Efficiently Adding Link Facilities to Improve Quality Stability at Failures

Noriaki Kamiyama
NTT Service Integration Laboratories
Tokyo 180-8585, Japan
E-mail: kamiyama.noriaki@lab.ntt.co.jp

Abstract— **When a link or node fails, flows are detoured around the failed portion, so the hop count of flows and the link load could change dramatically as a result of the failure. As real-time traffic such as video or voice increases on the Internet, ISPs are required to provide stable quality as well as connectivity at failures. For ISPs, how to effectively improve the stability of these qualities at failures with the minimum investment cost is an important issue, and they need to effectively select a limited number of locations to add link facilities. In this paper, efficient design algorithms to select the locations for adding link facilities are proposed and their effectiveness is evaluated using the actual backbone networks of 36 commercial ISPs.**

I. Introduction

The Internet has mainly been used for e-mail, web browsing, and file exchange. In addition, many users have recently started using Internet-based telephone services instead of the traditional ones on the public switched telephone network (PSTN). More and more people use the Internet as an everyday tool, and the requirement for reliability will grow as the Internet becomes a social infrastructure. In particular, high reliability is required for backbone networks that relay a huge amount of traffic. Under ordinary circumstances, we face the risk of failures on many links or routers on the Internet for various reasons, such as the cutting of fibers, failure of optical transmission devices, misconfiguration of routers, bugs in software, and overloading of routers [1]. Therefore, the Internet is required to be robust enough to sustain connectivity between nodes and maintain stable services in the event of failures, under the assumption that failures are unavoidable.

Open shortest path first (OSPF) is used as a routing protocol within a single autonomous system (AS) on the Internet. Using OSPF, routers autonomously set detour routes for flows[1] when failures occur. Hence, the Internet can sustain high connectivity at times of failure although the hop length of flows or the load of links could increase significantly as a result of the detours. This will degrade the quality of real-time services such as voice or video because the packet network delay increases and the flow throughput decreases. Therefore, when investigating Internet reliability, it is important to evaluate the stability of quality, e.g., the flow hop length or link load, as well as the connectivity at the time of failure.

Although the easiest way to improve the connectivity and stability at failures is to deploy many large-capacity links, the investment cost of facilities and the operational cost increase dramatically. To maintain supremacy in the competitive market, Internet service providers (ISPs) must provide users with high-reliability services at a low cost. Therefore, ISPs need to effectively improve reliability with a limited investment cost. One approach to this problem is to optimally design the network topology while minimizing the total cost from the given traffic demand matrix [2]. However, accurately estimating the traffic demand matrix is difficult, and the optimality degrades as time passes because the traffic demand continues to change. Hence, a realistic solution for ISPs is to partly add links or capacity to the existing networks in accordance with the long-term change in the traffic demand matrix. To implement this adaptive method, we need to effectively pinpoint the locations where links or link capacities should be added by analyzing the main causes of the failure-induced degradation of connectivity or stability.

Although various methods have been proposed to evaluate the importance of each link within the network [3]-[6], all the methods consider only maintaining the connectivity at failures. It has been reported that 70% of unintentional failures, excluding ones caused by maintenance, are single-link failures (denoted *SLF*, hereafter) [1]. Therefore, using publicly available data for the backbone networks of 36 commercial ISPs, we evaluated the connectivity and increase ratio of flow hop length at SLFs [7]. We clarified that almost all 36 networks are highly robust against SLFs from the viewpoint of connectivity. We also analyzed the main causes of increase of flow hop length at SLFs and gave guidelines for effective link addition. However, in Ref. [7], we did not show how to optimally select locations to add links to effectively improve the stability of flow hop length at SLFs.

In this paper, we propose a design method to effectively select locations to add links to satisfy the maximum increase ratio of flow hop length at SLFs below the given bound with the smallest number of additional links. Moreover, we also evaluate the stability of the link load and propose a design method to optimally select links for which capacities are increased to satisfy the maximum increase ratio of link load at SLFs below the given bound with the smallest number of capacity-increased links. In Section II, we describe the properties of the 36 networks evaluated. We summarize the evaluation results of flow hop length stability at SLFs shown in Ref. [7] and also investigate the stability of the link load at SLFs in Section III. In Section IV, we propose design

[1] In this paper, we define a flow as a set of packets transferred between a pair of source and destination nodes.

methods to select locations to add links and select existing links for which capacities are increased to effectively improve the stability at SLFs. In Section V, we numerically show the results of applying the proposed design methods and clarify their effectiveness. Finally, we conclude the paper with a brief summary in Section VI.

II. Network Topologies Used in Evaluation

We evaluated 36 commercial ISP backbone networks whose topologies and link capacities are publicly available at [9]. We list the name, node count n, bidirectional link count m, average node degree $E(d)$, maximum node degree $Max(d)$, and minimum and maximum link capacities of the networks in Table I. Moreover, the maximum node degree is plotted against the average node degree for each of the 36 networks in Fig. 1. On the basis of Fig. 1, we classify the 36 networks into the following 3 groups.

Full mesh Each node connects with almost all the other nodes. $Max(d)$ is close to $E(d)$, and $E(d)$ is large. Here, we define networks with $E(d) \geq Max(d) - 1$ as full-mesh networks. Two networks are categorized as this type.

H&S There are several hub nodes connected to each other, and $Max(d)$ is large. Other nodes connect to one or more hub nodes, so this type of network is called hub and spokes (H&S). Airline topologies are known to form this structure. A node can reach other nodes with a small hop count through hub nodes, so the hop distance between nodes tends to be small. However, the traffic load tends to concentrate on hub nodes. Here, we define networks with $Max(d) \geq 10$ and $E(d) < Max(d) - 1$ as H&S networks. There are 12 networks of this type.

Ladder No hub nodes exist, and the topology structure is a combination of some loops. Both $E(d)$ and $Max(d)$ are small. Although the total link cost is reduced, the hop distance between nodes tends to be large. Freeway topologies are known to form this structure. Here, we define networks with $Max(d) < 10$ and $E(d) < Max(d) - 1$ as ladder networks. We categorized 22 networks as this type.

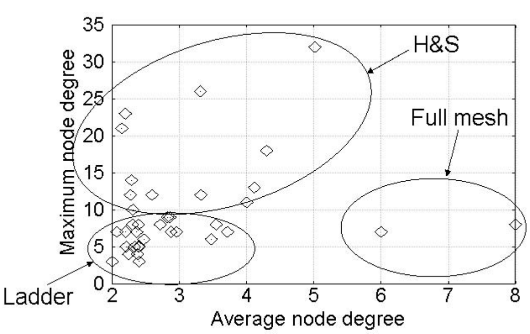


Fig. 1. Average and maximum node degree of 36 networks.

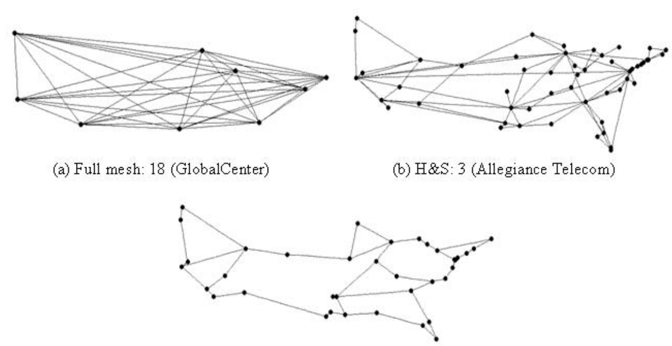
The network type for the 36 networks is also shown in Table I. Topology examples of the three network types are shown in Fig. 2. Although all the three networks shown in Fig. 2 are in

TABLE I

Basic properties of 36 ISP backbone networks.

ID	Name	n	m	Node degree		Link capacity		Type
				Av.	Max.	Min.	Max.	
1	above.net	22	25	2.27	12	45	622	H&S
2	AGIS	82	92	2.24	4	45	822	Ladder
3	Allegiance Telecom	53	88	3.32	12	45	2488	H&S
4	At Home Network	46	55	2.39	5	2488	2488	Ladder
5	AT&T WorldNet	93	154	3.31	26	45	2488	H&S
6	BBN Planet	41	49	2.39	8	45	622	Ladder
7	Cable & Wireless	19	33	3.47	6	45	2488	Ladder
8	CAIS Internet	37	44	2.38	4	45	622	Ladder
9	CompuServe Network Services	16	23	2.88	7	1.5	45	Ladder
10	CRL Network Services	35	50	2.86	9	45	45	Ladder
11	DataXchange Network Inc.	8	24	6.00	7	45	155	Full mesh
12	EPOCH Networks Inc.	29	30	2.07	7	45	45	Ladder
13	EUnet	28	30	2.14	21	45	45	H&S
14	Exodus	14	19	2.71	8	155	622	Ladder
15	Genuity	48	53	2.21	5	45	45	Ladder
16	GeoNet Communications Inc.	13	15	2.31	10	1.5	45	H&S
17	GetNet International	5	6	2.40	3	45	45	Ladder
18	GlobalCenter	9	36	8.00	8	45	45	Full mesh
19	GoodNet	27	58	4.30	18	45	155	H&S
20	IDT Corp	15	18	2.40	5	45	45	Ladder
21	ipf.net	5	5	2.00	3	1.5	4	Ladder
22	iSTAR Internet Inc.	20	22	2.20	7	1.5	45	Ladder
23	MindSpring	41	45	2.20	23	1.5	45	H&S
24	Nap.Net, LLC	6	7	2.33	5	45	45	Ladder
25	Netrail Incorporated	17	21	2.47	6	45	45	Ladder
26	PSINet	78	110	2.82	9	2	155	Ladder
27	Qwest	14	26	3.71	7	622	2488	Ladder
28	RNP	27	35	2.59	12	1	25	H&S
29	Savvis Communications	28	56	4.00	11	45	155	H&S
30	ServInt Internet Services	23	34	2.96	7	155	155	Ladder
31	Sprint	22	39	3.55	8	155	9952	Ladder
32	Telstra Internet	21	24	2.29	14	2	106	H&S
33	UUNET	128	321	5.02	32	1	10000	H&S
34	Verio	35	72	4.11	13	45	622	H&S
35	VisiNet	11	13	2.36	7	45	45	Ladder
36	XO Communications	33	38	2.30	8	155	10000	Ladder

the USA, we confirmed that the topological structures of the three network types are completely different from each other.



(a) Full mesh: 18 (GlobalCenter)

(b) H&S: 3 (Allegiance Telecom)

(c) Ladder: 8 (CAIS Internet)

Fig. 2. Examples of network topologies.

III. Evaluation of Quality Stability at SLFs

In this section, we investigate the quality stability of the 36 commercial ISP backbone networks at SLFs. First, we investigate the increase ratio of flow hop length at SLFs shown in Ref. [7]. Second, we newly evaluate the increase ratio of link load at SLFs. These properties depend on the routing method. In this paper, we assume OSPF and consider two cases of link weight: *inv.cap*, i.e., setting values inversely proportional to the link capacity (Cisco-recommended setting [10]), and *min.hop*, i.e., setting identical values for all links. In the case of min.hop, paths with minimum hop length are always selected. When multiple paths with the smallest cost exist between a pair of source and destination nodes, we assume that traffic between these two nodes is evenly split among all the smallest-cost paths.

A. Increase Ratio of Flow Hop Length

For each network, we define h, the average hop length of flows, as

$$h = \frac{2}{n(n-1)} \sum_{s,d \in \boldsymbol{V}} h_{sd}, \qquad h_{sd} = \frac{1}{\sigma_{sd}} \sum_{p \in \boldsymbol{P}_{sd}} h_{sd,p}, \tag{1}$$

where $\boldsymbol{V}$ is the set of all nodes, h_{sd} is the average hop length of the flow from node s to node d (denoted flow sd hereafter), σ_{sd} is the number of paths of flow sd, $\boldsymbol{P}_{sd}$ is the set of paths of flow sd, and $h_{sd,p}$ is the hop length of path p of flow sd.

Let R_f denote the average ratio of the number of SLF patterns affecting each flow to the total link count m. R_f in almost all networks was smaller than about 0.15 [7], so we consider only SLF patterns affecting each flow to evaluate the average increase in flow hop length at SLFs. In other words, we define h'_{sd}, the average hop length of flow sd at SLFs, as

$$h'_{sd} = \frac{1}{m_{sd}} \sum_{l \in \boldsymbol{E}_{sd}} h_{sd,l}, \quad h_{sd,l} = \frac{1}{\sigma_{sd,l}} \sum_{p \in \boldsymbol{P}_{sd,l}} h_{sd,l,p}, \tag{2}$$

where m_{sd} is the number of links whose SLFs affect flow sd, $\boldsymbol{E}_{sd}$ is the set of these links, $h_{sd,l}$ is the average hop length of flow sd at the SLF of link l, $\sigma_{sd,l}$ is the number of paths provided between nodes s and d at the SLF of link l, $\boldsymbol{P}_{sd,l}$ is the set of these paths, and $h_{sd,l,p}$ is the hop length of path p belonging to $\boldsymbol{P}_{sd,l}$. We define ξ, the average increase ratio of flow hop length, as

$$\xi = \frac{2}{n(n-1)} \sum_{s,d \in \boldsymbol{V}} \frac{h'_{sd}}{h_{sd}}. \tag{3}$$

Moreover, we define ξ_{max}, the maximum increase ratio of flow hop length, as

$$\xi_{max} = \max_{s,d \in \boldsymbol{V}} \left\{ \max_{l \in \boldsymbol{E}_{sd}} \frac{h_{sd,l}}{h_{min,sd}} \right\}, \tag{4}$$

where $h_{min,sd}$ is the minimum hop length of flow sd in the normal state.

Figure 3 shows scatter diagrams of ξ and ξ_{max} for the 36 networks in (a) inv.cap and (b) min.hop. Both ξ and ξ_{max} in inv.cap were larger than those in min.hop in some networks. Because the flow length of ladder networks was large, the increase ratio of flow hop length also tended to be large in these networks. In particular, the ξ_{max} of 5 networks, 2, 4, 8, 15, and 26, were extremely large.

B. Increase Ratio of Link Load

We simply use the number of flows taking each link l to evaluate the load of link l. Similar to Betweeness [8], we define v_l, the average number of flows taking link l, as

$$v_l = \frac{2}{n(n-1)} \sum_{s,d \in \boldsymbol{V}} \frac{\sigma_{sd}(l)}{\sigma_{sd}}, \tag{5}$$

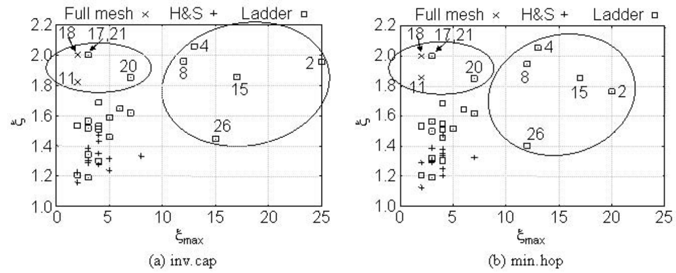


Fig. 3. Flow length increase ratio at SLF.

and define v, the average number of flows on each link, and v_{max}, the maximum number of flows on a link, as the average and maximum value of v_l among m links.

Let R_l denote the average number of SLF patterns at which the number of flows on each link increases divided by m. In almost all the networks, R_l is less than 0.5, so we consider only SLF patterns at which the number of flows on each link increases to evaluate the average increase in flow count on each link at SLFs. We define ε, the normalized average increase ratio of flow count on each link, as

$$\varepsilon = \frac{1}{m_\varepsilon(n-1)} \sum_{l \in \boldsymbol{E}_\varepsilon} \varepsilon_l, \quad \varepsilon_l = \frac{1}{m_{\varepsilon,l}} \sum_{e \in \boldsymbol{E}_{\varepsilon,l}} \frac{v'_{l,e}}{v_l}, \tag{6}$$

where ε_l is the average increase ratio of flow count on link l at SLFs, $v'_{l,e}$ is the flow count on link l at the SLF of link e, $m_{\varepsilon,l}$ is the number of SLF patterns at which $v'_{l,e} > v_l$, $\boldsymbol{E}_{\varepsilon,l}$ is SLF link set $v'_{l,e} > v_l$, m_ε is the number of links with $m_{\varepsilon,l} > 0$, and $\boldsymbol{E}_\varepsilon$ is the link set with $m_{\varepsilon,l} > 0$.

Moreover, we define ε_{max}, the normalized maximum increase ratio of flow count on each link at SLF, as

$$\varepsilon_{max} = \frac{1}{n-1} \max_{l \in \boldsymbol{E}} \varepsilon_{max,l}, \quad \varepsilon_{max,l} = \max_{e \in \boldsymbol{E}_{\varepsilon,l}} \frac{v'_{l,e}}{v_l}, \tag{7}$$

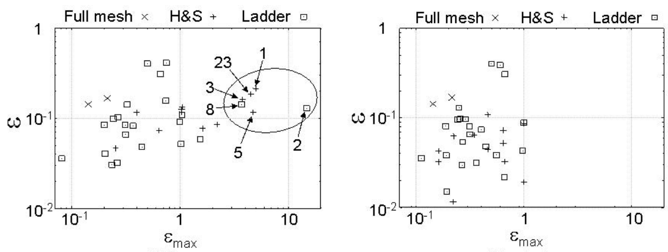
where $\boldsymbol{E}$ is the set of all links and $\varepsilon_{max,l}$ is the maximum increase ratio of flow count on link l at SLFs. When calculating ε and ε_{max}, we set $v_l = 1$ for links with $v_l = 0$. Because there are $n-1$ flows originating from or directed to one specific node, the increase ratio of flow count on each link at SLFs tends to be larger as n increases. To evaluate the increase ratio of flow count fairly over networks with a wide range of n, we evaluate the normalized value divided by $n-1$.

Figure 4 plots scatter diagrams of ε and ε_{max} for the 36 networks in (a) inv.cap and (b) min.hop. The difference of flow count on each link increases in inv.cap, so the increase ratio of flow count at SLFs is larger in inv.cap. In six of the networks, ε_{max} is especially large in inv.cap.

IV. Selection Method to Add Link Facilities

A. Location Selection for Link Addition

In this section, we propose a method of selecting locations to add links to effectively reduce the maximum and average increase ratio of flow hop length, ξ_{max} and ξ, at SLFs. We first show the concept of the maximum shortest loop.



(a) inv.cap (b) min.hop

Fig. 4. Flow count increase ratio at SLF.

1) Definition of Maximum Shortest Loop: The concept of the *maximum shortest loop* was originally proposed in Ref. [7]. Here, we define it more precisely. When path p can be divided into two partial parts, p_1 and p_2, we describe it as $p = p_1 + p_2$. When path p can be divided into three or more partial paths, we describe it in the same way. Let $w(p)$ denote the cost of path p, and let $w_{sd}^{(min,1)}$ and $w_{sd}^{(min,2)}$ be the cost of the smallest path and the second-smallest path of flow sd, respectively. Flow examples between nodes A and B are shown in Fig. 5. Assuming that $w(p_a) = w(p_b)$, $w(p_d) = w(p_e)$, $w(p_f) = w(p_g)$, and $w(p_a + p_c + p_d) < w(p_f)$, there are four paths, $p_a + p_c + p_d$, $p_a + p_c + p_e$, $p_b + p_c + p_d$, and $p_b + p_c + p_e$, with the smallest cost of $w_{AB}^{(min,1)}$ for flow AB. These four paths are used in the normal state. We define $\boldsymbol{\Omega}_{sd}$, the critical path set of flow sd, as the set of partial paths that all the paths with the cost of $w_{sd}^{(min,1)}$ take. In the example shown in Fig. 5, path p_c is the critical path of flow AB. There exist end node pairs that have no critical paths.

When failures occur on the critical path, none of the paths with the smallest cost $w_{sd}^{(min,1)}$, i.e., ones used in the normal state, can be used. Let $\boldsymbol{T}(-\boldsymbol{\Omega}_{sd})$ denote the topology generated by removing all the links included in $\boldsymbol{\Omega}_{sd}$ from the original topology $\boldsymbol{T}$. The smallest-cost paths between nodes s and d on $\boldsymbol{T}(-\boldsymbol{\Omega}_{sd})$ with the path cost of $w_{sd}^{(min,2)}$ are detour paths for flow sd when none of the primary paths can be used. In the example shown in Fig. 5, paths p_f and p_g are detour paths at failures on the critical path p_c.

We define the *shortest loop* for nodes A and B, denoted SL_{AB}, as the minimum cost path among ones originating from node A that travels via node B and returns to node A without taking the same link more than once, including in the reverse direction. If there is no shortest loop for two nodes A and B, there exists at least one link that any path must take between node A and B. Thus, the connectivity between nodes A and B is certainly lost at an SLF of this critical link, and flow AB is not considered when we evaluate the stability at SLFs. Therefore, we consider only node pairs for which a shortest loop exists. On SL_{AB}, there are two paths that do not share any links between nodes A and B. If flow sd has no critical path, paths with the cost of $w_{sd}^{(min,2)}$ are never used at any SLFs and the increase ratio of flow hop length at SLFs is always unity because the shortest loop for nodes s and d is a combination of any two paths sharing no links with the cost of $w_{sd}^{(min,1)}$.

If flow sd has one or more critical paths, on the other hand, the shortest loop for nodes s and d is a combination of any path with $w_{sd}^{(min,1)}$ and any path with $w_{sd}^{(min,2)}$. Therefore, the increase ratio of flow hop length at SLFs is $\max\ h(p_2)/\min\ h(p_1)$ in this case, where $h(p)$ is the hop length of path p and p_1 and p_2 are $p_1 \in \boldsymbol{P}_{sd}$ and $p_2 \in \boldsymbol{P}_{sd}(-\boldsymbol{\Omega}_{sd})$, respectively. $\boldsymbol{P}_{sd}(-\boldsymbol{\Omega}_{sd})$ is the set of minimum-cost paths of flow sd on $\boldsymbol{T}(-\boldsymbol{\Omega}_{sd})$.

We also define the *maximum shortest loop* as the shortest loop having the maximum value of $\max\ h(p_2)/\min\ h(p_1)$; the SLF of a link on the critical path of the maximum shortest loop gives ξ_{max}.

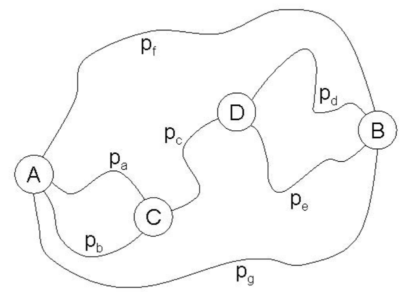


Fig. 5. Example of shortest loop.

2) Design Algorithm: From the given network topology, we aim to satisfy $\xi_{max} \leq \alpha_\xi$, where α_ξ is a given parameter. In other words, we try to bound the maximum increase ratio of flow hop length at any SLFs below the given value α_ξ. We propose a greedy-based algorithm to select the smallest number of locations to add links to satisfy $\xi_{max} \leq \alpha_\xi$ from the given network topology. The principal of the proposed algorithm called *Algorithm A* is repeatedly finding the maximum shortest loop and adding links to effectively reduce the hop length of the found maximum shortest loops. Here, we describe Algorithm A.

Algorithm A:

- A-1 For each node pair sd, derive $\boldsymbol{P}_{sd}$, the path set used in the normal state, and $\boldsymbol{P}_{sd}(-\boldsymbol{\Omega}_{sd})$, the path set used at SLFs on the critical path.
- A-2 Select sd pairs in descending order of the value of $\max\ h(p_2)/\min\ h(p_1)$, and let s^* and d^* denote the selected sd pair.
- A-3 Let $\boldsymbol{T}(+e_{ab,c})$ denote the topology generated by adding a link with capacity c between nodes a and b, where c is selected from $\boldsymbol{C}$, the link capacity set of existing links, and nodes a and b are any disjoint nodes on the path p with the maximum hop length among paths included in $\boldsymbol{P}_{s^*d^*}(-\boldsymbol{\Omega}_{s^*d^*})$. Derive ξ_{max} and ξ for each pair of a, b, and c, and select a^*, b^*, and c^* on which ξ_{max} is minimized among all possible pairs of a, b, and c. When multiple pairs of a, b, and c have the smallest ξ_{max}, select one with the smallest ξ.
- A-4 If ξ_{max} increases (or ξ does not decrease while ξ_{max} is constant), go back to step A-2 and select the next sd pair. Else, add a link with the capacity of c^*

between nodes a^* and b^*, and go back to step A-1 if $\xi_{max} > \alpha_\xi$. Otherwise, the procedure is completed.

In the case of min.hop, link capacities do not affect routing, so we set the link capacity of added links to the median of $\boldsymbol{C}$. Next, let us investigate the required calculation time of Algorithm A. In A-1, we need to obtain the minimum-cost path on $\boldsymbol{P}_{sd}(-\boldsymbol{\Omega}_{sd})$ for each sd pair using the Dijkstra algorithm. Because the calculation time required in the Dijkstra algorithm is $O(n^2)$ for node count n, the required calculation time in A-1 is $O(n^4)$. The amount of calculation in A-2 is $O(1)$. To derive ξ_{max} and ξ for each pair of a, b, and c, we need $O(mn^3)$ time. Hence, the required calculation time in A-3 is $O(h_{max}^2 n_c mn^3)$, where h_{max} is the maximum hop length of paths included in $\boldsymbol{P}_{s^*d^*}(-\boldsymbol{\Omega}_{s^*d^*})$ and n_c is the cardinality of link capacities. The correlation between n_c and m is weak, and n_c is far smaller than m. Moreover, although h_{max} is far smaller than n, h_{max} has a positive correlation with n. Therefore, we can bound the upper limit of the required calculation time in A-3 by $O(mn^5)$, and the total calculation time to find each location to add a link in Algorithm A is $O(mn^5)$.

B. Link Selection for Capacity Increase

Next, let us consider selecting links for which capacities are increased to effectively improve the stability of the flow count on each link at SLFs. From the given network topology, we aim to satisfy $\epsilon_{max} \leq \alpha_\epsilon$, where α_ϵ is a given parameter. In other words, we try to bound the maximum increase ratio of flow count on each link at any SLFs below the given value α_ϵ. We propose using a greedy-based algorithm to select the smallest number of links for which capacities are increased to satisfy $\epsilon_{max} \leq \alpha_\epsilon$ from the given network topology. Because link capacities do not affect routing in min.hop, we consider only the case of inv.cap. If we increase the capacity of links with a large value of $\varepsilon_{max,l}$, we can dramatically decrease the difference of the flow count on each link at the normal state with that at SLFs, and we can expect a large reduction of ε_{max} and ε. Thus, the principal of the proposed algorithm called *Algorithm B* is to repeatedly try to increase the capacities of links with larger values of $\varepsilon_{max,l}$. Here, we describe Algorithm B.

Algorithm B:

- B-1 Derive $\varepsilon_{max,l}$ for each link l.
- B-2 Select link l in descending order of $\varepsilon_{max,l}$, and let l^* denote the selected link.
- B-3 Derive ε_{max} and ε when increasing c_{l^*}, the capacity of selected link l^*, to any c where $c \in \boldsymbol{C}$ and $c > c_{l^*}$. Let c^* denote the capacity candidate of l^* that gives the minimum value of ε_{max}. When multiple capacity candidates give the minimum ε_{max}, the one giving the minimum value of ε is selected.
- B-4 If ε_{max} increases (or ε does not decrease while ε_{max} is constant), go back to step B-2 and select the next target link. Otherwise, increase c_{l^*} to c^* and go back to step B-1 if $\varepsilon_{max} > \alpha_\varepsilon$. Otherwise, the procedure is completed.

Let us investigate the required calculation time of Algorithm B. By updating the sd pair set taking link l in the normal state, we can bound the required calculation time in B-1 to $O(mn_s n^2)$, where n_s is the number of source nodes s for which any paths take link l. n_s is bounded by n, so the calculation time in B-1 is bounded by $O(mn^3)$. To derive ε_{max} and ε for each pair of l and c, we need $O(mn^3)$ time, so the required time in B-3 is $O(n_c m^2 n^3) \simeq O(m^2 n^3)$. Therefore, the total required time to select one link for which capacity is increased is $O(m^2 n^3)$.

C. Combined Algorithm

If we independently apply Algorithm A or Algorithm B to the given network, we cannot expect to improve the stability of a quality that is not considered at the SLFs, i.e., the increase of flow count on each link in the former case or the increase of flow hop length in the latter case. In some cases, these qualities could be degraded by independently applying these proposed algorithms. When redesigning the existing network, simultaneously giving the target level for the two stability qualities at SLFs is desirable. In this section, we propose an algorithm to effectively and simultaneously select both the locations for link addition and the existing links for which to capacities are increased to satisfy both $\xi_{max} \leq \alpha_\xi$ and $\varepsilon_{max} \leq \alpha_\varepsilon$ from the given α_ξ and α_ε.

Let $\zeta_{link,1}$ and $\zeta_{link,2}$ denote the improve ratio of ξ_{max} and ξ, respectively, when adding one link at the location selected by steps A-1 to A-3 in Algorithm A. Moreover, let $\zeta_{cap,1}$ and $\zeta_{cap,2}$ denote the improve ratio of ε_{max} and ε, respectively, when increasing the capacity of one existing link selected by steps B-1 to B-3 in Algorithm B. We have

$$\begin{aligned}
\zeta_{link,1} &= (\xi_{max} - \xi'_{max})/\xi_{max}, \\
\zeta_{link,2} &= (\xi - \xi')/\xi, \\
\zeta_{cap,1} &= (\varepsilon_{max} - \varepsilon'_{max})/\varepsilon_{max}, \\
\zeta_{cap,2} &= (\varepsilon - \varepsilon')/\varepsilon,
\end{aligned}$$

where ξ_{max}, ξ, ε_{max}, and ε are evaluation values before adding a link facility at the selected position, and ξ'_{max}, ξ', ε'_{max}, and ε' are evaluation values after adding a link facility at the selected position.

When $\zeta_{link,1} > \zeta_{cap,1}$ (or $\zeta_{link,1} < \zeta_{cap,1}$), we add one link at the location selected by Algorithm A (or increase the capacity of the link selected by Algorithm B). When $\zeta_{link,1} = \zeta_{cap,1}$, we add one link if $\zeta_{link,2} > \zeta_{cap,2}$ or increase link capacity if $\zeta_{link,2} < \zeta_{cap,2}$, in the same way. We summarize the algorithm called *Algorithm C* as follows.

Algorithm C:

- C-1 Select one candidate location for link addition by executing steps A-1 to A-3 of Algorithm A.
- C-2 Select one candidate link for capacity increase by executing steps B-1 to B-3 of Algorithm B.
- C-3 Choose the action plan on the basis of the following conditions
  (a) Add one link at the location selected by C-1 if all three following conditions are satisfied: (i) $\xi_{max} >$

α_ξ, (ii) both $\zeta_{link,1}$ and $\zeta_{link,2}$ are equal to or greater than zero or at least one is greater than zero, and (iii) $\zeta_{link,1} > \zeta_{cap,1}$, or $\zeta_{link,1} = \zeta_{cap,1}$ and $\zeta_{link,2} > \zeta_{cap,2}$.
(b) Increase the capacity of the link selected by C-2 if all three following conditions are satisfied: (i) $\varepsilon_{max} > \alpha_\varepsilon$, (ii) both $\zeta_{cap,1}$ and $\zeta_{cap,2}$ are equal to or greater than zero or at least one is greater than zero, and (iii) $\zeta_{link,1} < \zeta_{cap,1}$, or $\zeta_{link,1} = \zeta_{cap,1}$ and $\zeta_{link,2} < \zeta_{cap,2}$.

C-4 If one link is added or capacity is increased on one link, go to C-1. Otherwise, the algorithm is completed.

The calculation time required in C-1 dominates the entire procedure, so the total calculation time required to find one location to add a facility is $O(mn^5)$.

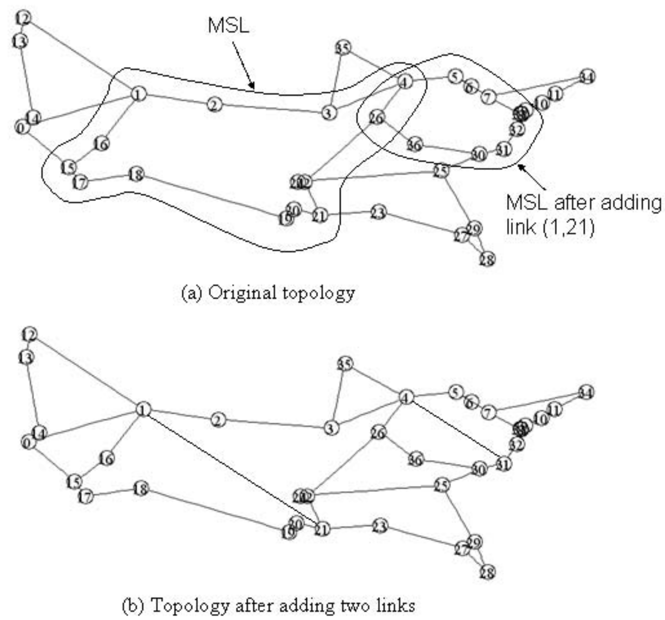
V. Numerical Evaluation

In this section, we apply the three proposed algorithms to the 36 networks described in Section II and clarify their effectiveness. We show the results only when inv.cap is used in link weight settings.

A. Results of Applying Algorithm A

First, we show the results of Algorithm A. To see the tendency of locations where links were added, we show the original topology of network 8 (CAIS Internet) in Fig. 6(a) and the topology after two links were added to this network in Fig. 6(b). The maximum shortest loop (MSL) of this network is 1-2-3-4-26-22-21-20-19-18-17-15-16-1 with 13 hops, and we confirm that a link was first added between nodes 1 and 21, which halved the hop length of the MSL. We also confirm that nodes with high degree were selected for link addition because adding a link between high degree nodes effectively reduces the average increase ratio of flow hop length, ξ. After a link was added between nodes 1 and 21, the closed loop 4-5-7-8-32-31-30-25-22-26-4 became the MSL and the maximum increase ratio of flow hop length, ξ_{max}, was decreased to 9. The second location for link addition was between nodes 4 and 31, and we confirm that a link was also added at the location where the hop length of the MSL was halved.

To clarify the effectiveness of Algorithm A, we compare it with the case called *Optimum A*, in which the strictly optimum locations to add links were selected. Optimum A checked all the possible patterns to add links and found the minimum number of locations to add links satisfying $\xi_{max} \leq \alpha_\xi$. In Optimum A, we initially set $k = 1$ and derived ξ_{max} and ξ for all the possible patterns of adding k links to the given network topology. We selected the link addition pattern giving the minimum ξ_{max}. When there were multiple patterns giving the minimum ξ_{max}, we selected from them the allocation pattern with the minimum ξ. If $\xi_{max} \leq \alpha_\xi$ was not satisfied, this process was repeated after incrementing k. For given k, we need to check $O(n^{2k})$ patterns of link addition, and $O(mn^3)$ calculation time is necessary to derive ξ_{max} and ξ for each pattern. Therefore, we need $O(mn^{3+2k})$ calculation time to obtain the optimum link addition pattern for given k. Although the required calculation time in Optimum A is equal to Algorithm A when $k = 1$, the calculation time in Optimum A exponentially increases as k increases.



(a) Original topology

(b) Topology after adding two links

Fig. 6. Topology of "CAIS Internet".

We compare the results of Algorithm A and Optimum A when applied to 4 networks in which n is less than or equal to 4 and $\xi_{max} \geq 4$. We set $\alpha_\xi = 3$. In Table II, the number of links added by the two algorithms is summarized for each of the four networks. The number of links added to satisfy $\xi_{max} \leq 3$ in Algorithm A wholly agreed with that in Optimum A in all four networks. Moreover, all the locations and link capacities added in the 4 networks except network 20 were identical in the two algorithms. Hence, we confirm that we can obtain an effective pattern of link addition close to the optimum using the proposed Algorithm A.

TABLE II
Comparison of Algorithm A and Optimum A.

ID	ξ_{max}	Algorithm A	Optimum A
9	4	2	2
14	4	1	1
20	7	3	3
25	4	1	1

Next, we show the results when applying Algorithm A to five networks in which ξ_{max} was extremely large and setting $\alpha_\xi = 5$. Table III summarizes the four stability criteria, ξ_{max}, ξ, ε_{max}, and ε, before and after Algorithm A was applied as well as the link count in the original topology, m, and the number of links added, x. Except for network 2, we confirm that we can reduce ξ_{max} about 50 to 80% by simply adding a small number (about 10% of m) of links. We can also reduce ξ about 20 to 50%. However, the stability qualities related to the flow count, i.e., ε_{max} and ε, were degraded in some networks because these qualities were not considered when Algorithm A was applied.

TABLE III

RESULTS WHEN ALGORITHM A WAS APPLIED.

ID	m	x	ξ_{max}	ξ	ε_{max}	ε
2	92	42	$25 \rightarrow 5$	$1.95 \rightarrow 1.02$	$14.8 \rightarrow 14.8$	$0.13 \rightarrow 0.46$
4	55	8	$24 \rightarrow 5$	$2.07 \rightarrow 1.43$	$0.26 \rightarrow 0.27$	$0.03 \rightarrow 0.03$
8	44	5	$12 \rightarrow 5$	$1.96 \rightarrow 1.40$	$3.69 \rightarrow 3.69$	$0.14 \rightarrow 0.16$
15	53	3	$17 \rightarrow 5$	$1.92 \rightarrow 1.27$	$0.24 \rightarrow 0.16$	$0.03 \rightarrow 0.03$
26	110	9	$15 \rightarrow 5$	$1.45 \rightarrow 1.19$	$1.54 \rightarrow 9.57$	$0.06 \rightarrow 0.27$

B. Results of Applying Algorithm B

Next, we show the results of Algorithm B. To see the tendency of links for which capacities were increased, we again show the original topology of network 8 (CAIS Internet) in Fig. 7. First, Algorithm B increased the capacity of link (7,34) from 45 to 155 Mbps, and we confirm that this link had the smallest capacity on the shortest loop 7-8-9-10-11-34-7 with a 6-hop length. The cost of path 7-34 was 0.0222 and greater than the cost of path 7-8-9-10-11-34, i.e., 0.0129, so link (7,34) is not used in the normal state. However, as a result of detouring some of the flows on this shortest loop to link (7,34) at SLFs of other links, such as link (9,10) on this shortest loop, the increase ratio of flow count on link (7,34) becomes large. The second link for which capacity was increased from 45 to 155 Mbps was link (1,12), and we also confirm that this link had the smallest capacity on the shortest loop 1-0-14-13-12-1 with a 5-hop length. Hence, we can say that Algorithm B tends to select links with small capacities on the shortest loops consisting of links with a large variance of capacities.

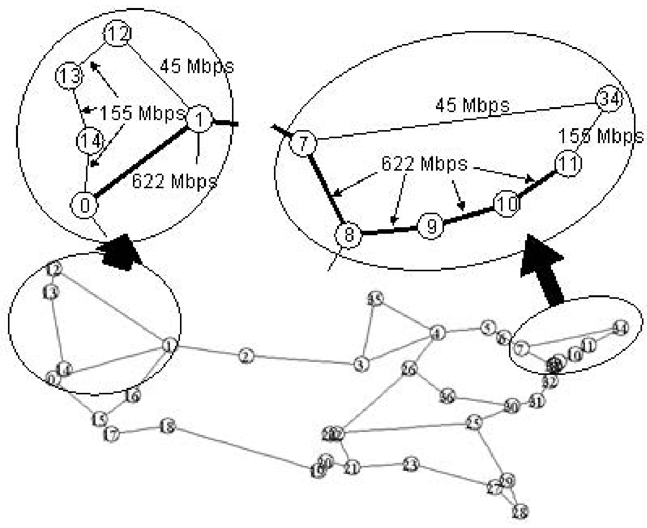


Fig. 7. Topology of "CAIS Internet".

To clarify the effectiveness of Algorithm B, we compare it with the case called *Optimum B*, in which the strictly optimum links for which capacities were increased were selected. Optimum B checked all the possible patterns to increase capacities among existing links and found the minimum number of links for which capacities are increased satisfying $\varepsilon_{max} \leq \alpha_{\varepsilon}$. In Optimum B, we initially set $k = 1$ and derived ε_{max} and ε for all the possible patterns of selecting k links on the given network topology. We selected the k links giving the minimum ε_{max}. When there were multiple patterns giving the minimum ε_{max}, we selected from them the allocation pattern with the minimum ε. If $\varepsilon_{max} \leq \alpha_{\varepsilon}$ was not satisfied, this process was repeated after incrementing k. For given k, we need to check $O(m^k)$ patterns of link selection, and $O(mn^3)$ calculation time is necessary to derive ε_{max} and ε for each pattern. Therefore, we need $O(m^{k+1}n^3)$ calculation time to obtain the optimum link set for given k. Although the required calculation time in Optimum B is equal to that of Algorithm B when $k = 1$, the calculation time in Optimum B exponentially increases as k increases.

We compare the results of Algorithm B and Optimum B when applied to three networks in which m is less than or equal to 39, the cardinality of link capacities is 3 or less, and $\varepsilon_{max} > 1$. We set $\alpha_{\varepsilon} = 1$. In Table IV, the number of capacity-increased links by the two algorithms is summarized for each of the three networks. The number of capacity-increased links to satisfy $\varepsilon_{max} \leq 1$ in Algorithm B wholly agreed with that of Optimum B in all three networks. Moreover, all the links and link capacities after increase in the three networks except network 8 were identical in the two algorithms. In network 8, only the link capacity after an increase of one link was different between the two algorithms. Hence, we confirm that we can obtain an effective set of links close to the optimum using the proposed Algorithm B.

TABLE IV

COMPARISON OF ALGORITHM B AND OPTIMUM B.

ID	ε_{max}	Algorithm B	Optimum B
6	1.03	1	1
8	3.69	3	3
31	1.05	2	2

Next, we show the results when applying Algorithm B to six networks in which ε_{max} was extremely large and setting $\alpha_{\varepsilon} = 1$. Table V summarizes the four stability criteria before and after Algorithm B was applied as well as the link count in the original topology, m, and the number of capacity-increased links, y. We confirm that we can reduce ε_{max} to less than unity by only increasing the capacities of a small number (about less than 10% of m) of links. We can also reduce ε about 50 to 80%. However, the stability qualities related to the flow hop length, i.e., ξ_{max} and ξ, were almost constant because these qualities were not considered when Algorithm B was applied.

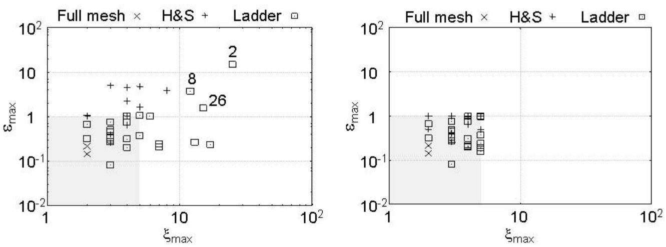
TABLE V

RESULTS WHEN ALGORITHM B WAS APPLIED.

ID	m	y	ε_{max}	ε	ξ_{max}	ξ
1	25	1	$5.00 \rightarrow 1.00$	$0.21 \rightarrow 0.10$	$3 \rightarrow 3$	$1.29 \rightarrow 1.31$
2	92	1	$14.8 \rightarrow 0.98$	$0.13 \rightarrow 0.03$	$25 \rightarrow 25$	$1.95 \rightarrow 1.97$
3	88	4	$3.79 \rightarrow 0.96$	$0.16 \rightarrow 0.06$	$8 \rightarrow 8$	$1.34 \rightarrow 1.36$
5	154	7	$4.74 \rightarrow 0.49$	$0.12 \rightarrow 0.02$	$5 \rightarrow 5$	$1.32 \rightarrow 1.34$
8	44	3	$3.69 \rightarrow 0.55$	$0.14 \rightarrow 0.05$	$12 \rightarrow 12$	$1.96 \rightarrow 2.06$
23	45	4	$4.53 \rightarrow 1.00$	$0.19 \rightarrow 0.10$	$4 \rightarrow 4$	$1.27 \rightarrow 1.34$

C. Results of Applying Algorithm C

Finally, we show the results of applying Algorithm C to the 36 networks when setting the design target as $\xi_{max} \leq 5$ and $\varepsilon_{max} \leq 1$. Figure 8(a) plots ε_{max} against ξ_{max} in the original topologies for each of the 36 networks, and we aim to make all the points of the 36 networks fall within the gray-painted area. In three H&S networks, 13, 16, and 19, twelve ladder networks, 7, 9, 12, 14, 17, 21, 22, 24, 25, 27, 35, and 36, and all the full mesh networks, the original network topologies and link capacities had already satisfied the design target, $\xi_{max} \leq 5$ and $\varepsilon_{max} \leq 1$. The remaining 19 networks, i.e., about 53% of all 36 networks, did not satisfy the design target, and we applied Algorithm C to these 19 networks. Figure 8(b) also shows ε_{max} against ξ_{max} after Algorithm C was applied for each network. We confirm that all the networks satisfied the design target by applying Algorithm C.



(a) Before modification (b) After modification

Fig. 8. ξ_{max} and ε_{max} before and after modification.

Table VI(a) summarizes x, the required number of locations to add links, and y, the required number of links for which capacities are increased, to satisfy the design target, $\xi_{max} \leq 5$ and $\varepsilon_{max} \leq 1$, in each of the 9 H&S networks whose original networks did not satisfy the design target. In the same way, Table VI(b) summarizes x and y to satisfy the design target in each of the ten ladder networks whose original networks did not satisfy the design target. As shown in Fig. 8(a), H&S networks tend to have large ε_{max} although ξ_{max} is small, so y was larger than x. On the other hand, ladder networks, except networks 2, 8, and 26, tend to have large ξ_{max} although ε_{max} is small, so x was larger than y. Both ξ_{max} and ε_{max} are large in networks 2, 8, and 26, so both link additions and link capacity increases were necessary to satisfy the design target and both x and y are large in these three networks. We confirm that the quality stabilities at SLFs can be dramatically improved and the design target can be satisfied by simply adding a few links or increasing the capacities of a few existing links in all the networks except networks 2, 8, and 26. Using proposed Algorithm C, we can effectively find the smallest number of locations to add links or the links for which capacities are increased to satisfy the design target.

TABLE VI

REQUIRED NUMBER OF ADDED LINKS AND CAPACITY-INCREASED LINKS.

ID	x	y	ID	x	y
1	0	1	**29**	0	1
3	1	4	**32**	0	2
5	0	7	**33**	0	7
23	0	4	**34**	0	2
28	0	4			

(a) H&S networks

ID	x	y	ID	x	y
2	17	12	**15**	3	0
4	9	0	**20**	1	0
6	0	1	**26**	6	4
8	7	6	**30**	1	0
10	1	0	**31**	0	2

(b) Ladder networks

VI. CONCLUSION

At the time of link or node failure, flows are detoured around the failed portion, so the hop count of flows and the link load could change dramatically as a result of failure. For ISPs, how to effectively improve the stability of qualities at failures with the minimum investment cost is an important issue, and they need to effectively select a limited number of locations to add link facilities. We proposed efficient algorithms to select locations where links should be added to effectively improve the stability of flow hop length at single link failures (SLFs) and to select links for which capacities should be increased to effectively improve the stability of flow count on each link at SLFs. We also proposed an algorithm combining both these algorithms to consider the stability of both qualities at SLFs. Through numerical evaluation using 36 actual ISP backbone networks, we clarified that we can obtain desirable locations to add link facilities close to the optimum using the proposed design algorithms. Only a few locations are critical for improving the quality stability at SLFs, and we can dramatically improve these qualities by simply adding a few link facilities by using the proposed algorithms. We will investigate the method of adding link facilities when the traffic demand matrix obeys the gravity model, and when the link weights are optimally designed in future work. We will also investigate the design method considering the costs of link deployments in future work.

REFERENCES

[1] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot, “Characterization of Failures in an IP Backbone, IEEE INFOCOM 04.

[2] E. C. G. Wille, M. Mellia, E. Leonardi, and M. A. Marsan, Topological Design of Survivable IP Networks Using Metaheuristic Approaches, Lecture Notes in Computer Science, Springer-Verlag, pp. 191?206, 2005.

[3] L. B. Page and J. E. Perry, Reliability Polynomials and Link Importance in Networks, IEEE Trans. Reliability, 43(1), pp. 51?58, 1994.

[4] F. P. Tsen, T. Sung, M. Lin, L. Hsu, and W. Myrvold, Finding the Most Vital Edges with Respect to the Number of Spanning Trees, IEEE Trans. Reliability, 43(4), pp. 600?603, 1994.

[5] H. Shen, Finding the K Most Vital Edges with Respect to Minimum Spanning Tree, IEEE NAECON 1997.

[6] H. Tangmunarunkit et al., Network Topology Generators: Degree-Based vs. Structural, ACM SIGCOMM 2002.

[7] N. Kamiyama and H. Miwa, “Connectivity and Stability at Failures in ISP Backbone Networks, IEEE GLOBECOM 08.

[8] P. Mahadevan et al., “The Internet AS-Level Topology: Three Data Sources and One Definitive Metric, ACM CCR, 36(1), pp.17?26, 2006.

[9] http://www.caida.org/tools/visualization/mapnet/Data/

[10] Cisco, Configuring OSPF, 1997.