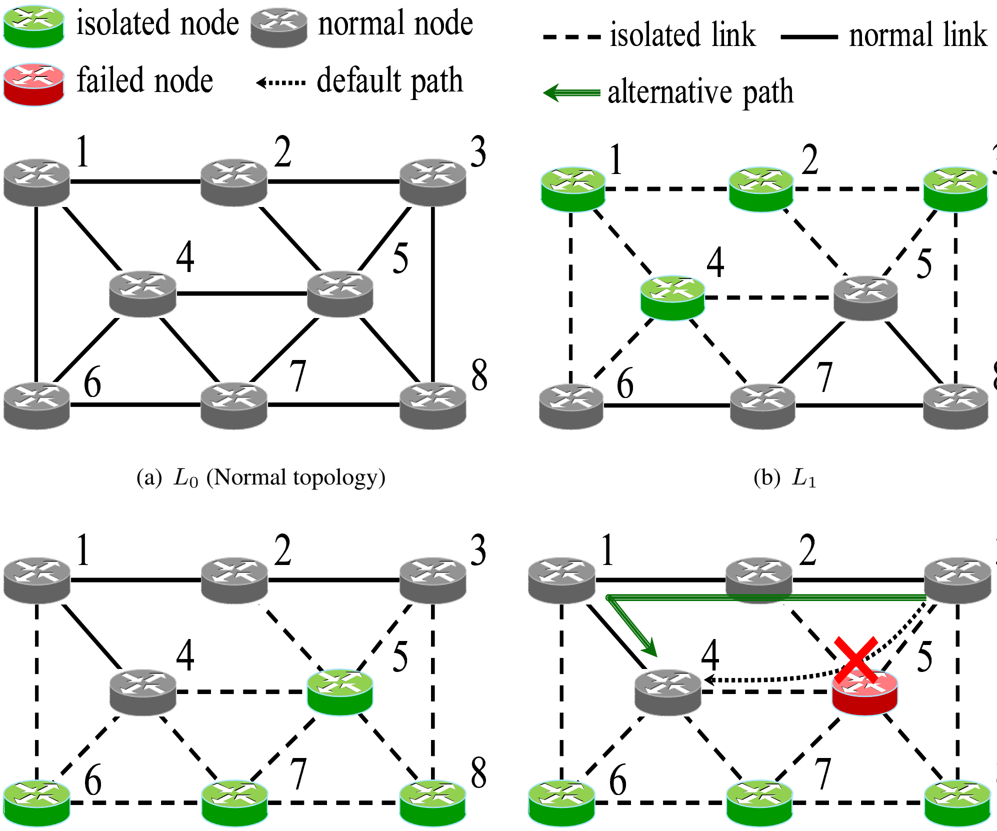
In this section, the author explains Resilient Routing Layers (RRL) [32] that is the basis of the proposed method. In Subsection 2.1, an overview of RRL is explained. The problem of RRL against multiple simultaneous failures is described in Subsection 2.2, and the advantages to adapt RRL for overlay networks are presented in Subsection 2.3.

2.1 Overview

RRL pre-calculates multiple network topologies and routing configurations, which are called Routing Layers (RLs), from the original network topology. In each RL, RRL assumes that a failure of the network node(s) occur, and configures the network topology to recover the failure without degrading the reachability of other parts of the network. All network nodes share the calculated RLs and select an identical RL when network failures occur. RRL utilizes the original network topology as long as no failure occurs.

Figure 3 illustrates an example of the application of RRL to a sample network topology. In this thesis, the node that is assumed to be down in each RL is denoted as an *isolated node*, and the node that is not assumed to be down in each RL is denoted as a *normal node*. The calculated RLs are defined a *Routing Layer Set* (RLSet). With the exception of the original network topology, each RL has at least one isolated node. The weight of the link connected to the isolated node is set to the maximum value so that the isolated node is prevented from using as a route among other nodes. That is, the links connecting to the isolated node are used only when the isolated node is either the source or destination node. Such links are denoted as *isolated links* and the rest links are denoted as *normal links*. When a node detects its adjacent node failure, the node selects an RL in which the failed node is isolated. Once the node selects an appropriate RL from the RLSet, all transmitted packets can avoid the failure.

In Figure 3, the paths among normal nodes only use normal links, as shown solid lines in each RL. Figure 3(a) represents the original network topology L_0. L_0 is utilized while no failure is detected in the network. In L_1 in Figure 3(b), nodes 1, 2, 3, and 4 are isolated nodes. In L_2 in Figure 3(c), nodes 5, 6, 7, and 8 are isolated. That is, every network node is isolated in at least one RL in RLSet. Note that the weight of the isolated links, as shown dashed lines in Figure 3 is set to the maximum value, since they are isolated links that connect to isolated nodes. Using this



(a) L_0 (Normal topology)

(b) L_1

(c) L_2

(d) RLSet accommodating of L_2

Figure 3: Resilient Routing Layers

figure, let us consider a data transmission from node 3 to node 4. When there is no failure in the network, L_0 is utilized and the route becomes 3-5-4, assuming that each RL utilizes the route by Dijkstra's shortest path algorithm. When node 5 is down, the route from node 3 to node 4 becomes unavailable since it includes the failed node. In this case, L_2 is utilized since node 5 is isolated in L_2. Then the route from node 3 to node 4 becomes 3-2-1-4, as shown in Figure 3(d).

2.2 Accommodation of multiple network failures

RRL can recover from a single node failure completely, meaning that it can keep the reachability of all nodes except the failed node when all nodes are isolated in any RL. This is because each node in the network is isolated in at least one RL in the RLSet. In [32], Hansen et al. show the following evaluation results: even when the network has thousands of nodes, RRL requires as few as tens of RLs to keep all network nodes isolated in at least one RL. In addition, when multiple nodes fail simultaneously, the failures can be recovered by utilizing the RL that isolates all failed nodes simultaneously. In other words, RRL has a potential ability of recovering from multiple simultaneous failures.

However, recovering from multiple simultaneous failures requires almost infinite number of RLs to cover all failure patterns. Since the number of RLs is limited by the memory space of the network nodes and/or the required bit size of packet header to identify RLs, it is impossible that the RLSet covers all failure patterns. Isolating a lot of nodes in an identical RL can decrease the number of RLs that cover multiple simultaneous failures. On the other hand, as the number of isolated nodes in an RL increases, the number of available links in the RL decreases since the number of isolated links increases. This results in the increase of the path length (hop count) among nodes in the RL. Therefore, to realize high recovery performance against multiple simultaneous failures, we should carefully configure the number of RLs in RLSet, the number of isolated nodes in each RL, as well as the selection of nodes as isolated in each RL. However, to the best of our knowledge, no other research results have been reported on RRL-based proactive recovery methods for multiple simultaneous failures.

2.3 RRL implementation for overlay networks

In the network with RRL, all nodes share the same RLSet. When a node detects a failure, the node searches the appropriate RL against the failure from the RLSet, and forward the packet according to the configurations in selected RL. The identifier of selected RL is informed to the next-hop node by putting the identifier on the packet header. Therefore, the node can correctly forward packet without waiting routing convergence in the network. In [32, 34], the authors noted that RRL could be implemented at various layers. In [34], they show an example of RRL implementation in a Multi-Protocol Label Switching (MPLS) network. In an IP network, RRL can be implemented by utilizing unused bits of the IP packet header to designate the identifier of the currently used RL. One of the significant shortcomings in implementation at MPLS and IP layers is the requirement standardization process. The other problem is that RRL must be implemented for all network nodes (MPLS switches or IP routers) in the network.

Therefore, the author accommodates the overlay networking techniques to RRL to solve these problems. When RRL is implemented on overlay networks, we receive both benefit of the reactive and the proactive recoveries. This is because the routing protocols on the underlay network, such as BGP, generally employ the reactive recovery, and RRL implemented on the overlay network employs the proactive recovery. Furthermore, since RRL implemented on the overlay network can route the traffic with liberating the limitation on IP routing by ISPs, for example the peering links are limited its usage as only two ASes interconnected by themselves, the recovery performance with RRL may much improve.

3 Proposed method

In this section, the author proposes a proactive recovery method based on RRL. The proposed method can recover from multiple simultaneous failures by utilizing overlay networking technique. In Subsection 3.1, preconditions and notations on the proposed method are explained. In Subsection 3.2, the author proposes the algorithm to construct RLSet for failure recovery. Subsection 3.3 introduces two selection methods of appropriate RL to detected failures from RLSet. In Subsection 3.4, the author describes how to accommodate network topology changes such as node joining or deletion, at both of overlay network and underlay network.

3.1 Preconditions and notations

In the proposed method, the author assumes that an overlay network, which consists of overlay nodes and overlay links, is given in advance and RLSet is constructed through the algorithms in Subsection 3.2. Note that it is important to consider the appropriate setting (selecting of overlay nodes and determining overlay network topology) for the initial overlay network for effective recovery in underlay networks. Although this is one of our future work, the algorithms in this section can be applied to existing overlay networks to increase their reliability and robustness against failures.

The author also assumes the relationships between underlay network and overlay network is determined as in Figure 1. Furthermore, it is assumed that all overlay nodes know the underlay network topology and underlay routes between all overlay nodes. Table 1 summarizes the notations utilized for the explanation of the proposed method in the following subsections.

3.2 RLSet construction algorithm

The proposed method constructs an RLSet that accommodates multiple simultaneous failures in overlay network while keeping the number of RLs small, by utilizing the information of the underlay network topology. Specifically, each overlay node first constructs multiple RLs as initial RLSet. The constructed RLs are then aggregated among neighboring overlay nodes with some merging procedures. Finally, all RLs are gathered at one overlay node and they construct an RLSet. Note that the proposed method can decrease the computation time for RLSet construction due to its distributed construction algorithm, and improve the recovery performance since the

Table 1: Notations for proposed method

G_u	Underlay network
G_o	Overlay network
$V(G)$	Set of nodes in G, $V(G_o) \subseteq V(G_u)$
$E(G)$	Set of links in G
i, j, k, l, m, n	Identifier of nodes, $1 \leq i, j, k, l, m, n \leq \vert V(G_u) \vert$
v_i	i-th underlay node in $V(G_u)$
$s(v_i)$	State of v_i, $s(v_i) \in \{0, 1\}$ (0: merging, 1: waiting)
u_{ij}	Underlay link between $v_i, v_j \in V(G_u)$, $u_{ij} \in E(G_u)$
$d(v_i)$	Degree of v_i, $d(v_i) := \vert \{u_{ij} \mid \forall j, u_{ij} \in E(G_u)\} \vert$
o_{ij}	Overlay link between $v_i, v_j \in V(G_o)$, $o_{ij} \in E(G_o)$
$\mathcal{L}(v_i)$	Set of RLs held by v_i
s, t	Identifier of RLs and function of overlay link's weight, $1 \leq s, t \leq \vert \mathcal{L}(v_i) \vert$
L^i_s	s-th RL in $\mathcal{L}(v_i)$
w^i_s	Function of overlay link's weight with L^i_s The weight of o_{jk} is a with $L^i_s \Leftrightarrow w^i_s(o_{jk}) = a$
L_0	RL correspond to the original overlay network $L_0 := (G_u, w_0)\ s.t.\ \forall u_{ij} \in E(G_u), w_0(u_{ij}) = 1$
$w^i_{s,t}$	Function of overlay link's weight with $L^i_{s,t}$ that is merged L^i_s and L^i_t $w^i_{s,t}(o_{jk}) = a \Leftrightarrow \max(w^i_s(o_{jk}), w^i_t(o_{ij})) = a$
$r_{jk}(L^i_s)$	Set of underlay link on the path between $v_j, v_k \in V(G_o)$ with L^i_s $r_{jk}(L^i_s) := \{u_{lm} \mid u_{lm}$ on the path from v_j to v_k with $L^i_s\}$

number of failure patterns that is covered in each RL increases.

3.2.1 Initial RLSet construction

First, each overlay node constructs the initial set of RLs that accommodates failures around the node. The detailed method is shown as follows.

The overlay node v_i constructs RLs each of which isolates one overlay link o_{ij} connecting to v_i. When there exists overlay links share some underlay links which construct o_{ij}, those overlay links are isolated in a single RL. This is because such overlay links are likely to fail simultaneously. When the resulting RL isolating all overlapping overlay links is not a connected graph, the author utilizes multiple RLs to isolate those overlay links so that the generated RLs keep the connectivity. On the other hand, when an underlay link is shared by all overlay links connected to an overlay node, the RL that isolates the overlay links becomes unconnected to the node. In this case, the author isolates the overlay node itself, instead of using such unconnected RL.

Here, a set of RLs at an overlay node is denoted as an *RLSet* and it is denoted as $\mathcal{L}(v_i)$ through the construction algorithm in Subsection 3.2. Algorithm 1 describes detailed construction algorithm for initial RLSet $\mathcal{L}(v_i)$ at overlay node v_i.

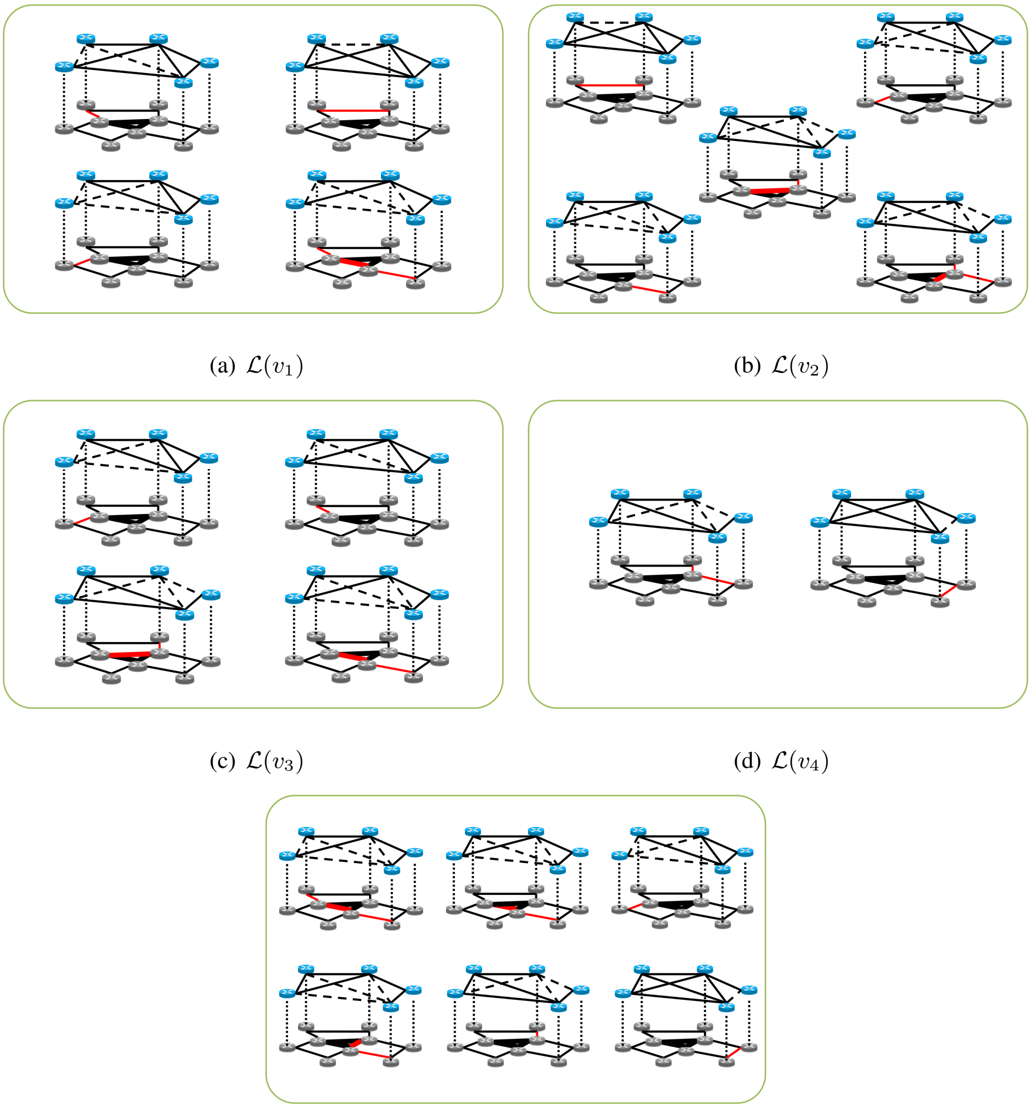
Figure 4 illustrates examples of the initial RLSets corresponding to the overlay network depicted in Figure 1. The red link in each RL represents the underlay link composing the overlay link that is isolated in the RL. The dotted links in the overlay network represents the overlay links isolated in the RL simultaneously since they share at least one of the red underlay links. In all initial RLSets except $\mathcal{L}(v_4)$, overlay links sharing the identical underlay links are isolated by multiple RLs since the connectivity of the RL is lost when those overlay links are isolated in a single RL.

3.2.2 RLSet integration

The initial RLSets constructed by the overlay nodes are aggregated and integrated into a single RLSet, which would be shared among all overlay nodes. In addition, the author tries to merge any two RLs into a single RL to decrease the number of RLs in RLSet and improve the recovery performance of RLSet. The merging algorithm is quite simple: the overlay links and overlay nodes isolated in both RLs become isolated in the merged RL. Here, two RLs are merged into a single

Algorithm 1 Initial RLSet construction

```
Input: Overlay node $v_i$, overlay network $G_o$
Output: Initial RLSet $\mathcal{L}(v_i)$ constructed by $v_i$
$\mathcal{L}(v_i) \leftarrow \emptyset$ /* $\mathcal{L}(v_i)$ initialize as empty set */
$s \leftarrow 1$
for all $j$ s.t. $v_j \in V(G_o)$ and $\exists o_{ij} \in E(G_o)$ do
    /* Isolate $o_{kl}$ that overlaps the underlay link with $o_{ij}$ in $L_s^i$ */
    for all $u_{mn} \in r_{ij}(L_0)$ do
        if $L_s^i \neq L_0$, and $L_s^i$ is not the connected graph when all $o_{kl}$ overlapping $u_{mn}$ are isolated then
            /* When no more overlay link cannot be isolated in $L_s^i$, the other overlay links are isolated in new RL $L_{s+1}^i$ */
            $\mathcal{L}(v_i) \leftarrow \mathcal{L}(v_i) \cup \{L_s^i\}$
            $s \leftarrow s + 1$
        end if
        for all $r_{kl}(L_0) \ni u_{mn}$ do
            $w_s^i(o_{kl}) \leftarrow \infty$
        end for
    end for
    $\mathcal{L}(v_i) \leftarrow \mathcal{L}(v_i) \cup \{L_s^i\}$
    $s \leftarrow s + 1$
end for
return $\mathcal{L}(v_i)$
```



(a) $\mathcal{L}(v_1)$

(b) $\mathcal{L}(v_2)$

(c) $\mathcal{L}(v_3)$

(d) $\mathcal{L}(v_4)$

(e) $\mathcal{L}(v_5)$

Figure 4: Example of initial RLSets

RL only when the following conditions are satisfied.

(1) The merged RL keeps the connectivity.

(2) The overlay node that is isolated in neither the RLs does not become isolated in the merged RL.

(3) All isolated overlay nodes in the merged RL are connected to at least one normal overlay node.

Note that the failure that can be recovered in the original RLs can be recovered in the merged RL. Moreover, the number of failure patterns that can be recovered in the merged RL increases since the number of isolated overlay nodes and links increases. This is one of important effects in merging RLs. Another advantage is that it requires shorter time and smaller memory usage in maintaining RLSet and searching RL in RLSet since the number of RLs in RLSet decreases. However, the excessive merging has a bad effect on the path length in the merged RL since the number of available overlay links decreases due to the increase of isolated overlay links. We should also take care of the merging order of RLs in the initial RLSet at all overlay nodes since it determines the recovery performance of the merged RL from multiple simultaneous failures, as described in Subsection 2.2. Considering these issues, the author proposes the merging and integration process of RLs in the initial RLSets at all overlay nodes into a single RLSet as follows.

Algorithm 2 presents the construction algorithm of RLSet from RLs in the initial RLSets at all overlay nodes. In this algorithm, the author introduces two states: *merging* and *waiting* and each overlay node behaves as follows. First, all overlay nodes become the merging state, and the overlay node v_p that has the most underlay links is selected as the starting point of the integration. Second, v_p becomes the waiting state and aggregates the RLSet constructed by its adjacent overlay nodes whose state is merging into the RLSet at v_p. Before aggregating v_i's RLSet into v_p's RLSet, v_i becomes the merging state and aggregates recursively the RLSets at its adjacent overlay nodes whose state is merging into v_i's RLSet. By the recursive aggregating the RLSets, all RLSets throughout the network are aggregated with only information of the state of adjacent overlay nodes.

Detailed recursive behavior of an overlay node v_i whose state is merging is as follows. First, v_i becomes the merging state. Second, v_i adds the RLs in all RLSets at adjacent overlay nodes whose state is merging to v_i's RLSet $\mathcal{L}(v_j)$. Then, for all RL pairs $L^i_s, L^i_t \in \mathcal{L}(v_i)$, L^i_s and L^i_t are merged into a single RL $L^i_{s,t}$ when $L^i_{s,t}$ satisfies three conditions described above. When $L^i_{s,t}$ does not satisfies the conditions, L^i_s and L^i_t are not merged since there exists failure patterns from which either L^i_s or L^i_t can recover and $L^i_{s,t}$ cannot recover. Furthermore, to avoid the bad effect of merging RLs described above, two RLs L^i_s and L^i_t are merged into $L^i_{s,t}$ only when the following condition are satisfied.

$$f(L^i_{s,t}) \quad \geq \quad \frac{f(L^i_s) + f(L^i_t)}{2} \tag{1}$$

where,

$$\begin{aligned} f(L^i_s) &= \alpha I(L^i_s) - \beta A(L^i_s) \qquad (2) \\ I(L^i_s) &= \mid \{o_{jk} \mid w^i(o_{jk}) = \infty\} \mid \\ A(L^i_s) &= \frac{1}{N_o} \sum_{j \ s.t. \ v_j \in V(G_o)} \sum_{k \ s.t. \ k \neq j, v_k \in V(G_o)} |r_{jk}(L^i_s)| \\ N_o &= \frac{2}{\mid V(G_o) \mid (\mid V(G_o) \mid -1)} \end{aligned}$$

The functions I and A gives the number of isolated links in RL and the average hop counts between overlay nodes in RL, respectively. Equation (2) defines an evaluation metric for the effectiveness of RL, where α and β are the parameters to determine the contribution degree of I and A in Equation (1). When $L^i_{s,t}$ satisfies the conditions for merging and Equation (1), L^i_s, L^i_t are removed from $\mathcal{L}(v_i)$ and $L^i_{s,t}$ is added to $\mathcal{L}(v_i)$. This merging process is repeated until there becomes no RL pair that can be merged.

Figure 5 shows examples of merging process. The RL pair that satisfies the conditions for merging and Equation (1) is shown in Figure 5(a). Since the merged RL isolates one additional overlay link while keeps path length small, it is expected that the merged RL improves the recovery performance. Figure 5(b) shows the RL pair that satisfies the conditions for merging but does not satisfy Equation (1). The merged RL isolates two additional overlay links but increases path length largely because the merged RL becomes chain-like topology. The RL pair shown in Figure 5(c) cannot be merged since the merged RL is not connected graph. The RL pair shown in Figure 5(d) cannot be also merged since the merged RL isolates an additional overlay node.

Through the above process, all RLs in all overlay nodes are integrated into a single RLSet at v_p, which is the starting point of the aggregation. The RLSet should be distributed to overall network to share it by all overlay nodes. The distribution method is out of scope of this thesis since the existing methods [35, 36] can be utilized.

3.3 RL selection

When packets are routed according to the proposed method, there are two ways to select an RL from the RLSet constructed according to the algorithms in Subsection 3.2, which are static RL selection and dynamic RL selection. We summarize the details of both selection methods since both of them have advantages and disadvantages.

3.3.1 Static RL selection

In static RL selection, when a failure is detected by a source node, the source node selects an RL from RLSet according to the detected failed nodes and keeps using the RL until packets arrive at the destination node. In detail, the source node selects an RL in which all failed nodes are isolated. When more than one RL are found, the source node selects one of them that has the smallest number of isolated nodes. In this case, the proposed method can guarantee full network reachability. Conversely, when there is no RL in which all failed nodes are isolated, the source node selects the RL that has the largest number of failed nodes as isolated. In this case, the selected RL cannot completely guarantee network reachability. Obviously, static RL selection is simpler than dynamic RL selection described below, since there is no need for the intermediate nodes to select an RL in a packet-by-packet manner. This algorithm is shown in Algorithm 3.

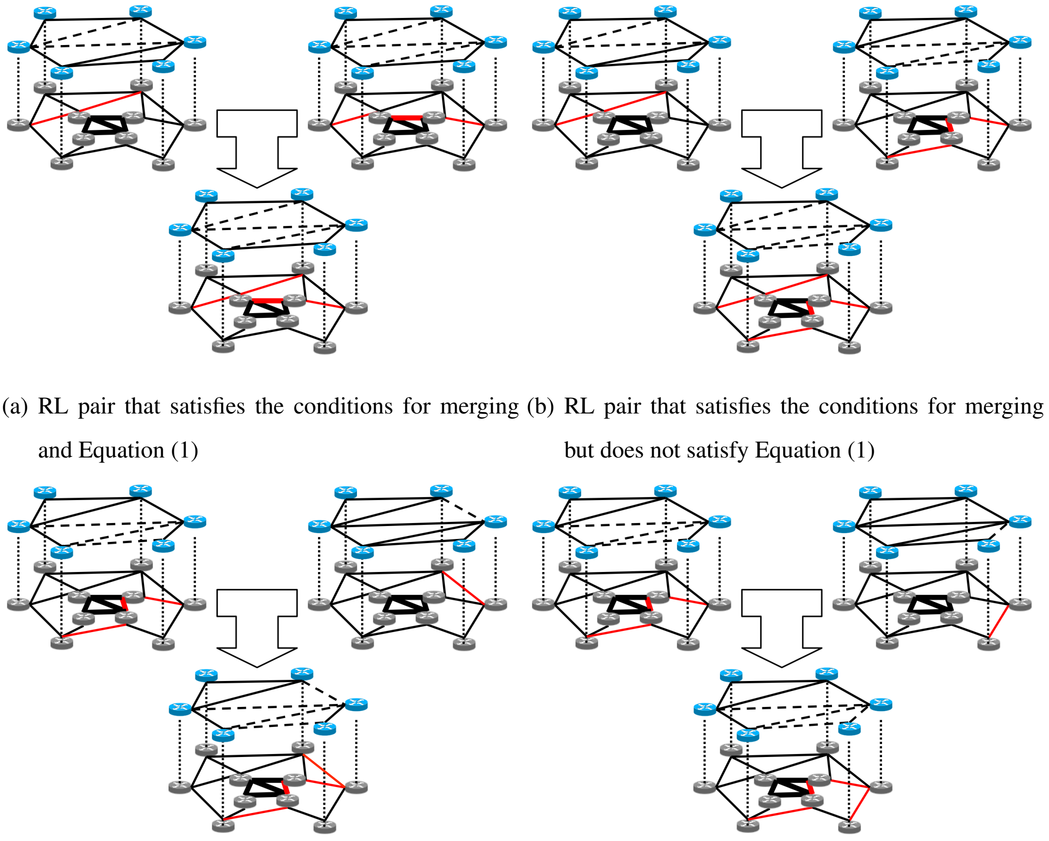
3.3.2 Dynamic RL selection

The dynamic RL selection permits intermediate nodes to change the RL to be used. In detail, when one of the intermediate nodes finds that it cannot forward a packet to the next-hop node because the failure is not covered by currently-used RL, the node will change RL so that the packet can be forwarded to the next-hop node. In general, this on-demand RL selection creates a routing loop by repeated changes of RLs in some intermediate nodes. However, in the proposed method, we avoid routing loop by forcing the node to use a new RL that has larger number of isolated nodes than the current RL. The proposed method can forward packets to the destination node unless RLs in RLSet are not exhausted. This algorithm is shown in Algorithm 4.

Algorithm 2 RLSet construction

```
Input: Overlay network $G_o$, the initial RLSet $\mathcal{L}(v_i)$ constructed by $\forall v_i$
Output: RLSet $\mathcal{L}(v_p)$ that aggregated all initial RLSet
$p \leftarrow i$ s.t. $v_i \in V(G_o)$
for all $v_i \in V(G_o)$ do
    if $d(v_p) < d(v_i)$ then
        $p \leftarrow i$ /* The overlay node that has the most underlay links is selected as $v_p$ */
    end if
    $s(v_i) \leftarrow 0$ /* All overlay nodes are initialized as the merging state */
end for
return Integration$(v_p)$ /* Aggregate the initial RLSet recursively */

Integration$(v_i)$
$s(v_i) \leftarrow 1$ /* $v_i$ becomes the waiting state */
for all $j$ s.t. $s(v_j) = 0$ and $o_{ij} \in E(G_o)$ do
    /* Aggregate the RLSets $\mathcal{L}(v_j)$ into $\mathcal{L}(v_i)$, where $v_j$ is adjacent to $v_i$ in overlay network
    and the state of $v_j$ is merging */
    $\mathcal{L}(v_i) \leftarrow \mathcal{L}(v_i) \cup$ Integration$(v_j)$
end for
repeat
    /* Repeat the following behavior until there is no RL pair can be merged */
    $\mathcal{L}' \leftarrow \mathcal{L}(v_i)$
    for all $L^i_s \in \mathcal{L}(v_i)$ do
        for all $L^i_t \in \mathcal{L}(v_i)$ s.t. $s \neq t$ do
            Merge $L^i_s$ and $L^i_t$ into $L^i_{s,t}$
            if $L^i_{s,t}$ satisfies the conditions for merging and Equation (1) then
                $\mathcal{L}(v_i) \leftarrow \mathcal{L}(v_i) - \{L^i_s, L^i_t\} \cup \{L^i_{s,t}\}$ /* Replace $L^i_s$ and $L^i_t$ with $L^i_{s,t}$ */
            end if
        end for
    end for
until $\mathcal{L}(v_i) = \mathcal{L}'$
return $\mathcal{L}(v_i)$
```



(a) RL pair that satisfies the conditions for merging and Equation (1)

(b) RL pair that satisfies the conditions for merging but does not satisfy Equation (1)

(c) RL pair that cannot be merged since the topology of the merged RL is unconnected

(d) RL pair that cannot be merged since the merged RL isolates one additional overlay node

Figure 5: Merging process of RLs

Algorithm 3 Static RL Selection

```
if there exists a certain RL in which all failures are isolated then
    Select the RL with the minimum number of safe nodes
else
    Select the RL with the minimum number of isolated nodes and which makes failures isolated the most
end if
```

Algorithm 4 Dynamic RL Selection

```
if there exists a certain RL in which all failures are isolated then
    Select the RL with the minimum number of isolated nodes
else
    Select $L_0$ tentatively
    repeat
        if next node is active then
            Forward next node with tentatively selected RL
        else if next node is down then
            if tentative RL is not last RL then
                Select next RL tentatively
            else
                Abort forwarding
            end if
        end if
    until reach destination node
end if
```

This dynamic mechanism can increase the network reachability after recovery, even when there is no RL in RLSet that makes all failures isolated. However, it may increase the processing delay at intermediate nodes.

3.4 Accommodation of network topology changes

In general, the computer networks are always changing by adding and removing network elements and occurring failures. For the proposed method, we should consider changes in both underlay and overlay networks. In what follows in this subsection, the author describes the methods to accommodate changes in overlay networks and those in underlay networks, respectively, to keep the recovery performance of the proposed method.

3.4.1 Partial reconstruction for overlay network changes

Ideally, the RLSet should be recalculated and distributed to network nodes against every change in overlay networks. However, the frequent recalculation and distribution of RLSet should be avoided due to the viewpoints of calculation overhead and distribution delay. Therefore, the proposed method employs partial reconstruction of RLSet that can be done in parallel at each overlay node.

When a new overlay node/link is added to the existing overlay network, the new overlay node/link is added to the overlay network topologies of all RLs in RLSet. At this point, the newly-added node/link is not isolated in any RL, so the RLSet does not support any failure patterns including the node/link. Therefore, the newly-added node/link should be isolated in some RLs in RLSet. When an overlay link is added, it is isolated in RLs in which the new link is connected to at least one isolated node. When an overlay node is added, each overlay node searches RLs in RLSet where the new node is connected to at least one normal overlay node. Among such RLs, each overlay node selects one RL with minimum number of isolated nodes and isolate the newly-added node in the selected RL. Finally, each overlay node modifies the routing configurations for each RL in RLSet, and the newly-added overlay node receives the reconstructed RLSet from its adjacent overlay node. Note that the above calculations can be done at each overlay node in a distributed fashion. Therefore, no information exchanges is required between overlay nodes. Algorithms 5 and 6 describe the pseudo codes for overlay nodes when adding overlay nodes and overlay links, respectively.

On the other hand, when an overlay node/link is removed from the overlay network, each overlay node removes the overlay node/link and modifies the routing configurations for each RL in RLSet. Algorithms 7 and 8 are the pseudo codes of the process in each overlay node. Note that the proposed method utilizes the RLSet constructed before the overlay network changes until removing algorithms complete reconstructing RLSet. Furthermore, the proposed method maintains the old RLSet for a while to be used when removed overlay nodes and links join the overlay network again in short time because of node reboot, link resetting, and so on.

However, by utilizing the above algorithm, the recovery performance may degrade in some situations. The author explains the problem by using Figure 6, which depicts the case when a new overlay node connects to one isolated overlay node and one normal overlay node (Figure 6(a)), and the case when a new overlay node connects only to two isolated overlay nodes (Fig. 6(b)).

Algorithm 5 New overlay node addition

Input: RLSet $\hat{\mathcal{L}}$, added overlay node v_n, set of added overlay links E'_o

Output: Partial reconstructed RLSet $\hat{\mathcal{L}}$

```
if $v_i \neq v_n$ then
    /* A new overlay node and links are added to the overlay network */
    $V(G_o) \leftarrow V(G_o) \cup \{v_n\}$
    $E(G_o) \leftarrow E(G_o) \cup E'_o$
    $I_M \leftarrow \infty$
    for all $L_s \in \hat{\mathcal{L}}$ do
        /* Search RL in which can isolate $o_{nj} \in E'_o$ */
        if $I(L_s) < I_M$ and $v_n$ connects at least one normal overlay node in $L_s$ then
            $I_M \leftarrow I(L_s)$, $M \leftarrow k$
        end if
    end for
    /* Isolate $o_{nj} \in E'_o$ in $L_M$ */
    for all $o_{nj} \in E'_o$ do
        $w_M(o_{nj}) \leftarrow \infty$
    end for
end if
for all $L_s \in \hat{\mathcal{L}}$ do
    Calculate the routing configuration of $L_s$
end for
return $\hat{\mathcal{L}}$
```

Algorithm 6 New overlay link addition

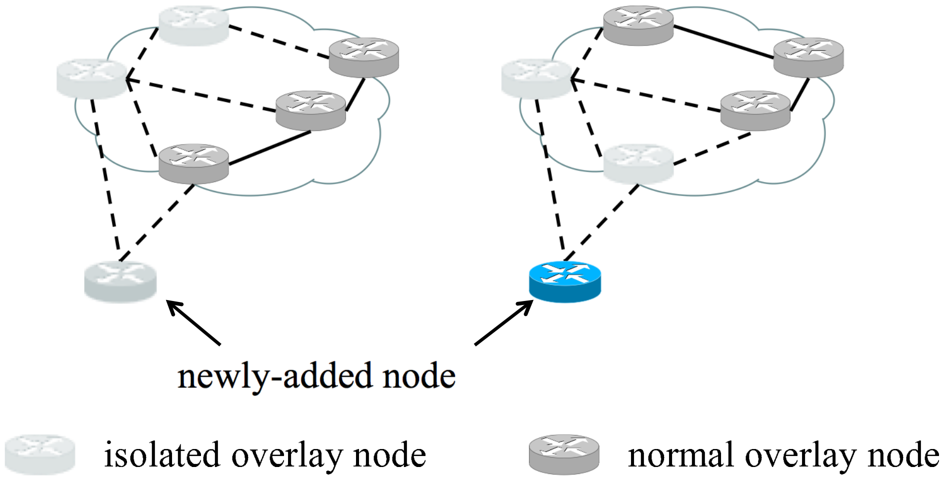
```
Input: RLSet $\hat{\mathcal{L}}$, added overlay link $o_{nm}$
Output: Partial reconstructed RLSet $\hat{\mathcal{L}}$
if $v_i \neq v_n$ then
    /* A new overlay link is added to the overlay network */
    $E(G_o) \leftarrow E(G_o) \cup \{o_{nm}\}$
    $I_M \leftarrow \infty$
    for all $L_s \in \hat{\mathcal{L}}$ do
        /* Isolate $o_{nm}$ when $o_{nm}$ connects at least one isolated overlay node */
        if $o_{nm}$ connects at least one isolated overlay node in $L_s$ then
            $w_s(o_{nm}) \leftarrow \infty$
        end if
    end for
end if
for all $L_s \in \hat{\mathcal{L}}$ do
    Calculate the routing configuration of $L_s$
end for
return $\hat{\mathcal{L}}$
```

Algorithm 7 Overlay node deletion

```
Input: RLSet $\hat{\mathcal{L}}$, removed overlay node $v_i$
Output: Partial reconstructed RLSet $\hat{\mathcal{L}}$
/* Overlay node and its overlay links are removed from the overlay network */
$V(G_o) \leftarrow V(G_o) - \{v_i\}$
for all $j$ s.t. $o_{ij} \in E(G_o)$ do
    $E(G_o) \leftarrow E(G_o) - \{o_{ij}\}$
end for
for all $L_s \in \hat{\mathcal{L}}$ do
    Calculate the routing configuration of $L_s$
end for
return $\hat{\mathcal{L}}$
```

Algorithm 8 Overlay link deletion

```
Input: RLSet $\hat{\mathcal{L}}$, removed overlay link $o_{ij}$
Output: Partial reconstructed RLSet $\hat{\mathcal{L}}$
/* Overlay node and its overlay links are removed from the overlay network */
$E(G_o) \leftarrow E(G_o) - \{o_{ij}\}$
for all $L_s \in \hat{\mathcal{L}}$ do
    Calculate the routing configuration of $L_s$
end for
return $\hat{\mathcal{L}}$
```



(a) An RL in which new node is isolated

(b) An RL in which new node connects only isolated nodes

Figure 6: Problems in joining new nodes

In the former case, the newly-added overlay 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 path to and from the node in this RL. Therefore, when this RL is selected for failure recovery, the overlay network reachability degrades.

To solve this problem, we need the overall reconstruction of the RLSet, which means that each overlay node constructs the RLSet according to Algorithms 1 and 2, to maintain the recovery performance. In Section 4, the author evaluates the performance degradation caused by this problem and discusses the appropriate interval for overall recalculation of RLSet against network growth.

3.4.2 Overall reconstruction for underlay network changes

When the underlay network changes, the proposed method should reconstruct the RLSet since the proposed method is based on the correlation among overlay links in terms of utilizing underlay link. Specifically, each overlay node constructs the RLSet according to Algorithms 1 and 2.

4 Performance evaluations

In this section, the author presents evaluation results of recovery performance of the proposed method. The evaluation method is shown in Subsection 4.1. The results of overlay network reachability and the path length are shown in Subsection 4.2 and Subsection 4.3, respectively. In Subsection 4.4, the author represents the evaluation results of the performance degradation caused by the partial reconstruction of RLSet as described in Subsection 3.4.

4.1 Evaluation method

To evaluate the proposed method, two kinds of network topologies are utilized for underlay network topology. The utilized underlay networks are as follows.

- AT&T topology

  This is a router-level topology in the actual ISP in the United States, which can be found in [37]. The underlay network topology has 523 underlay nodes (routers) and 1304 underlay links, meaning that the average degree is around 2.5.

- BA topology

  This topology is generated by the topology generator BRITE [33], which follows the Barabási-Albert (BA) model in [38]. The topology starts with a network topology of 50 nodes and 194 links, and add a new node with 2 links to the network in a one-by-one manner. The topology growth continues until the topology has 100 nodes and 294 links. In the evaluation, the authors generate 100 different topologies for averaging the results.

The author assumes that overlay networks are built on those underlay networks. Figure 7 shows examples of underlay and overlay network topologies, which is generated by Graphviz [39]. Note that part of underlay nodes become overlay nodes, and overlay links are established among those overlay nodes. Two kinds of overlay network topology are also utilized. Here, the parameter n means the number of overlay nodes, and the parameters that determines the number of overlay links in ER topology and BA topology are denoted as l_e and l_b, respectively. Besides, the ranges of l_e and l_b are from 1 to $(n-1)/2$.

- ER topology

  This topology follows the Erdös-Rényi (ER) model in [40]. In the topology, n underlay nodes become overlay nodes in this topology, and all overlay node pairs establish an overlay link between them with the probability $2l_e/(n-1)$. The topology finally has n nodes and approximately nl_e links

- BA topology

  This topology follows the Barabási-Albert (BA) model in [38]. The topology starts with a full mesh network topology of l_b nodes and $l(l-1)/2$ links, and add a new node with l_b links to the network in a one-by-one manner. The topology growth continues until the topology has n nodes and $l(2n-l-1)/2$ links.

Note that the number of overlay links are approximately the same in both topologies when $n \gg l_e$, $n \gg l_b$, and $l_e = l_b$. For example, when $n = 100$, $l_e = 4$, and $l_b = 4$ are given, the number of overlay links in ER topology is around 400 and that in BA topology is 390.

For simulating the multiple simultaneous failures, the author utilizes the following two failure types.

- Random failures

  The failures grow by stopping randomly-selected underlay links.

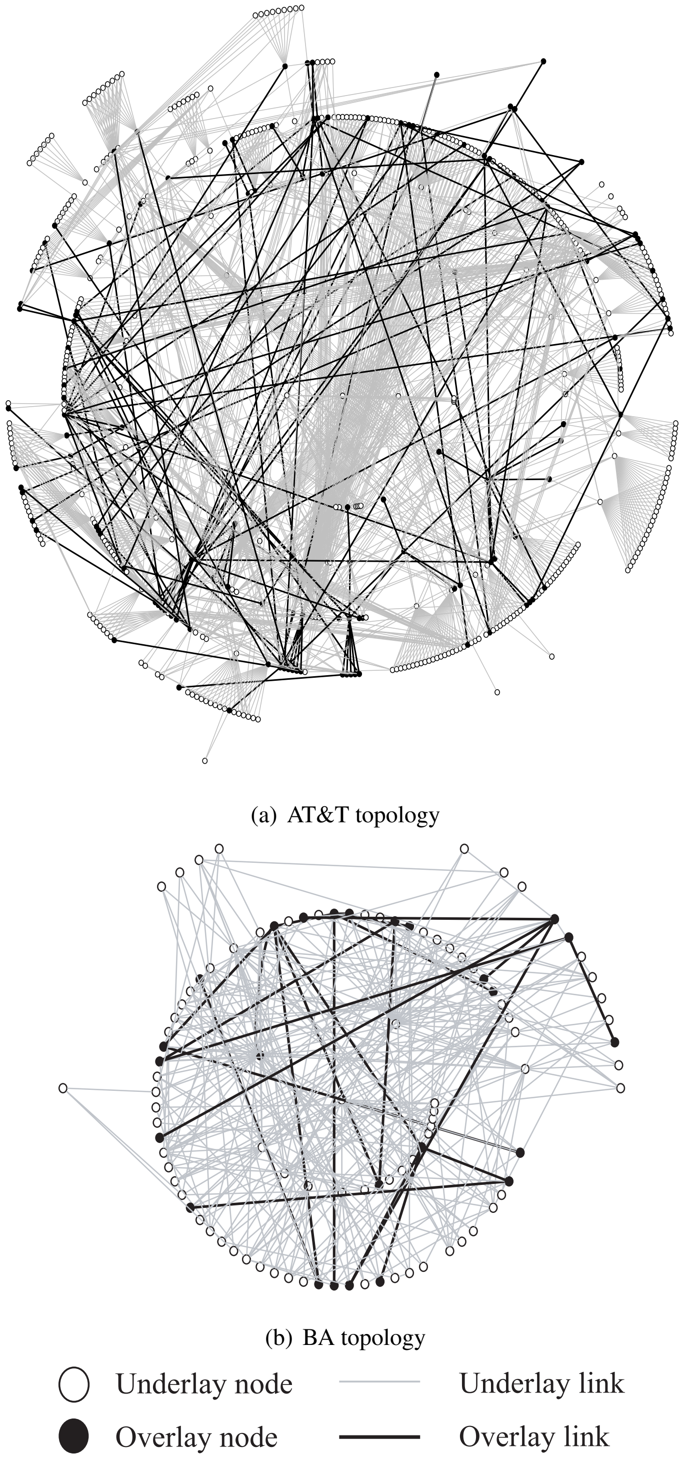
- Adjacent failures

  The failures stop randomly-selected underlay links firstly, and then they grow by stopping the underlay link that is adjacent to the failed underlay links.

In the following evaluations, the author sets $\alpha = \beta = 1$ in Equation (1), $l_e = 4$, $l_b = 4$ and the ratio of the number of overlay nodes to the number of underlay nodes, which is defined as *overlay node density*, is 0.25 except as otherwise noted. The overlay routing selects the path to minimize the number of hop counts in overlay network.

Table 2 shows the relationships between the number of overlay nodes and RLs generated by the proposed method when BA topology is utilized for overlay network. From this table, we can find that the increase of the number of RLs is smaller than the increase the number of overlay nodes in BA topology. This is because when the number of overlay nodes increases, the candidate RLs for merging also increase.

The author evaluates the overlay network reachability defined by the ratio of reachable overlay node pairs after recovering from the failure to all overlay node pairs in the overlay network except


(a) AT&T topology

(b) BA topology

(c) Types of nodes and links in above graphs

Figure 7: Examples of underlay network and overlay network

Table 2: Relationships between the number of overlay nodes and generated RLs

Underlay network	# of overlay nodes	# of overlay links	Average degree	# of average RLs
BA topology	10	34	3.4	10.7
	25	94	3.8	16.4
	50	194	3.9	24.1
	75	294	3.9	34.8
	100	394	3.9	45.9
AT&T topology	131	514	3.9	399

the failed nodes. In addition, the path length, which means underlay hop counts, between all reachable node pairs is evaluated.

In the evaluation results given in the following subsections, the author plots the results of two extreme cases for comparison purposes: *Ideal*, which represents the results of the ideal case where we recalculate the routing configurations after removing failed underlay links, and *Normal*, which represents the results in the original topology without applying any failure recovery mechanisms. Ideal and Normal provide the upper and lower limit of the network reachability, respectively. In addition, the results of the proposed method with static and dynamic RL selection are denoted as *Proposal* and *ProposalDY*, respectively.

4.2 Overlay network reachability

Figure 8 shows the evaluation results of the overlay network reachability as a function of the number of failed underlay links in the underlay network based on BA topology and the overlay network based on ER topology, against random and adjacent failures, respectively. In addition , the author denotes this network as BA-ER network and utilizes the similar abbreviation in what follows. Figure 9 shows the corresponding results in BA-BA network. From these figures, it is found that the proposed method improves the overlay network reachability in almost all cases. For example in Figure 9(a), the dynamic RL selection improves the overlay network reachability from 41 % to 81 % when 64 underlay links go down simultaneously.

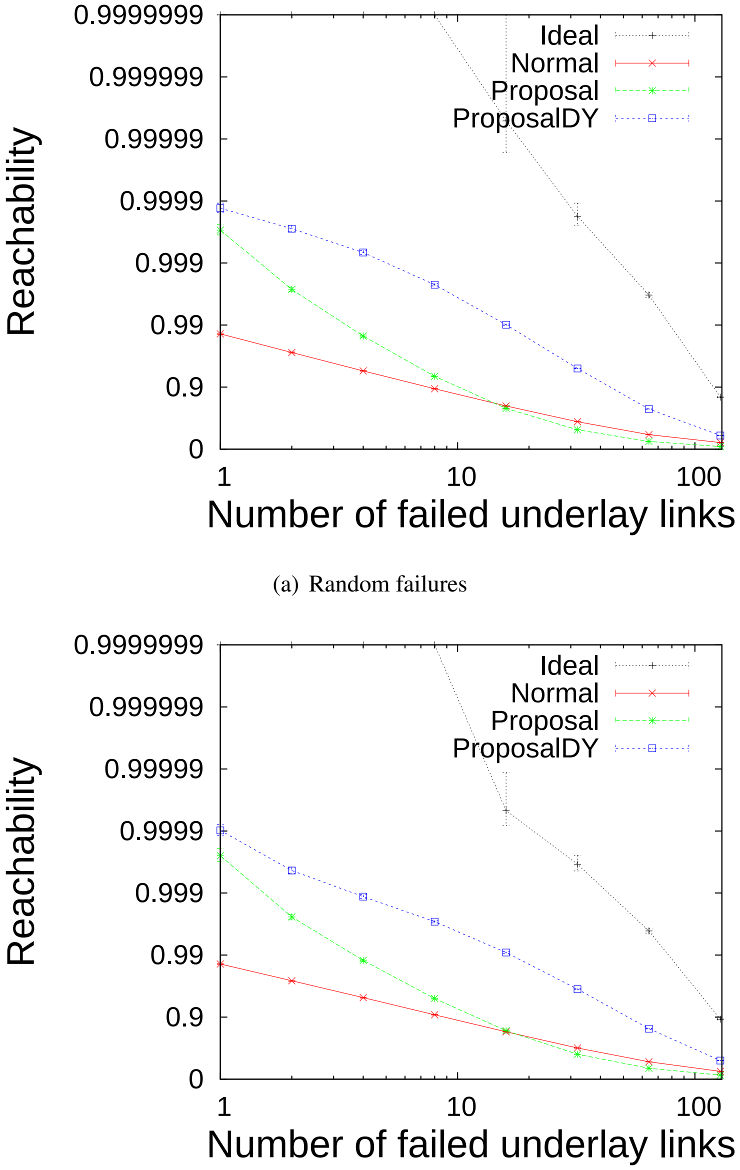
More precisely, the dynamic RL selection provides larger improvement in the overlay network reachability than the static RL selection. This is because that the difference behavior of static and dynamic RL selection affects the overlay network reachability when there is no RL in RLSet that makes all failures isolated. With static RL selection, the proposed method cannot provide complete reachability since it utilizes the RL that does not isolates all failures. In contrast, the dynamic RL selection can avoid all failures by utilizing multiple RLs. However, the dynamic selection is inferior to Ideal case since it avoids utilizing the RL that selected before.

On the other hand, when a large number of underlay links go down simultaneously, the performance of the static RL selection becomes lower than that of the original topology. This may be attributable to the selected RL by static RL selection because the static selection selects the RL that has the largest number of failed overlay links as isolated rather than the original topology. That is, static RL selection is effective against the small number of failures, but it may decrease the overlay network reachability when a large number of underlay links fail simultaneously.

Furthermore, it is found that the difference between the results against the random and adjacent failures is quite small. In addition, we can observe that BA-ER network and BA-BA network show the similar performance when comparing Figures 8 and 9. This is because the proposed method can recover effectively from multiple simultaneous failures against any overlay networks by considering the correlation among overlay links in terms of utilizing underlay link. Therefore, the following evaluations in this subsection utilize BA topology for overlay network and adjacent failures to avoid redundant explanations except as otherwise noted.

Figure 10 shows the corresponding results to Figure 9 with the AT&T topology as underlay network. We can see from these figures that the dynamic RL selection gives the similar performance to Ideal case especially when the number of failure links increases. For example, the overlay network reachability is improved from 51% to 97% when 128 underlay links fails simultaneously. The reason is that the upper limit of the number of RL switching at intermediate overlay nodes increases since the AT&T topology has more nodes than the BA topology (in this case the number of RLs is 399 as shown in Table 2). In contrast, the static RL selection degrades the reachability. This is caused by the increase of the failure patterns as increasing the number of underlay links.

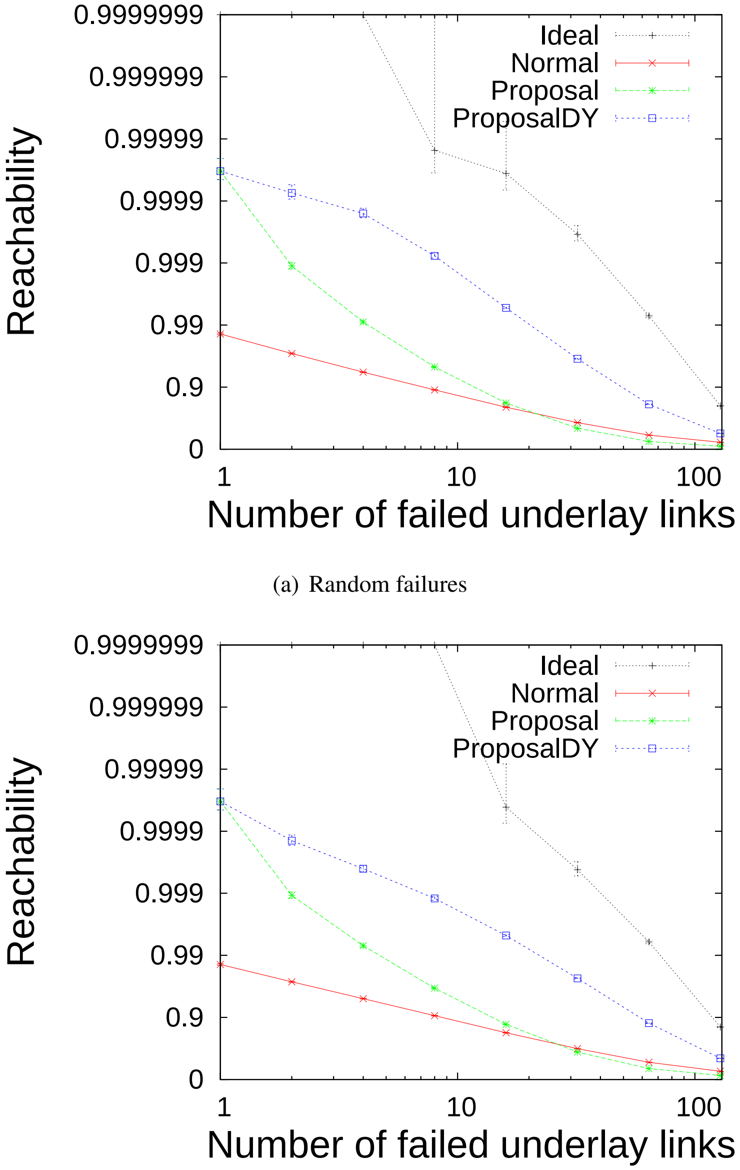
The changes in the overlay network reachability with overlay node density are shown in Figure 11. From this figure, we can observe that as the number of overlay nodes increases, the reachability of the dynamic RL selection increases. The reason is the increase of the number of RLs in RLSet



(a) Random failures

(b) Adjacent failures

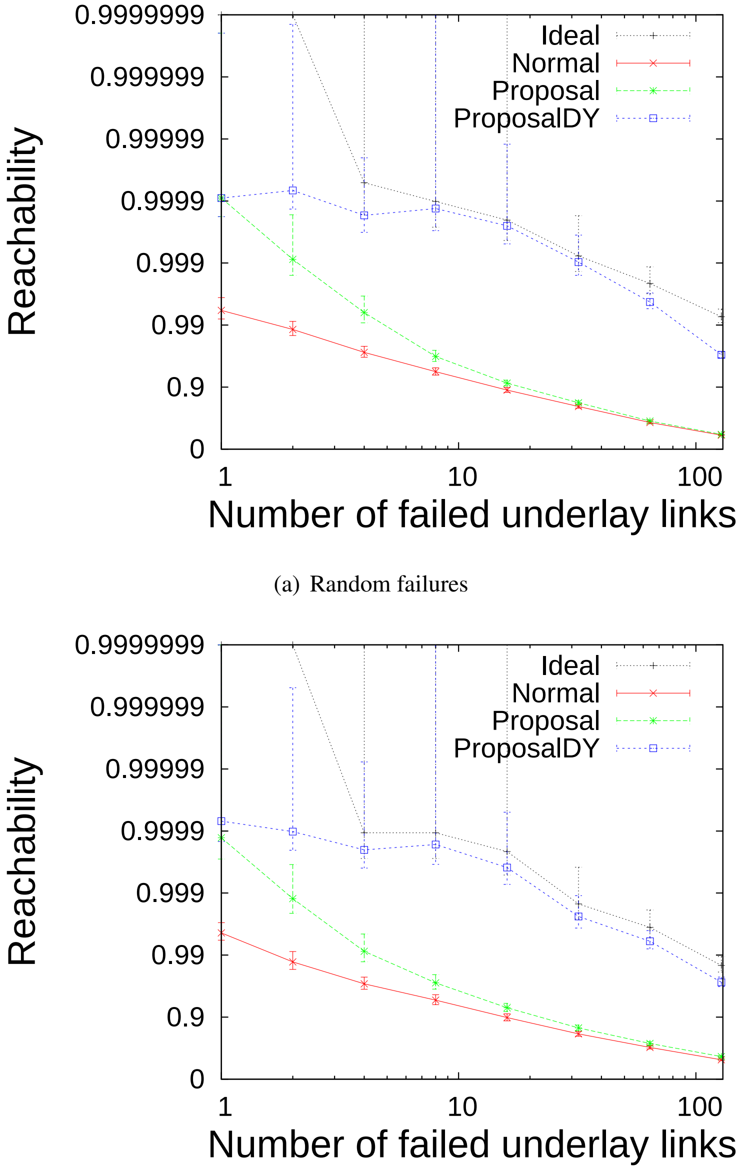
Figure 8: Overlay network reachability in BA-ER network



(a) Random failures

(b) Adjacent failures

Figure 9: Overlay network reachability in BA-BA network



(a) Random failures

(b) Adjacent failures

Figure 10: Overlay network reachability in AT&T-BA network

since the number of overlay nodes increases as described in Table 2. Inversely, the reachability of the static RL selection decreases as the number of overlay nodes increases. This is because that when there are a small number of overlay nodes, RLs that isolate lots of overlay links are often constructed since the overlay links overlap many underlay links. In contrast, when the number of overlay nodes increases, the RLs that isolate a large number of overlay links are hardly constructed since the overlay links overlap few underlay links.

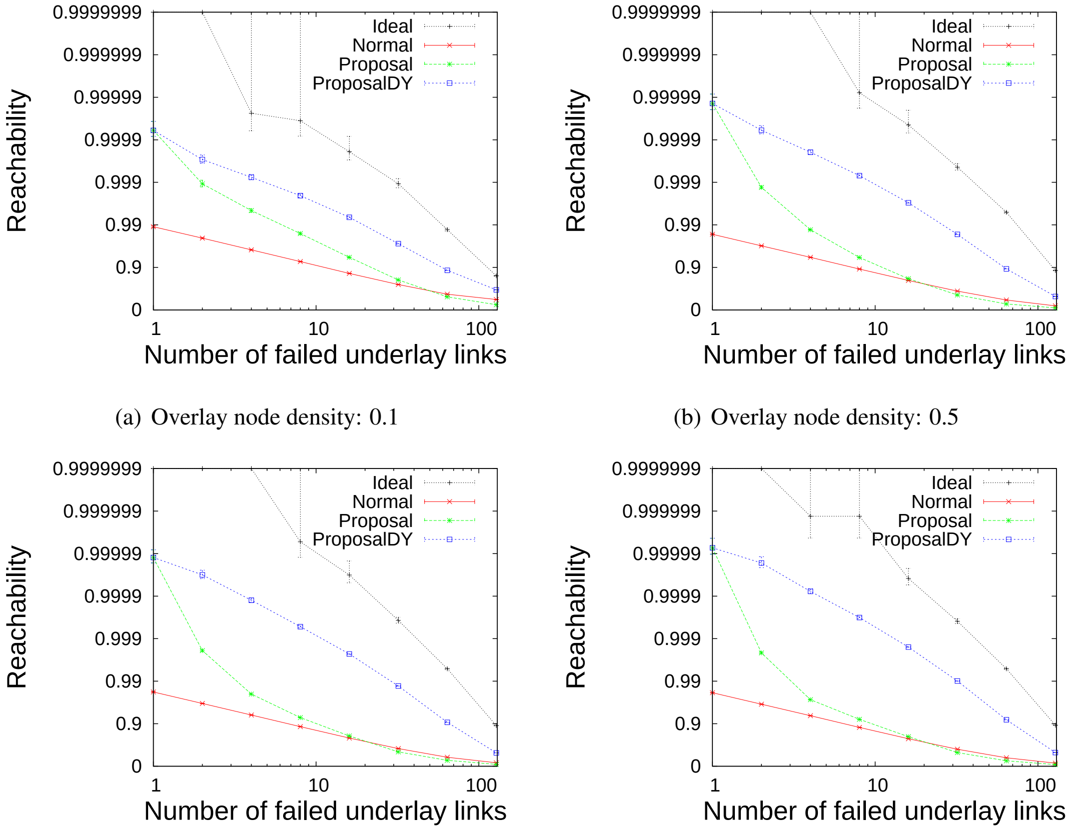
Figure 12 illustrates the effect of α on the overlay network reachability. From this figure, it is found that the reachability of the proposed method with both RL selection decreases when α is extremely small. The reason is that the number of isolated overlay links in each RL decreases since the most of RL pairs in RLSet cannot be merged. In a similar fashion, the changes in the overlay network reachability in the case of β = 10, 100, and 1000 are shown in Figure 13. Again, it is seen that the reachability of the proposed method decreases when β becomes large since very few RL pairs are likely to be merged into a single RL with large β.

4.3 Path length

Figure 14 shows the evaluation results of the average path length as a function of the number of failed underlay links in BA-ER network, against random and adjacent failures, respectively. Figure 15 shows the corresponding results in BA-BA network. From these figures, we can observe that the average path length of the static and dynamic RL selections increases by up to 43% and 27 %, respectively. This is attributable to the isolation of overlay links in RLs. More precisely, the reason why the path length of the dynamic RL selection is smaller than those of the static RL selection is that when there is no RL that all failures are isolated, the dynamic RL selection utilizes shorter paths by selecting the RL that has smaller number of isolated overlay links, while the static RL selection utilizes longer paths by selecting the RL that has the largest number of failed nodes as isolated.

On the other hand, when the number of failed underlay links increases, the results of the original topology and the static RL selection decrease. Note that these are not the results of decreasing the path length itself but that the overlay node pairs in long path lose the connectivity.

Figure 16 presents the corresponding results to Figure 15 in the AT&T-BA network. From this figure, we can see that the proposed method slightly increases the average path length by up to 1.6 % compared with that of Ideal against both failures. This is because the difference of path length



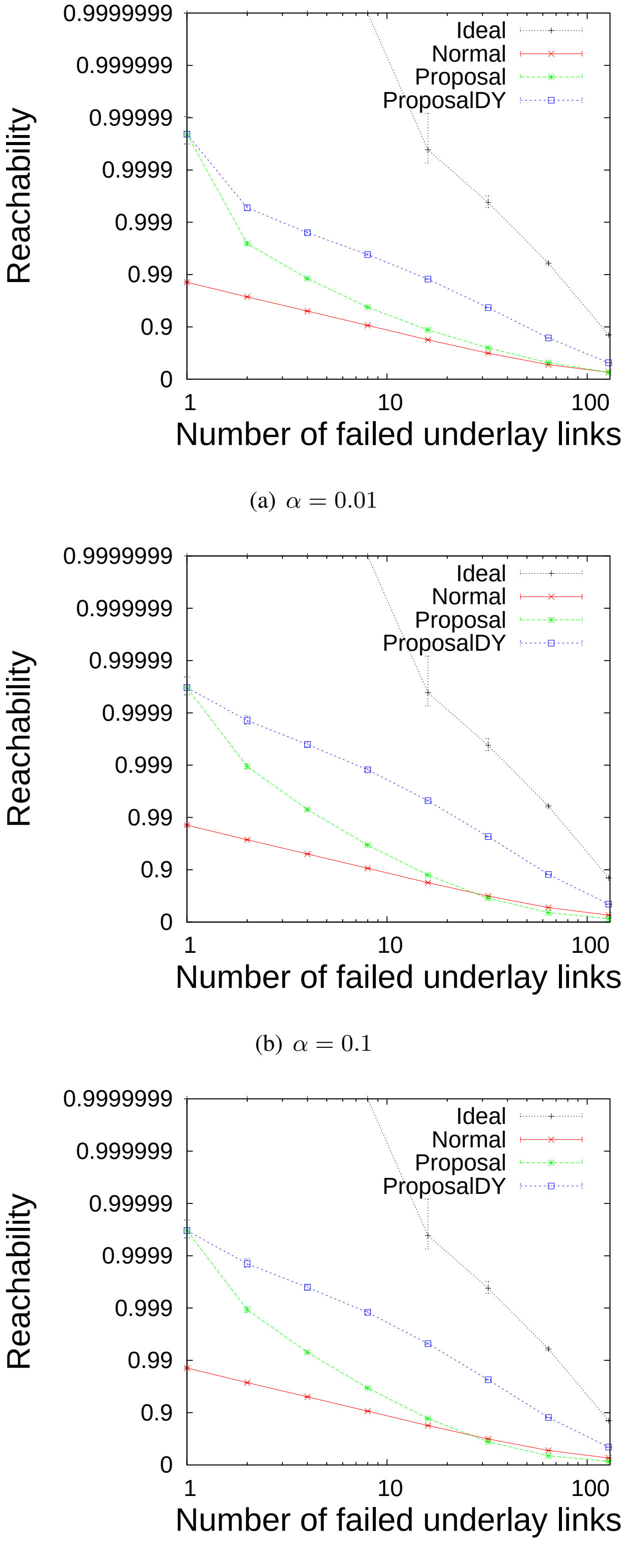
(a) Overlay node density: 0.1

(b) Overlay node density: 0.5

(c) Overlay node density: 0.75

(d) Overlay node density: 1

Figure 11: Changes in overlay network reachability with overlay node density

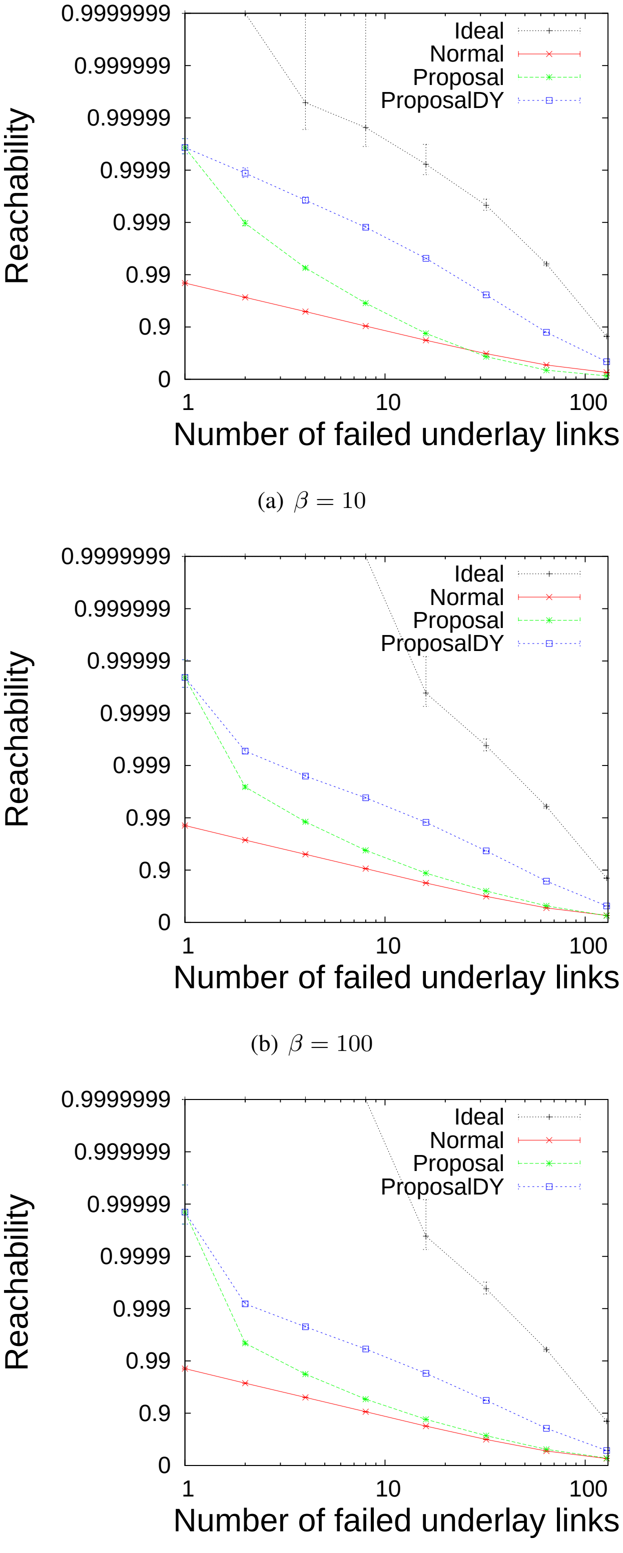


(a) $\alpha = 0.01$

(b) $\alpha = 0.1$

(c) $\alpha = 10$

Figure 12: Changes in overlay network reachability with α

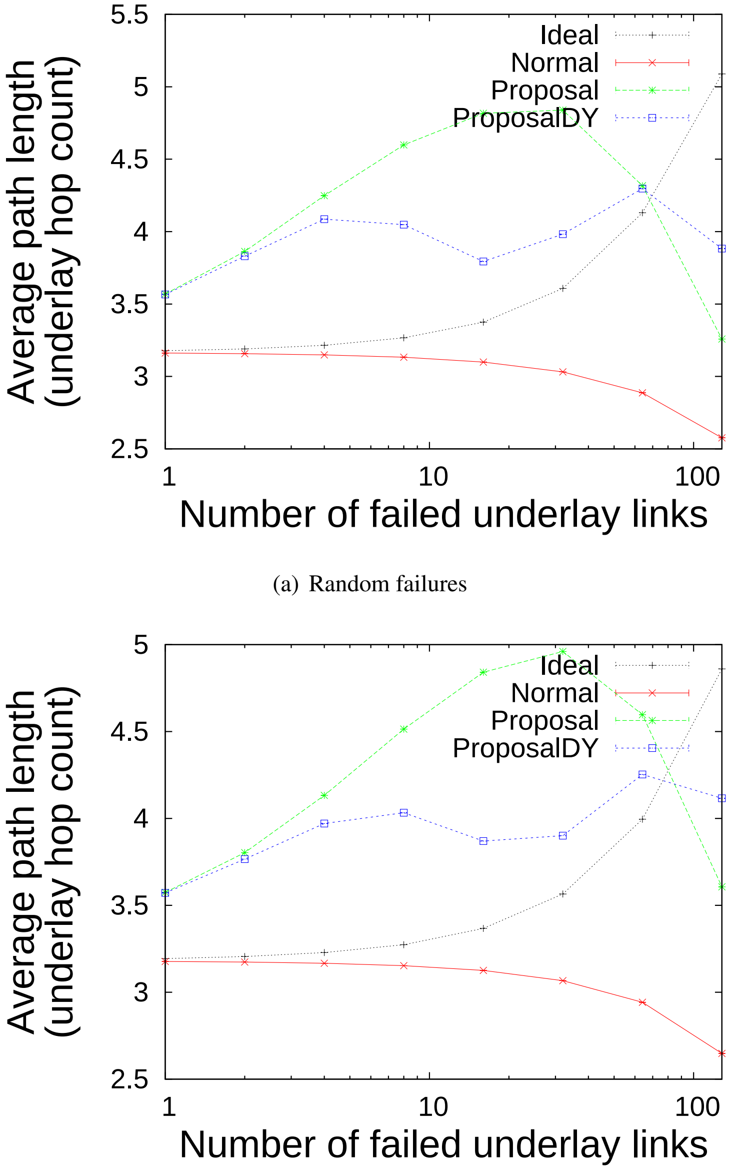


(a) $\beta = 10$

(b) $\beta = 100$

(c) $\beta = 1000$

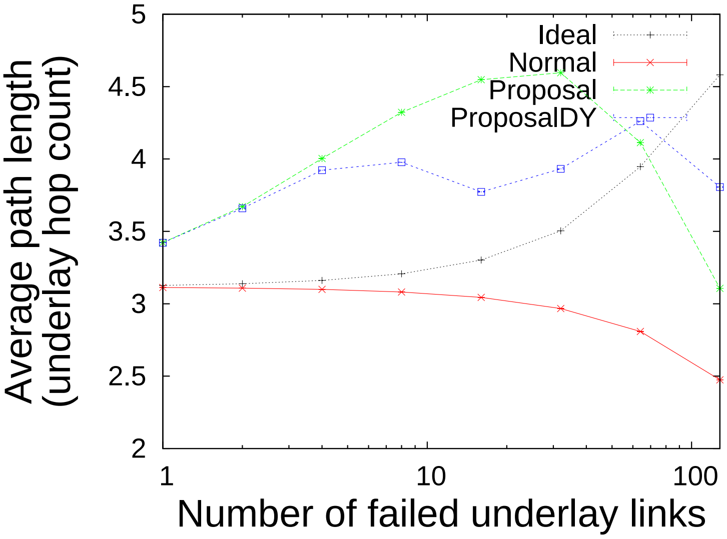
Figure 13: Changes in overlay network reachability with β



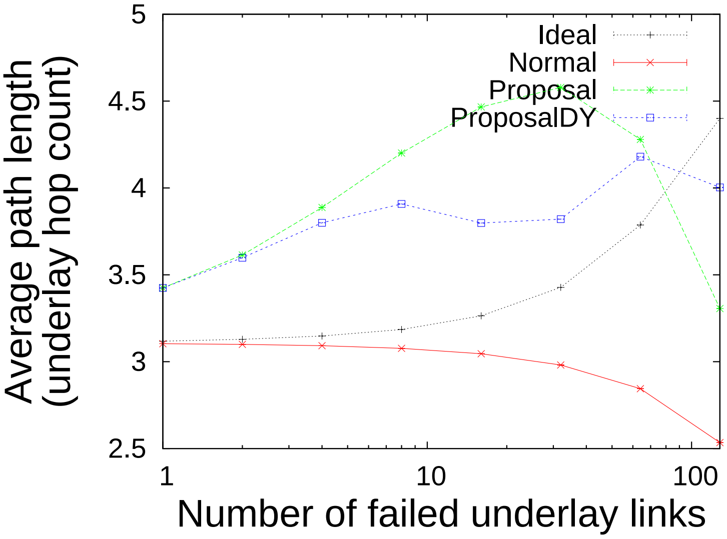
(a) Random failures

(b) Adjacent failures

Figure 14: Average path length in BA-ER network



(a) Random failures



(b) Adjacent failures

Figure 15: Average path length in BA-BA network

in underlay network between the shortest and alternative paths is small compared to small size networks, such as BA topology underlay networks. In other words, the increase of the underlay network size improves the redundancy of the overlay network. The rest of results are similar to the results in Figure 15.

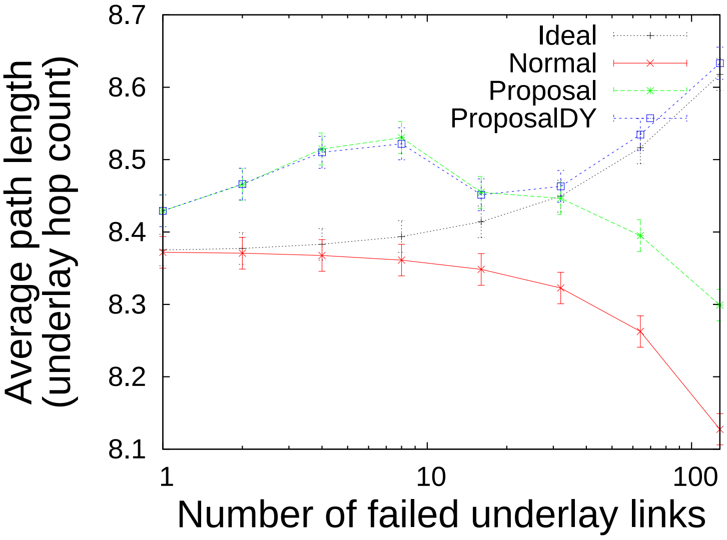
Figure 17 shows the changes in the average path length in the same situation in Figure 11. From this figure, we can see that as increasing overlay node density the average path length of all methods increases. This is due to the increase of the average overlay hop counts since the number of overlay nodes increases.

Figure 18 presents the distribution of path length when the number of failed underlay links changes. We can see that the proposed method has longer-hop paths regardless of the selection method and the number of failed links. This is because that the proposed method utilizes the network topology that has less links than the Ideal and Normal. This characteristic becomes stronger as the number of merged RL pairs increases, and the increase of average path length is caused by this characteristics. In addition, when the number of failed underlay links increases, the distributions in the original topology and the static RL selection concentrate around the shorter length. The reason is that the connectivity of longer-hop paths is often lost due to the decrease of reachability as shown in Figure 9(b).

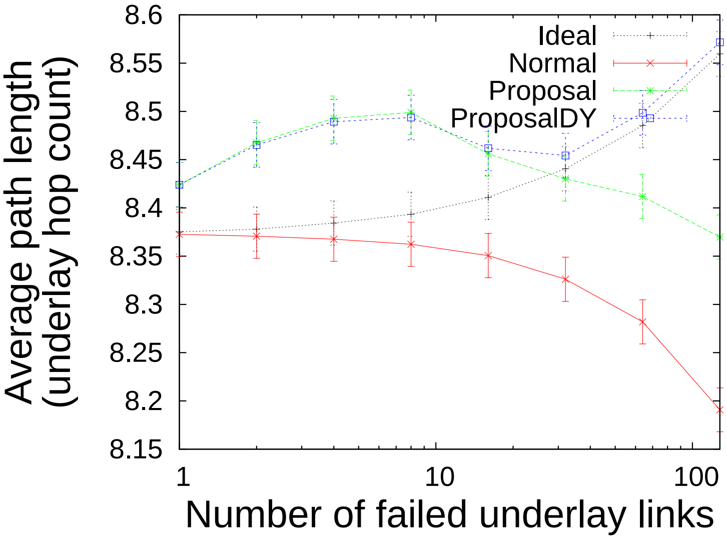
Figures 19 and 20 show the changes in the average path length with various values of α and β, respectively. we can see from these figures, the path length of the proposed method is similar to that of the ideal case when the number of merged RL pairs decreases by setting α to a small number and β to a large number. Therefore, the path length of the proposed method can be suppressed with appropriate values of α and β to obtain better recovery performance. In other words, there is a trade-off relationship between the number of recoverable failure patterns and the path length in the proposed method.

4.4 Performance with network growth

The author finally investigates the performance of the proposed method with network growth. In the following graphs, the author denotes the results of the proposed method with static and dynamic RL selection when RLSet is recalculated against every entry of a new overlay node, as *ProposalRC* and *ProposalRCDY*, respectively. Note that *Proposal* and *ProposalDY* do not reconstruct the RLSet and they utilize the partial reconstruction method described in Subsection 3.4.1.

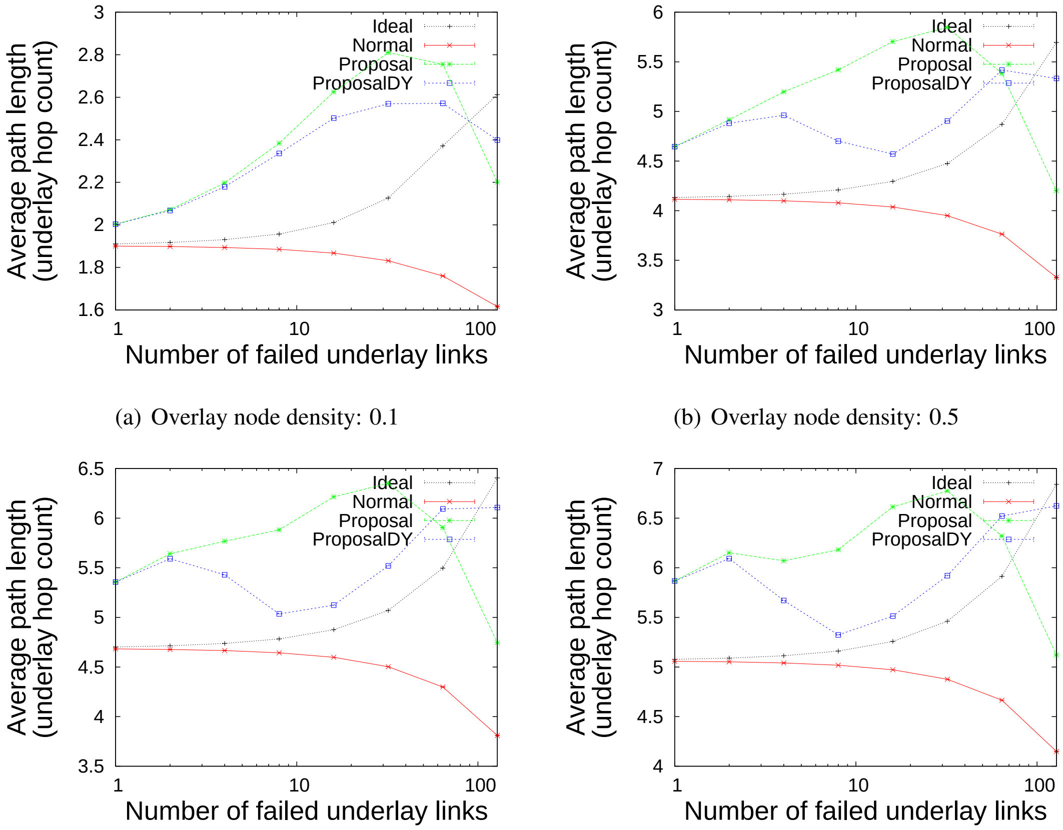


(a) Random failures



(b) Adjacent failures

Figure 16: Average path length in AT&T-BA network



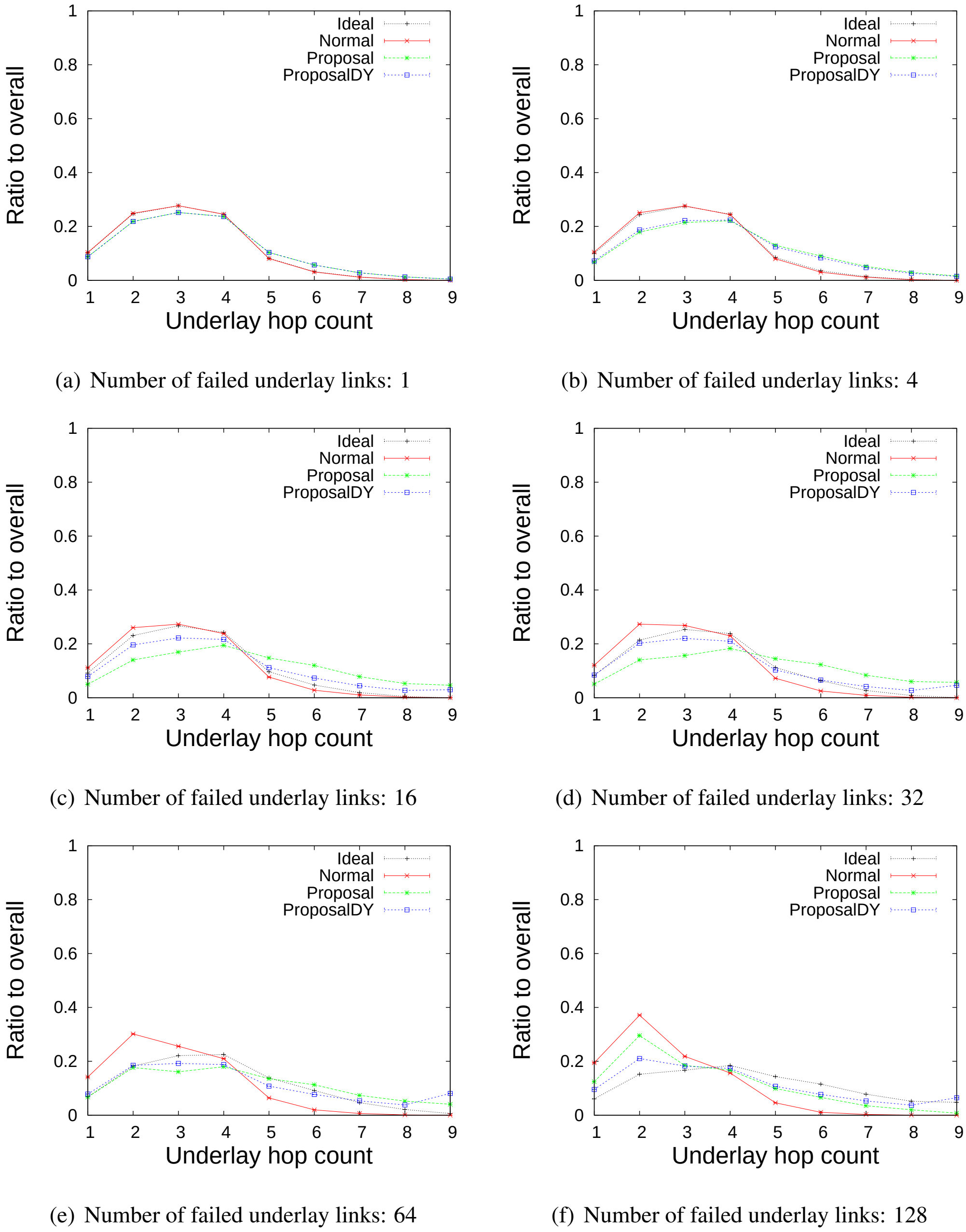
(a) Overlay node density: 0.1

(b) Overlay node density: 0.5

(c) Overlay node density: 0.75

(d) Overlay node density: 1

Figure 17: Changes in average path length with overlay node density



(a) Number of failed underlay links: 1

(b) Number of failed underlay links: 4

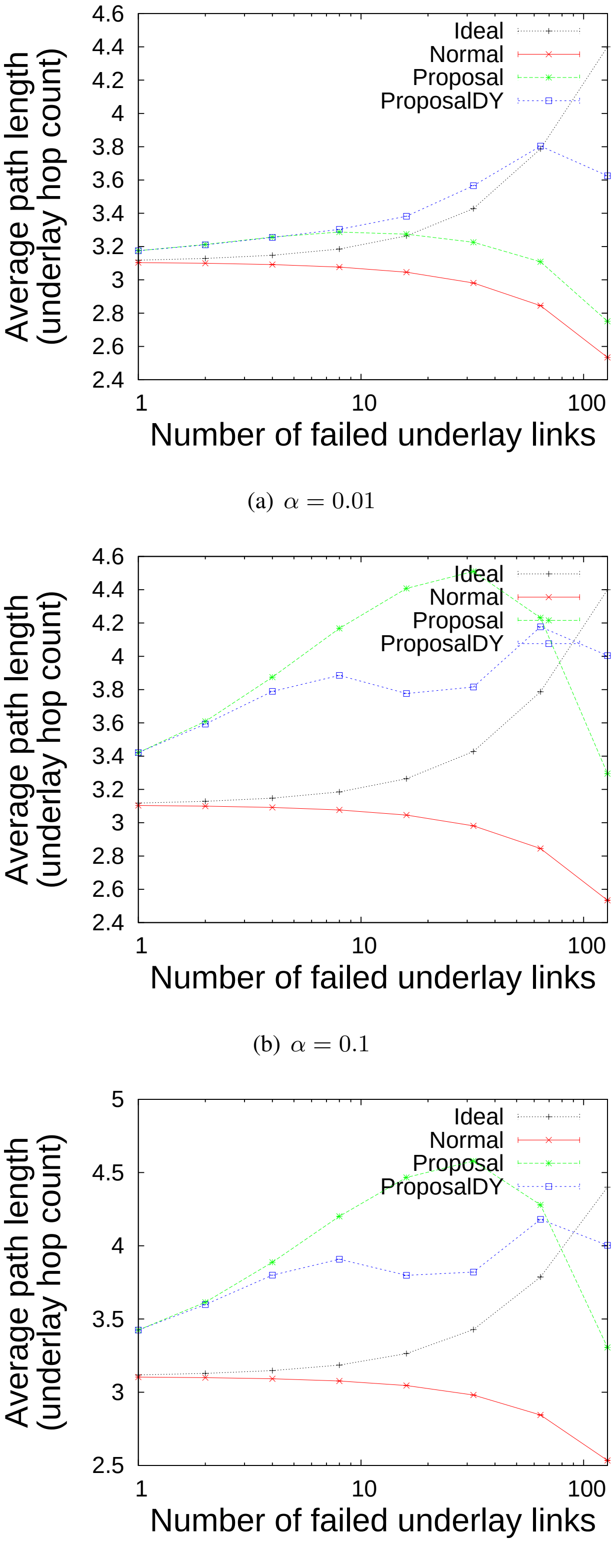
(c) Number of failed underlay links: 16

(d) Number of failed underlay links: 32

(e) Number of failed underlay links: 64

(f) Number of failed underlay links: 128

Figure 18: Distributions of path length against the number of failed underlay links

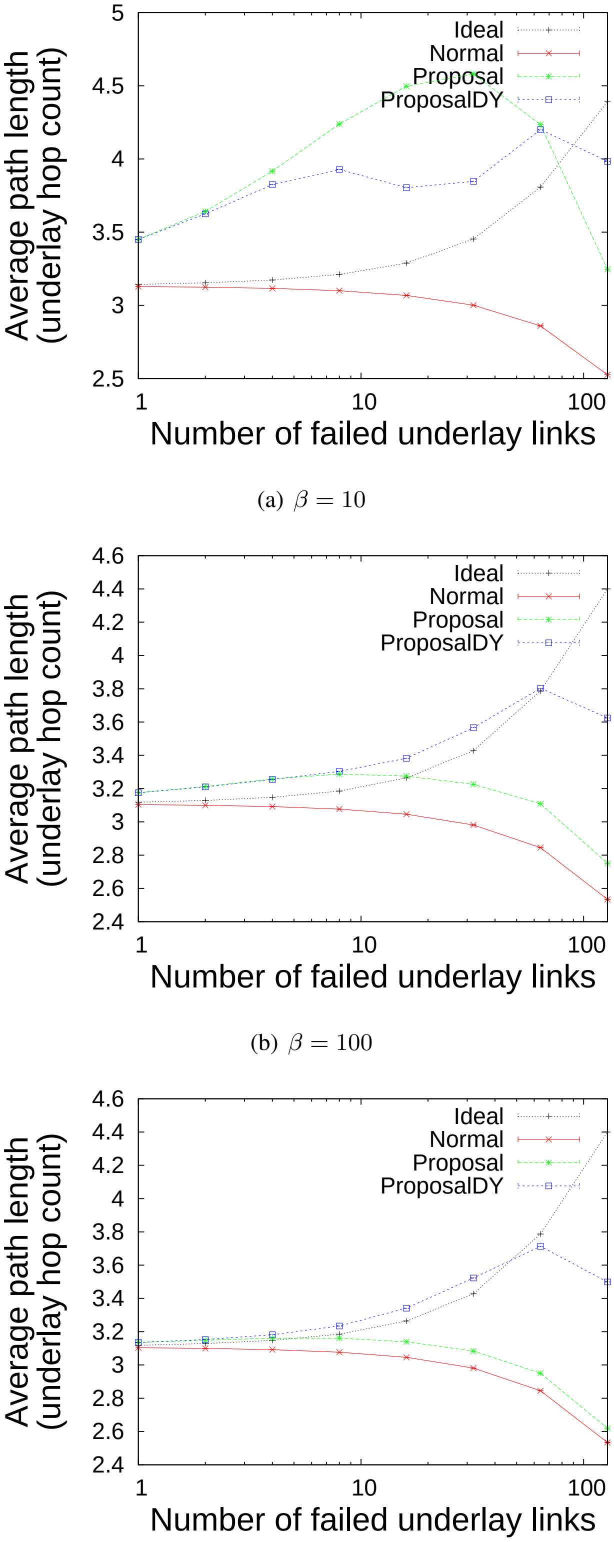


(a) $\alpha = 0.01$

(b) $\alpha = 0.1$

(c) $\alpha = 10$

Figure 19: Changes in average path length with α



(a) $\beta = 10$

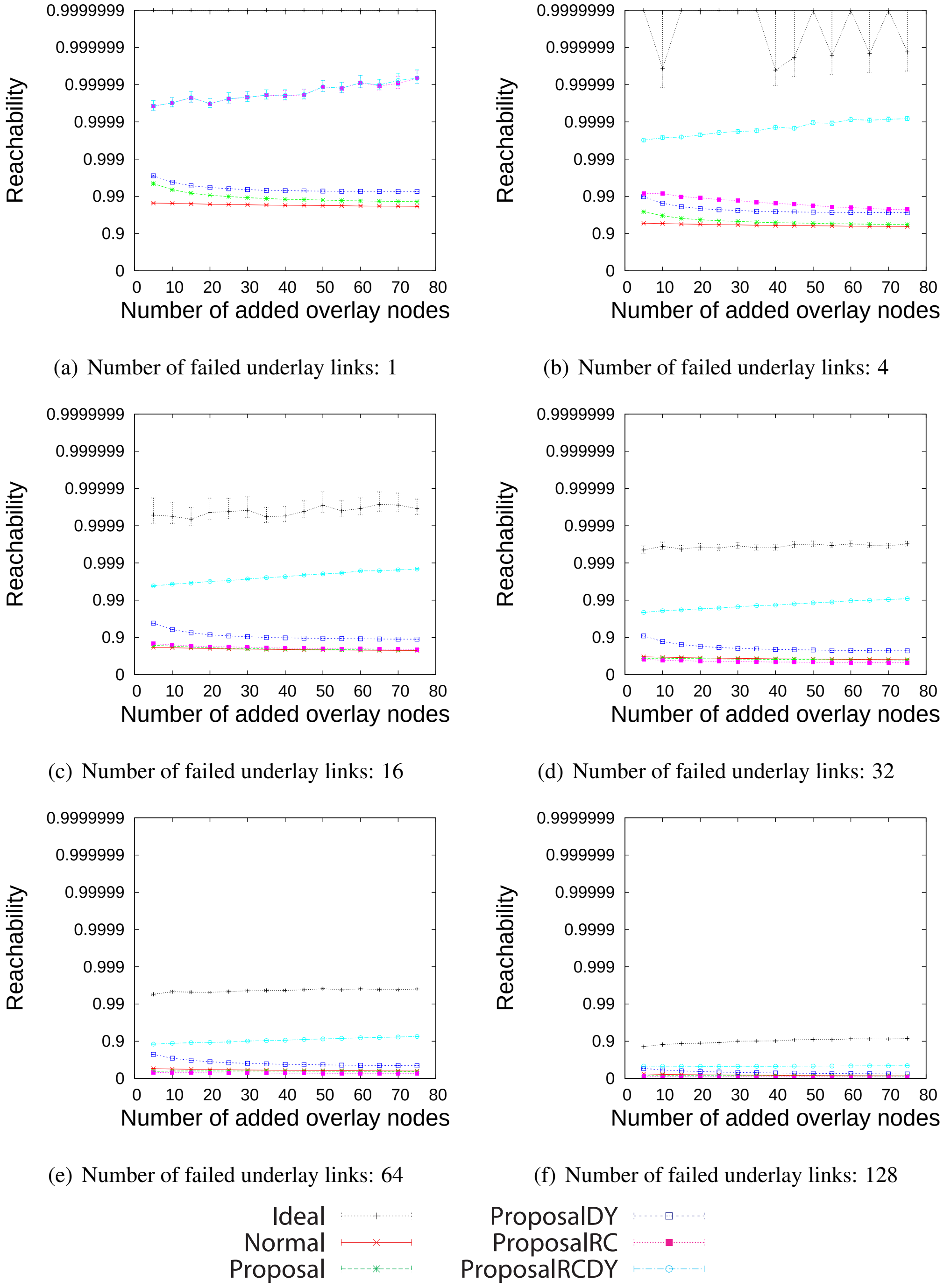
(b) $\beta = 100$

(c) $\beta = 1000$

Figure 20: Changes in average path length with β

Figure 21 shows the changes in the overlay network reachability as a function of the number of overlay nodes added to the overlay network after the calculation of RLSet, by using BA-BA network. From this figure, we can find that the partial reconstruction with dynamic selection improves the overlay network reachability slightly. The reason of this is that by isolating the newly-added overlay nodes with the partial reconstruction, the failures that are not assumed in RLSet before adding the node can be recovered by multiple partially-reconstructed RLs. In addition, we can observe that when the number of failed underlay links is small, the effectiveness of overall reconstruction is large. This is because the overall reconstruction can guarantee recovering from the failures of newly-added overlay links completely. We can also find that as increasing the number of added overlay nodes, the overall reconstructed RLSet improves its reachability. This is due to the increase of the number of RLs similar to in Figure 11.

Figure 22 shows the average path length in the same situations in Figure 21. From this figure, we can see that the average path length of RLSet with overall reconstruction is larger than that with partial reconstruction. The reason is that the number of available overlay links decreases with overall reconstruction since the RLSet with overall reconstruction also isolates newly-added overlay links in various RLs. On the other hand, as increasing the number of added overlay nodes, the average path length of Ideal case increases. This is because the ideal topology keeps connectivity between the overlay nodes in longer distance.



(a) Number of failed underlay links: 1

(b) Number of failed underlay links: 4

(c) Number of failed underlay links: 16

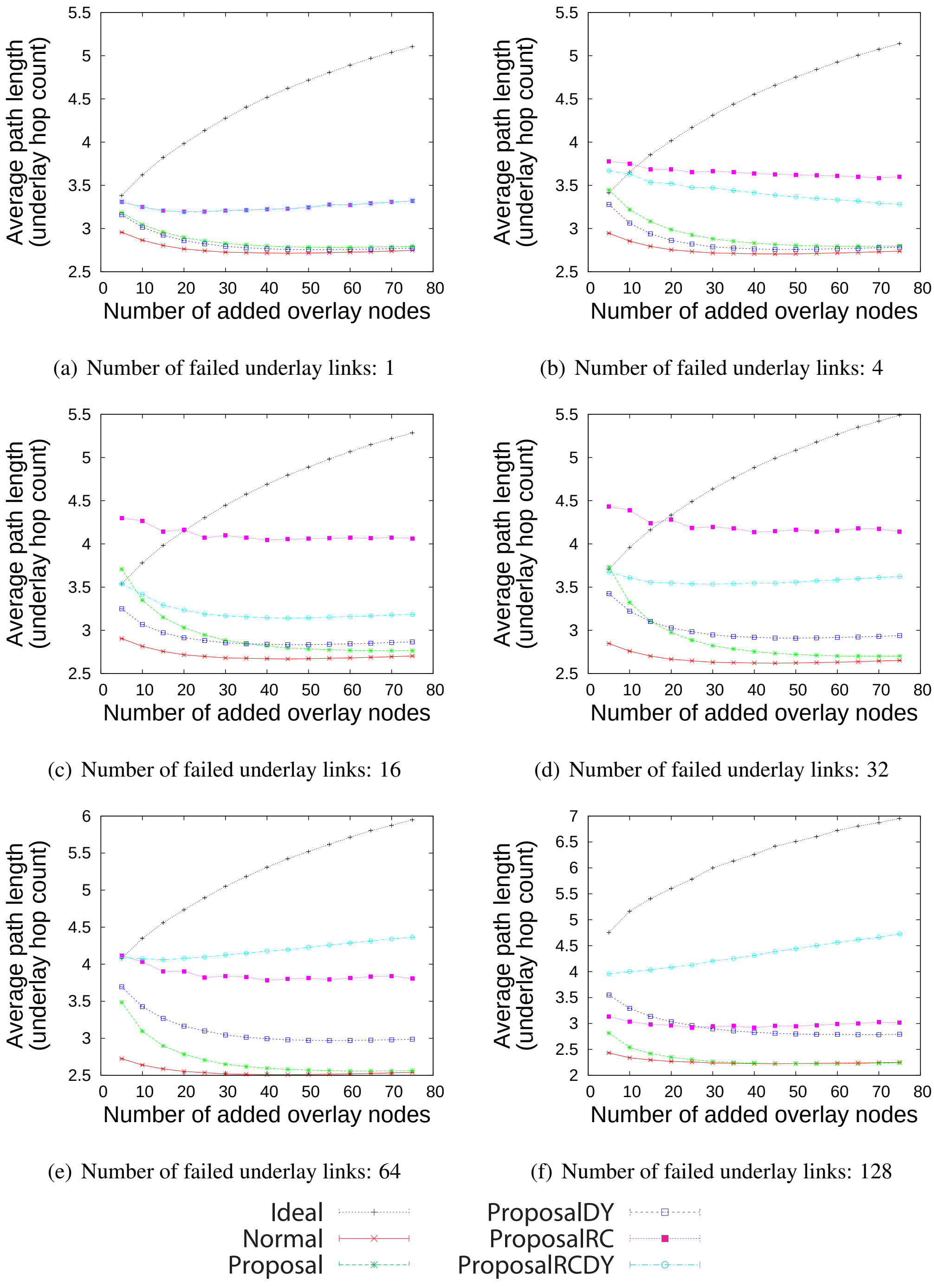
(d) Number of failed underlay links: 32

(e) Number of failed underlay links: 64

(f) Number of failed underlay links: 128

(g) Types of data lines used in above graphs

Figure 21: Overlay network reachability against network growth in BA-BA network



(a) Number of failed underlay links: 1

(b) Number of failed underlay links: 4

(c) Number of failed underlay links: 16

(d) Number of failed underlay links: 32

(e) Number of failed underlay links: 64

(f) Number of failed underlay links: 128

(g) Types of data lines used in above graphs

Figure 22: Average path length against network growth in BA-BA network

5 Conclusions

In this thesis, the author proposed the proactive recovery method for large-scale packet switching networks such as the current Internet, by utilizing overlay networking technique. For multiple simultaneous failures, the proposed method constructs multiple logical network topologies assuming various failure patterns in advance. More precisely, the proposed method is designed to construct the effective topologies for recovering from multiple simultaneous failures, by considering the correlation among overlay links in terms of usage underlay link. In addition, distributed topology construction algorithm of the proposed method extends the range of application, and topology integration algorithm of the proposed method improves recovery performance.

Through the numerical evaluation, the proposed method can improve overlay network reachability from 51 % to 97 % when 25% of network links are go down simultaneously, while it increase average path length only up to 1.6 %.

For future work, the author will try to eliminate assumption that each overlay node knows the complete information of underlay network, meaning that each overlay node knows the information measured by itself. The author also plans to apply the proposed method to unstable networks where nodes are frequently joining and leaving such as wireless ad-hoc networks.

Acknowledgement

I would like to express my sincere gratitude to Prof. Hirotaka Nakano of Osaka University. His comments and suggestions inspired my life not limited to only study.

I would like to give special thanks to Associate Prof. Go Hasegawa of Osaka University. It is impossible that I fulfill my study without his continuous and kind support. My great gratitude to him is ineffable.

I am deeply grateful to Prof. Masayuki Murata of Osaka University, for his excellent guidance and continuous support through my studies in this thesis. My study has been refined by his invaluable comments.

I am also thankful to my thesis committee members, Prof. Koso Murakami, Prof. Makoto Imase, and Prof. Teruo Higashino for their careful reading and constructive comments in completing this thesis.

I would also like to thank Assistant Prof. Yoshiaki Taniguchi of Osaka University, who gave me useful comments and supports. His supports helped me to brush my study up.

I am also indebted to all the members of Graduate School of Information Science and Technology, Osaka University, for their detailed and valuable instructions.

I want to give thanks to Mr. Rintaro Ishii, Mrs. Sayaka Kuriyama, Mr. Masafumi Hashimoto, Mr. Matsuda Kazuhito, Mr. Masakazu Murata, and the other friends in the Department of Information Networking of the Graduate School of Information Science and Technology of Osaka University for their support. My motivation for studies on information networks has been fostered by daily conversation with them.

Finally, I want to thank my parents and brothers, who have given me a loving environment where to develop. I cannot write this thesis without their continuous support.

References

[1] Y. Rekhter and T. Li, “A border gateway protocol 4 (BGP–4),” *RFC 1771*, Mar. 1995.

[2] A. Sahoo, K. Kant, and P. Mohapatra, “Characterization of BGP recovery time under large-scale failures,” in *Proceedings of ICC 2006*, Jun. 2006.

[3] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, “Delayed Internet routing convergence,” in *Proceedings of ACM SIGCOMM 2000*, vol. 9, no. 3, Aug. 2000, pp. 293–306.

[4] B. Zhang, D. Massey, and L. Zhang, “Destination reachability and BGP convergence time,” in *Proceedings of IEEE GLOBECOM 2004*, vol. 3, Apr. 2004, pp. 1383–1389.

[5] P. Chen, W. H. Cho, Z. Duan, and X. Yuan, “Traffic-aware inter-domain routing for improved internet routing stability,” in *Proceedings of the GLOBECOM 2008*, no. 68, Dec. 2008, pp. 2226–2231.

[6] A. Sahooa, K. Kantb, and P. Mohapatra, “BGP convergence delay after multiple simultaneous router failures: Characterization and solutions,” *Computer Communications*, vol. 32, no. 7–10, pp. 1207–1218, May 2009.

[7] A. S. Tanenbaum, *COMPUTER NETWORKS*, 3rd ed. Upper Saddle River, New Jersey 07458: Prentice-Hall, Inc., 1996.

[8] D. Pei, M. Azuma, D. Massey, and L. Zhang, “BGP-RCN: Improving BGP convergence through root cause notification,” UCLA CSD, Tech. Rep. CO80523-1873, Dec. 2004.

[9] A. Sahoo, K. Kant, and P. Mohapatra, “Improving BGP convergence delay for large–scale failures,” in *Proceedings of the DSN’06*, Jun. 2006, pp. 323–332.

[10] Y. Liao, L. Gao, R. Guerin, and Z.-L. Zhang, “Reliable interdomain routing through multiple complementary routing processes,” in *Proceedings of the 2008 ACM CoNEXT Conference*, no. 68, Dec. 2008, pp. 323–332.

[11] “BitTorrent,” available at http://www.bittorrent.com/.

[12] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis, “A framework for ip based virtual private networks,” *RFC 2764*, Feb. 2000.

[13] "Akamai," available at http://www.akamai.com/.

[14] "Globus," available at http://www.globus.org/.

[15] 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.

[16] Z. Xu, M. Mahalingam, and M. Karlsson, "Turning heterogeneity into an advantage in overlay routing," in *Proceedings of IEEE INFOCOM 2003*, vol. 2, Apr. 2003, pp. 1499–1509.

[17] S.-J. Lee, S. Banerjee, P. Sharma, P. Yalagandula, and S. Basu, "Bandwidth-aware routing in overlay networks," in *Proceedings of IEEE INFOCOM 2008*, Apr. 2008, pp. 1732–1740.

[18] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, "Overlay TCP for multi-path routing and congestion control," in *Proceedings of the IMA Workshop on Measurements and Modeling of the Internet*, Jan. 2004.

[19] G. Huston, "Interconnection, peering, and settlements," in *Proceedings of INET'99*, Jun. 1999.

[20] W. Norton, "Internet service providers and peering," available at http://www.equinix.com/pdf/whitepapers/PeeringWP.2.pdf.

[21] ——, "A business case for peering," available at http://www.equinix.com/pdf/whitepapers/Business_case.pdf.

[22] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 733–745, Dec. 2001.

[23] L. Subrmanian, S. Agarwal, J. Rexford, and R. H.Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proceedings of IEEE INFOCOM 2002*, Jun. 2002.

[24] Y. Zhu, C. Dovrolis, and M. Ammar, "Dynamic overlay routing based on available bandwidth estimation: A simulation study," *Computer Networks Journal*, vol. 50, pp. 739–876, Apr. 2006.

[25] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," in *Proceedings of ACM SIGCOMM conference on Internet Measurement*, Oct. 2001, pp. 91–100.

[26] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 29–40, Jan. 2004.

[27] 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.

[28] 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.

[29] O. Klopfenstein, "Robust pre-provisioning of local protection resources in MPLS networks," in *Proceedings of DRCN 2007*, Oct. 2007, pp. 1–7.

[30] 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*, vol. 1, Mar. 2004, pp. 176–186.

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

[32] 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.

[33] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Boston University Representative Internet Topology Generator," available at http://www.cs.bu.edu/brite/index.html#.

[34] A. Hansen, A. Kvalbein, T. Čičić, S. Gjessing, and O. Lysne, "Resilient routing layers for recovery in packet networks," in *Proceedings of the 2005 International Conference on Dependable Systems and Networks*, Jul. 2005, pp. 238–247.

[35] L. Lao, J.-H. Cui, M. Gerla, and S. Chen, "A scalable overlay multicast architecture for large-scale applications," *IEEE Transactions Parallel and Distributed Systems*, vol. 18, no. 4, pp. 449–459, Apr. 2007.

[36] S. El-Ansary, L. O. Alima, P. Brand, and S. Haridi, "Approximation and heuristic algorithms for minimum-delay application-layer multicast trees," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 473–484, Apr. 2007.

[37] N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP topologies with rocketfuel,” in *Proceedings of the 2002 SIGCOMM conference*, Oct. 2002.

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

[39] “Graphviz,” available at http://www.graphviz.org/.

[40] P. Erdös and Rényi, “On the evolution of random graphs,” *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, pp. 17–61, 1960.