NORTHWESTERN
UNIVERSITY

Electrical Engineering and Computer Science Department

Technical Report
NWU-EECS-11-08
August 8th, 2011

Toward Internet Emergency Response via Reconfiguration in Internet eXchange Points

Kai Chen, Chengchen Hu, Xitao Wen, Yan Chen, Bin Liu

Abstract

The Internet has become a key infrastructure for global information society. However, the availability of the Internet can be substantially challenged when a disaster strikes, because in such cases the recovery process as of today is largely manual, slow and inefficient. Recognizing that a key issue for obtaining availability is to find the connectivity to the lost destinations, we exploit Internet eXchange Points (IXPs) wherein multiple Autonomous Systems (ASes) exchange traffic. The core of this paper is to present IER, a first Internet Emergency Response design, to substantially speedup the recovery of the Internet availability after an emergency using IXPs. We introduce a detailed IER framework including the routing resource identification, resource allocation mechanisms as well as network reconfiguration strategies. We extensively evaluate our resource allocation mechanisms using synthetic data generated from realistic Internet AS topology and IXP dataset. Our results suggest that our resource allocation process is fast and is able to deliver reasonably good recovery rates in a series of settings, for example, in a major emergency, it can figure out how to recover 2.4+ million disconnected AS pairs within 11 seconds.

Keywords: Internet routing, IXP, Reliability

Towards Internet Emergency Response via Reconfiguration in Internet eXchange Points

Kai Chen, Chengchen Hu[†], Xitao Wen, Yan Chen, Bin Liu[⋆]
Northwestern University, [†]Xi'an Jiaotong University, [⋆]Tsinghua University

***Abstract*—The Internet has become a key infrastructure for global information society. However, the availability of the Internet can be substantially challenged when a disaster strikes, because in such cases the recovery process as of today is largely manual, slow and inefficient. Recognizing that a key issue for obtaining availability is to find the connectivity to the lost destinations, we exploit Internet eXchange Points (IXPs) wherein multiple Autonomous Systems (ASes) exchange traffic. The core of this paper is to present IER, a first Internet Emergency Response design, to substantially speedup the recovery of the Internet availability after an emergency using IXPs. We introduce a detailed IER framework including the routing resource identification, resource allocation mechanisms as well as network reconfiguration strategies. We extensively evaluate our resource allocation mechanisms using synthetic data generated from realistic Internet AS topology and IXP dataset. Our results suggest that our resource allocation process is fast and is able to deliver reasonably good recovery rates in a series of settings, for example, in a major emergency, it can figure out how to recover 2.4+ million disconnected AS pairs within 11 seconds.**

I. Introduction

A. Motivation

The Internet has become a key infrastructural component of the global information-based society, so any interruption to its availability would result in significant societal impacts. Thus, when listing its requirements on the Internet, the GENI initiative [14] states that "any future Internet should attain the highest possible level of availability."

However, while the current Internet has demonstrated remarkable availability and responsiveness in most cases, they can still be substantially challenged when a disaster or emergency strikes. Consider the Taiwan earthquake that struck on December 26, 2006 as an example. This earthquake was one of several major incidents that caused damage to the Internet. Even one week after the Taiwan earthquake, the number of outage Internet networks was still in the order of thousands [26]. This slow recovery process has caused substantial global and regional influences.

What is the recovery process? Fast repair of the outage links is clearly necessary. However, the repair of physical equipment is typically a time-consuming and expensive process. For the Taiwan earthquake, the repair of the damaged under-sea cables did not finish until Feburary 14, 2007 [26]. This delay is too long to be acceptable. To speedup the recovery process, the affected ASes have no choice but to seek out other connectivity options to achieve faster recovery. For the Taiwan earthquake, we conducted a survey of some involved networks. The survey results are quite revealing. After the earthquake, the options available to the operator of a network using some of the damaged links were quite limited. To recover its routes to important destinations before link repair, the operator had to seek out already existing or easily establish-able connectivity from its routers to some other routers with connectivity and capacity. For example, one major ISP indicated that it had to query other networks to recover both connectivity to important destinations (*e.g.*, DNS and MSN) and also capacity to avoid congestion. Obtaining 25 Gbps short-term capacity for temporary use from other networks by January 5, 2007, it was able to carry about 80% of its normal traffic. Although it succeeded in recovering most of the traffic before the repair of the damaged links, the delay was still unacceptable – 9 days after the quake.

Why is the long delay? Although negotiations on contracts and finalization of legal proceedings between two networks take time, the major delay happened even before these steps. In particular, an Internet network seeking recovery lacks crucial information: which other networks can effectively help me with this recovery? Without architecture support, in the current Internet, an affected network has to rely on a manual process. This leads to slow, ineffective Internet emergency response.

B. Related Work

There are a spectrum of research on Internet reliability and failure recovery. They broadly fall into 3 categories: intra-domain, inter-domain and overlay. For example, proposals such as R3 [25], REIN [24] and FCP [18] are mainly for addressing intra-domain routing failures, and RON [4] is discussed in the context of overlay network. Similar to our IER, schemes like R-BGP [17], BRAP [23], MIRO [28] etc focus on inter-domain routing and failures. However, the key difference is that these schemes leverage on existing policy-allowed valley-free paths (*i.e.*, Internet self-resilience) to handle the failures. This greatly limits their capability when intensive Internet emergencies such as the Taiwan earthquake happen. Because in such scenarios, many disconnected ASes do not have valley-free paths between them, thus cannot be recovered even though they may be physically connected.

C. Our Approach and Contributions

We therefore investigate how to address serious Internet failures via potential resources beyond the existing efforts relying on Internet self-resilience. In this line, our earlier short paper [15] points out IXP as a promising venue to recover the emergencies. However, how to systematically

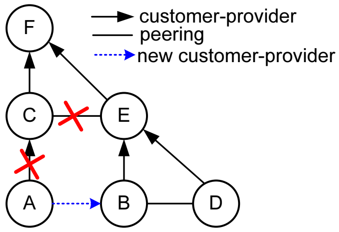


Fig. 1. The basic IER idea.

utilize these resources remains a challenge. A natural approach is to pre-provision the resources before an emergency happens, however this is very hard considering the huge failure scenario space. Similar space explosion problem exists in intra-domain failures [25]. So the reactive approach is a necessity. In this paper, we present IER, a first Internet Emergency Response design to identify, allocate and reconfigure potential routing resources in IXPs after emergencies.

To illustrate IER idea, see Fig. 1, suppose each node is an AS and the inter-domain routing follows valley-free policy (Section IV-A2). Consider the routing between A and D which originally uses path $ACED$, when link CE is cut, existing schemes such as RBGP and BRAP will use path $ACFED$ to survive. However, when link AC is cut, no scheme that exploits Internet self-resilience can recover such failure, and A loses connectivity to D. The IER idea is simply to find and set up easily-establishable connections (*i.e.*, relationships) to recover the Internet emergencies. In this case, if we are able to set up a new customer-provider relationship between A and B, then A obtains connectivity to D via B. However, finding the cases where the new relationships can be easily established is challenging. We explain why we select IXPs in Section II.

The objective of IER is not a totally automated system, as there may always be policy decisions and business procedures involving operators. Instead, the main goal is to speedup the discovery and negotiation of potential routing resources among ASes in IXPs during emergencies. To this end, we introduce a detailed IER framework including the resource identification, resource allocation mechanisms as well as network reconfiguration strategies. The highlights of this paper are,

- Inspired by the interaction with NANOG [19] operators, we propose a resource allocation scheme which captures the *general* interests of affected ASes and helper ASes. At the affected AS's side, we formulate and abstract the helper AS selection problem to be the Set Cover problem (NP-hard) and solve it with an efficient heuristic algorithm. At the helper AS's side, we formulate and abstract the affected AS selection problem to be the 0-1 Knapsack problem (NP-hard) and solve it with a polynomial dynamic programming algorithm after observing that all our inputs are strictly positive integers.
- We find that setting up new AS relationships in IXPs may introduce the side-effect of unexpected traffic shifting. We identify and analyze root causes of this problem, and propose solutions to address the problem accordingly.
- We extensively evaluate our resource allocation mechanisms using synthetic data generated from realistic AS topology and IXP dataset. Our results suggest that our resource allocation process is fast and is able to deliver reasonably good recovery rates in a series of settings.

Roadmap The rest of this paper is organized as follows. Section II overviews our problem and IER framework. Section III introduces the design of IER in detail. Section IV presents the evaluation results. Section V concludes the paper.

II. Background and Overview

A. Why IXPs?

After an emergency that causes damage to the AS reachability, the AS will react and utilize its backup path. The existing backup connectivity, however, is not sufficient during severe failures [27]. Then, an AS would seek additional network connectivity from other ASes if it does not want to wait for the repair of the damaged equipment. Throughout this paper, we call the AS who loses connectivity *affected AS*, and the one who can provide connectivity *helper AS*.

Consider a router $R1$ belonging to the affected AS, and a router $R2$ belonging to another AS. We say that there exists a *candidate emergency recovering connectivity* between $R1$ and $R2$ if the following four conditions are satisfied,

- *Fast on-demand connection*: Either there already exists a physical link connecting $R1$ and $R2$ or a physical link is easy and fast to establish.
- *Reachability:* Router $R2$ has reachability to the desired destinations router $R1$ wants to reach.
- *Capacity:* There is available bandwidth on the paths from router $R2$ to the destinations for the rerouted traffic.
- *Policy allowed:* Policies at both ASes allow connectivity between $R1$ and $R2$.

Among these, the *fast on-demand connection* is probably the most challenging one. In the Internet, an IXP is a colocation that allows two ASes to exchange traffic by means of mutual peering agreements. In most IXPs, the participant ASes are connected via Layer-2 switches [21]. Importantly, although the colocated routers are physically connected by switches, whether to establish BGP sessions over the physical link or not is up to individual ASes. Two physically connected routers are not connected logically via BGP if two ASes do not have a business contract. Actually, we found that more than 80% AS pairs does not have contracts in more than 50% of all IXPs.

Observing that currently many ASes put their routers together in IXPs which makes IXP an ideal venue for fast on-demand connection, we introduce IER based on IXPs. Note that we currently focus on IXP-related AS connectivity recovery and have not considered elsewhere, and even though, our evaluation results suggest that in an IXP, IER can recover 2.4+ million disconnected AS pairs in a major emergency. In the following, we first show an example of fast on-demand connection in IXPs. Then, we overview the IER design.

B. Example of Using IXPs for Recovery

Fig. 2 is a scenario in Japan IXP (JPIX) and London IXP (LNIX) during the Taiwan earthquake incident. Here, we just draw several ASes in these two IXPs for illustration. In the

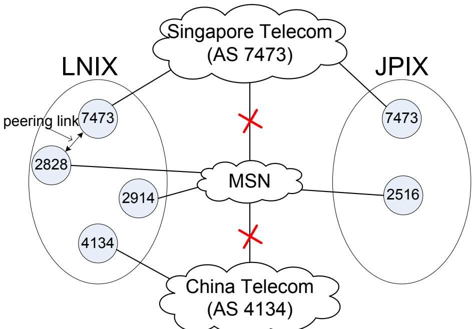


Fig. 2. Illustration of finding potential helper ASes in IXPs.

figure, the big ovals are IXPs and small circles are routers (from different ASes) within the IXPs.

In the earthquake, Singapore Telecom (AS 7473) and China Telecom (AS 4143) lost connectivity to MSN, while XO Communications (AS 2828) and NTT Communications (AS 2914) in LNIX and KDDI Corporation (AS 2516) in JPIX were still able to reach MSN. We also note that there is a peering relationship between AS 7473 and AS 2828. For the rest ASes, although they are physically connected via a Layer-2 switch, they have no BGP session with each other. To recover the connectivity from AS 4134 to MSN, we could set up a new provider-customer contract between AS 4134 and AS 2914 (or 2828) in LNIX, and the affected traffic from AS 4134 would traverse AS 2914 (or 2828) to reach MSN. To recover the connectivity from AS 7473 to MSN, we could either upgrade the peering link between AS 7473 and AS 2828 to a provider-customer link or set up a new provider-customer contract between AS 7473 and AS 2914 in LNIX (or AS 7473 and AS 2516 in JPIX). Then, the affected traffic from AS 7473 would traverse any of these helper ASes to reach MSN.

We note that an affected AS may appear in more than one IXP so that it can recover its connectivity in multiple IXPs simultaneously. For example, AS 7473 can recover its connectivity to MSN either in JPIX or in LNIX or both. Currently, we focus on the solution that for a destination an AS only selects the helper AS in one IXP and leave the more complex scenario as our future work.

C. IER Framework Overview

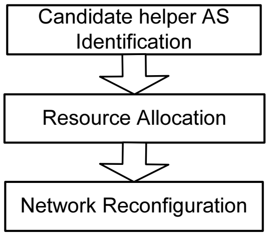


Fig. 3. The IER framework with three modules.

The IER framework is shown in Fig. 3. It contains three main modules: candidate helper AS identification, resource allocation and network reconfiguration.

1. Candidate helper AS identification. IER focuses on IXPs to recover emergencies. When an emergency happens, the first thing is to find candidate helper ASes that can potentially help to reach the lost destinations. To this end, IER employs the candidate helper AS identification module. In this module, we build a communication channel among the routers of IXP participant ASes. Over the channel, the affected ASes broadcast their desired resources (*e.g.*, lost destinations) and the helper ASes advertise what they can help. We will elaborate this procedure in Section III-A.

2. Resource allocation. Once the help information is advertised to the affected ASes, we enter the resource allocation module. The task here is to accommodate the demand-and-supply between affected ASes and helper ASes. Note that the resource allocation involves practical considerations. For example, resources are not free and helper ASes may charge money for their help. From an affected AS's perspective, it may want to reach as many lost destinations as possible with fewer new AS contracts. From an helper AS's perspective, it may want to sell as many resources as possible with fewer new contracts. We will elaborate this procedure in Section III-B.

3. Network reconfiguration. When the resources have been arranged, we need to reconfigure the routers accordingly. However, improper reconfiguration would cause the newly established contracts to carry unexpected traffic. To this end, we first identify and analyze the root causes for the unexpected traffic, and then propose the solutions to mitigate the problem. We will further elaborate this in Section III-C.

We note that ISP marketing is complex and cost-driven. The incentives behind IER are: 1. The serious emergency is not frequent and the risk for different ASes is comparable in a long run. The helper ASes may prepare for a rainy day in one incident. 2. The help process is relatively short and affects the helper ASes temporarily. 3. The helper ASes can also make profit by helping the affected ASes. Note that we by no means require all ASes to participate in IER. The effectiveness of IER depends on how many ASes join the recovery process. All designs below apply directly when partial ASes join IER.

III. Design

In this section, we introduce IER design in detail. First, we introduce the routing resource identification. Then, we elaborate the resource allocation procedure. Finally, we identify the unexpected traffic shifting and discuss how to address it.

A. Candidate Helper AS Identification

The main task of resource identification is to let the affected ASes know who can be the candidate helper ASes that can reach the lost destinations. Observing that in most IXPs the participant ASes (routers) are connected via Layer-2 switches, we propose to build a communication channel over the Layer-2 Ethernet. The channel building process is simple. If there is only one central switch connecting all the participant routers, the channel is trivially a star network. If there are multiple bridging switches, the channel is a tree topology.

The candidate helper AS identification process is in Fig. 4. Over the channel, the affected ASes first broadcast the desired destinations they want to reach and wait for response. At a helper AS's (*e.g.*, AS_j) side, it first waits sometime for a set of request information $\mathcal{D}$, then it checks its own information and decides a help set $\mathcal{H}_j$ ($\mathcal{H}_j \subseteq \mathcal{D}$) which contains all the destinations it can help to reach. At last, it broadcasts the help information over the channel. Then, at an affected AS's (*e.g.*, AS_i) side, once it receives a help set $\mathcal{H}_j$ from AS_j, it will determine if AS_j is a candidate helper AS by checking if $\mathcal{H}_j$ contains the desired destinations of itself. If so, we put these destinations into H_{ij} (*i.e.*, $H_{ij} = \mathcal{D}_i \cap \mathcal{H}_j$).

```
Resource_Identification($AS_i$, $AS_j$)
On an affected AS's (i.e., $AS_i$) side:
    broadcast the destination set $\mathcal{D}_i$ it wants to reach;
    once receiving the help information set $\mathcal{H}_j$ from AS$_j$;
    if $\mathcal{H}_j \cap \mathcal{D}_i \neq \Phi$
        identify $AS_j$ as a candidate helper AS;
        let $H_{ij} = \mathcal{D}_i \cap \mathcal{H}_j$;
On a helper AS's (i.e., $AS_j$) side:
    receiving the request information $\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, \cdots$ from
        different ASes $AS_0, AS_1, AS_2, \cdots$ in a time window;
    let $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots$;
    check $\mathcal{D}$ and its own information, and generate a help
        set $\mathcal{H}_j$ ($\mathcal{H}_j \subseteq \mathcal{D}$) it can help to reach;
    broadcast $\mathcal{H}_j$;
```

Fig. 4. Process of the candidate helper AS identification.

B. Resource Allocation

Once the help information is obtained by the affected ASes, we enter the resource allocation module. The main task of this module is to accommodate the demand-and-supply between affected ASes and helper ASes. Note that in IER design, we respect the interests of individual ASes. In other words, each affected/helper AS makes its own decision individually, entailing a decentralized resource allocation. We note that existing allocation schemes on bandwidth market such as [22] aim to achieve global welfare in a centralized manner. In their model, resources from different sellers are the same, while in our case resources from different helper ASes are used to reach different destinations. In addition, existing TE techniques [10] usually compute routing paths for load balancing in a centralized way given traffic matrices and topology. None of them is applicable to our problem.

1) Problem Analysis: We abstract the resource allocation problem using a flowchart in Fig. 5. The edges between helper ASes and destinations specify where these helper ASes can help to reach, which are known. The focus of resource allocation is between affected ASes and helper ASes, where we should specify edges connecting them.

When an affected (helper) AS has a list of candidate helper (affected) ASes, it will select its preferred helper (affected) ASes. There are factors such as pricing, policy, operation overhead, *etc* that may affect its decision. For example, a helper AS offering higher price will be considered with priority. In our model, we assume each affected/helper AS use the same pricing in emergencies. At an affected AS side, it is desirable for an AS to maximize its recovery rate. Further, given the same recovery rate, it is better to set up as fewer new contracts as possible. At a helper AS side, given its capacity, it may want to sell as many resources as possible to maximize its profit. Further, given the same amount, the helper AS may also want to sign fewer contracts since it may simplify the operations. Our model is not arbitrary, but the outcome of our interaction with NANOG operators when presenting [16]. While we think this is not the only choice, we believe it captures the *general*, not necessarily *all*, interests of affected ASes and helper ASes. Our schemes below can also be easily adopted in other cases such as varied pricing.

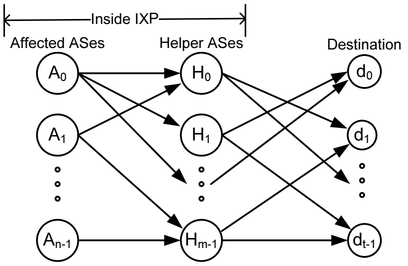


Fig. 5. Abstraction of the resource allocation problem.

2) Selection Scheme of An Affected AS: We now discuss how an affected AS, say AS_i, select its preferred helper ASes.

Formulation. As mentioned, a desired selection scheme from the affected AS's perspective is to maximize the recovery rate with as fewer new contracts as possible. So the key question is, given a set of candidate helper ASes (*i.e.*, $\mathbb{H} = \{H_{i0}, H_{i1}, \cdots, H_{ij}, \cdots\}$, how to choose the helper ASes to maximize the recovery rate while having minimal number of new contracts set up for recovery.

We call it *helper AS selection problem* and mathematically formulate it as follows:

Notation	Description
$\mathcal{D}_i$	the set of desired destinations AS_i wants to reach
$\mathcal{D}$	$\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots$, demand from all affected ASes
$\mathcal{H}_j$	the set of destinations of all affected ASes that AS_j can help
H_{ij}	the set of destinations of AS_i that AS_j can help
$\mathbb{H}$	$\mathbb{H} = \{H_{i0}, H_{i1}, \cdots, H_{ij}, \cdots\}$ is the potential help AS_i can get from all helper ASes
D	$D = H_{i0} \cup H_{i1} \cup \cdots \cup H_{ij} \cup \cdots$
d	a single destination that the affected AS wants to reach
α_H	an indicator showing whether helper AS H is selected
R_{ij}	the set of destinations requested from AS_i to AS_j
$\mathbb{R}$	$\mathbb{R} = \{R_{0j}, R_{1j}, \cdots, R_{ij}, \cdots\}$ is the requests from all affected ASes to AS_j
r_R	the amount of resources for request R
β_R	an indicator showing whether request R is satisfied

TABLE I
TABLE OF NOTATIONS.

$$Objective: \quad \text{minimize} \sum_{H \in \mathbb{H}} \alpha_H \tag{1}$$

$$Subject\ to: \quad \sum_{H:\ d \in H} \alpha_H \geq 1, \forall d \in D \tag{2}$$

$$\alpha_H \in \{0, 1\}, \forall H \in \mathbb{H} \tag{3}$$

The notations are in Table I. Note that in the formulation, our objective function 1 is to minimize the number of new contracts instead of maximizing the recovery rate. This is because constraint 2 ensures that all destinations must be covered by the set of selected helper ASes, which guarantees the maximal recovery rate. Constraint 3 shows that a helper AS is either selected (*i.e.*, $\alpha_H = 1$) or not (*i.e.*, $\alpha_H = 0$). We note that the helper AS selection problem is identical to the Set Cover problem [3], and so it is NP-hard.

Solution. We introduce a heuristic algorithm in Fig. 6. It is a greedy algorithm because at each round it selects the set that contains the maximum number of elements that are uncovered so far. Specifically, the algorithm runs as follows. At each round, the set U contains the elements that are still uncovered. The set C is the list of helper ASes that have already been selected. In line 4, among the remaining sets the algorithm greedily selects a set H_ω (a helper AS) that covers as many uncovered elements as possible. When H_ω is determined, it is removed from H and placed in C, and all its elements are removed from U. Finally, when the algorithm terminates, C is a subset of $\mathbb{H}$ that contains all the elements in D.

```
Helper_AS_Selection($AS_i$)
  /* Input: $\mathbb{H} = \{H_{i0}, H_{i1}, \cdots, H_{ij}, \cdots\}$ */
  /* Output: A set of helper ASes selected by $AS_i$ */
1     let $H = \mathbb{H}$;
2     let $U = D = H_{i0} \cup H_{i1} \cup \cdots \cup H_{ij} \cup \cdots$;
3     let $C = \Phi$;
4     while($U \neq \Phi$) do
5         select a helper $H_\omega \in H$ that maximizes $|H_\omega \cap U|$;
6         $H = H - \{H_\omega\}$;
7         $U = U - H_\omega$;
8         $C = C \cup \{H_\omega\}$;
9     return $C$;
```

Fig. 6. The helper AS selection procedure for an affected AS AS_i.

Algorithm Analysis. The algorithm in Fig. 6 can run fast in polynomial time. This is important for our emergency recovery process. Specifically, the loop body of the algorithm (line 4-8) is $O(|D| * |\mathbb{H}|)$ and the number of iterations is bounded by $min(|D|, |\mathbb{H}|)$. So the algorithm runs in $O(|D| * |\mathbb{H}| * min(|D|, |\mathbb{H}|))$ time. Furthermore, we find that the algorithm can return a set of helper ASes (*i.e.*, C) that is within a boundary of the optimal one (*e.g.*, C^*). Specifically,

Theorem 1: $|C| \leq f(max\{|H_\omega| : H_\omega \in \mathbb{H}\})|C^*|$, where ω is a helper AS, and $f(n) = \sum_{i=1}^{n} \frac{1}{i}$ is the n-th harmonic number.

Proof: Please see Appendix. ■

3) Selection Scheme of A Helper AS: We now discuss how a helper AS, say AS_j, choose which affected ASes to help.

Formulation. As mentioned, a desired selection scheme from the helper AS's perspective is to maximize the resources it can sell out within its capacity. Furthermore, give the same amount, it will pick fewer affected ASes if possible. Then, the key problem is, given a set of requests from the affected ASes (*i.e.*, $\mathbb{R} = \{R_{0j}, R_{1j}, \cdots, R_{ij}, \cdots\}$, how to maximize its profit by selling as many resources as possible while keeping the number of selected affected ASes to be minimum.

We call it *affected AS selection problem* and mathematically formulate it as follows:

$$Objective: \quad \text{maximize} \sum_{R \in \mathbb{R}} r_R \beta_R \tag{4}$$

$$Subject\ to: \quad \sum_{R \in \mathbb{R}} r_R \beta_R \leq c_j \tag{5}$$

$$\beta_R \in \{0, 1\}, \forall R \in \mathbb{R} \tag{6}$$

Objective function 4 specifies our goal of maximizing the help the helper AS (*i.e.*, AS_j) provides. Constraint 5 tells that AS_j has its bounded capacity c_j. Constraint 6 shows that a request is either satisfied (*i.e.*, $\beta_R = 1$) or not (*i.e.*, $\beta_R = 0$). We observe that the affected AS selection problem is equivalent to the 0-1 Knapsack problem [9], and so it is NP-hard.

Solution. In this paper, lacking traffic demand distribution among ASes and real bandwidth information, we use number of reachable ASes to quantify the bandwidth. In other words, $r_R = |R|$ and c_j is measured in terms of the total number of destination ASes it can help. Interestingly, though there is no polynomial algorithm for the original 0-1 Knapsack problem, we found a polynomial algorithm to our problem using dynamic programming. This is because, in our case, the request from the affected ASes and the available capacity of the helper AS are all strictly positive integers.

In our algorithm, we define $m[i, c]$ to be the maximum achievable value less than or equal to c while satisfying requests from up to i affected ASes. We assume r_1, r_2, $\cdots$, r_i are listed in non-decreasing order. Then, we derive $m[i, c]$ recursively as follows,

$$m[i, c] = \begin{cases} 0 & \text{if } i = 0 \\ 0 & \text{if } n = 0 \\ m[i-1, c] & \text{if } r_i > c \\ max\{m[i-1, c], m[i-1, c-r_i] + r_i\} & \text{if } r_i \leq c \end{cases}$$

According to the above equations, our solution can be found by calculating $m[n, c_j]$, where n is the total number of requests and c_j is the capacity of AS_j. Additionally, in order to keep minimal number of new contracts, the only thing to do is to choose $m[i-1, c]$ when $m[i-1, c]$ is equal to $m[i-1, c-r_i] + r_i$ in the last equation.

Algorithm Analysis. The above equations are standard format for dynamic programming. If we use a table to store previous computations, the algorithm will run in $O(nc_j)$ time and $O(nc_j)$ space.

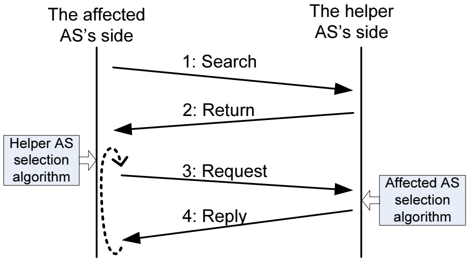


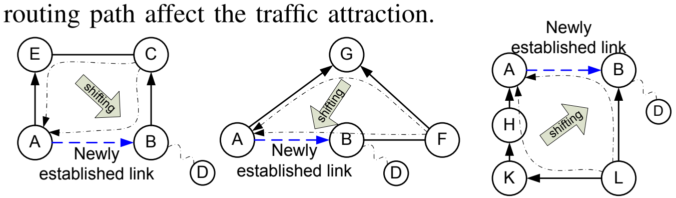
Fig. 7. A 4-way handshaking protocol for the whole allocation process.

4) The Entire Procedure: We use a 4-way handshaking protocol in Fig. 7 to describe the entire procedure of resource allocation. Step 1 and step 2 do the resource identification between the affected ASes and helper ASes as introduced in Section III-A. Between step 2 and step 3, each affected AS independently chooses the preferred set of helper ASes using the helper AS selection algorithm. After the preferred helper ASes are determined, the requests will be sent out in step 3. Up on receiving requests from a set of affected ASes, each helper AS will run the affected AS selection algorithm and then replies with yes or no at step 4. Then, at an affected AS's side, if all its requests are confirmed with yes, it is done with resource allocation procedure. Otherwise, if some of the requests are denied (*i.e.*, no) by some helper ASes, this means not all its desired destinations can be recovered at this moment. In next round, the affected AS will try to recover the unrecovered destinations again with other candidate helper ASes. The process will continue until all its desired destinations are recovered or there is no more resource available.

C. Network Reconfiguration

Once the resources have been allocated, we need to reconfigure routers and specify route advertisements accordingly. However, improper reconfiguration would cause the newly established links to attract unexpected traffic, which we call *unexpected traffic shifting*. In normal cases, the additional traffic is economically desirable if it brings extra revenue. But in emergencies, we disallow such traffic in order not to overload the recovering links. In what follows, we first identify scenarios of the unexpected traffic shifting and then discuss how to mitigate them.

1) Identifying Root Causes for the Unexpected Traffic Shifting: We investigate the root causes for the unexpected traffic shifting according to *valley-free prefer customer* Internet routing policies introduced in Section IV-A2. With this model, we observe that the export behaviors of ASes along the new routing path affect the traffic attraction.

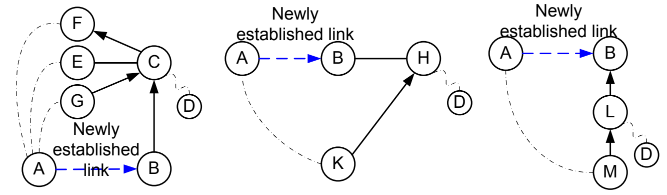


(a) Prefer customer to peer (b) Prefer peer to provider (c) Prefer the shortest path

Fig. 8. The root causes of the unexpected traffic shifting: due to the export behaviors of the helper AS.

Case 1: export behavior of the helper AS. As shown in Fig. 8, suppose A is the affected AS, C is the helper AS and D is the destination that B helps A to reach. In this case, it is natural to regard A as a temporary *customer* of B, and B would export path BA to all its providers, peers and customers.

- When the path BA is received by B's provider C (as shown in Fig. 8(a)), C will find a new path CBA to reach A. It may shift its traffic from CEA to CBA. This is because according to the routing policy C would prefer the path from its customer B to the one from its peer E (The case is the same if E is C's provider).
- When the path BA is received by B's peer F (as shown in Fig. 8(b)), F will find a new path FBA to reach A. It may shift its traffic from FGA to FBA. This is because according to the routing policy F would prefer the path from its peer B to the one from its provider G.
- When the path BA is received by B's customer L (as shown in Fig. 8(c)), L will find a new path LBA to reach A. It may shift its traffic from $LKHA$ to LBA. This is because according to the routing policy L would prefer the shortest path if all the paths are from its providers.



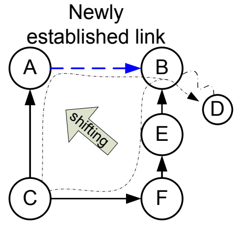
(a) Recovery path to D is along B's provider (b) Recovery path to D is along B's peer (c) Recovery path to D is along B's customer

Fig. 9. The root causes of the unexpected traffic shifting: due to the export behaviors of the ASes beyond the helper AS.

Case 2: export behaviors of the ASes beyond the helper AS. In Fig. 9, we use B's direct next-hop AS as an example. However, the analysis should generalize to all the intermediate ASes along the path from the helper AS to the destination. In the figure, we assume the path BA is advertised by B along the path towards D.

- When the path BA is received by B's provider C (as shown in Fig. 9(a)), since it is a path from its customer, C will export path CBA to all its neighbors.
  - When C's provider F receives the path CBA, it is possible for F to shift its best path from $F...A$ to $FCBA$ if: 1) the original path from F to A goes through F's provider or peer, or 2) the length of $F...A$ is longer than $FCBA$.
  - When C's peer E receives the path CBA, it is possible for E to shift its best path from $E...A$ to $ECBA$ if: 1) the original path from E to A goes through E's provider, or 2) the length of $E...A$ is longer than $ECBA$.
  - When C's customer G receives the path CBA, it is possible for G to shift its best path from $G...A$ to $GCBA$ if: 1) the original path from G to A goes through G's provider, and 2) the length of $G...A$ is longer than $GCBA$.
- When the path BA is received by B's peer H (as shown in Fig. 9(b)), since it is a path from its peer, H will export path HBA only to its customers.
  - When H's customer K receives the path HBA, it is possible for K to shift its best path from $K...A$ to $KHBA$ if: 1) the original path from H to A goes through H's provider, and 2) the length of $K...A$ is longer than $KHBA$.
- When the path BA is received by B's customer L (as shown in Fig. 9(c)), since it is a path from its provider, L will export path LBA only to its customers.
  - When L's customer M receives the path LBA, it is possible for M to shift its best path from $M...A$ to $MLBA$ if: 1) the original path from M to A goes through M's provider, and 2) the length of $M...A$ is longer than $MLBA$.

Case 3: export behaviors of the affected AS. In Fig. 10, as the affected AS (*i.e.*, A) regards the helper AS (*i.e.*, B) as its *provider*, it only exports the newly learned path $B...D$ to its customer C. Then, C may shift its traffic from $CFEB...D$ to $CAB...D$. This is because C would prefer the shortest path to D if all the paths are from its providers. The case is similar for all customers of A and their downstream ASes.



Prefer the shortest path

Fig. 10. The root causes of the unexpected traffic shifting: due to the export behaviors of the affected AS.

2) Mitigating the Unexpected Traffic Shifting: Though there are many different scenarios in our analysis, we find that the path attraction is mainly caused by two reasons: 1. An AS prefers the shortest AS path to reach a destination; 2. An AS prefers the path from its customers to the one from its peers, and at last the one from its providers. In the following, we discuss two possible solutions to address this problem.

Solution 1. To address the traffic attraction due to reason 1, we propose to use AS-path prepending. For example, in Fig. 8(c), we require AS B to prepend itself in the path (*e.g.*, $BBBA$) before exporting to L such that when L receives the new route $BBBA$ from B, it will keep its original path $LKJA$. This is because $LBBBA$ is longer than the original route $LKJA$. Furthermore, since in our IER framework, all participant ASes are cooperative and it is relatively easier to control the export behavior of these participant ASes. Therefore, in addition to AS prepending, we require that each helper AS only propagates the new paths in the direction of the destinations. For example, in Fig. 8(a), we can have B only advertise the new path BA towards D so that the unexpected traffic shifting will not happen.

Solution 2. An alternative method to address the unexpected traffic shifting is that, the new route is advertised in a way such that only the ASes along the path to the desired destination can learn it. To achieve this, we can use BGP communities attribute [7]. For example, in Fig. 9(a), B can append an attribute SELECTIVE_EXPORT to the new route learned from A and specify the community value to be D, and then exports this route to the next-hop AS toward D (*i.e.*, C). When C receives such a route, it only exports it to its next-hop AS towards D, and such process continues until the route is advertised to D. This solution can completely eliminate the unexpected traffic shifting, however it requires the cooperation of all ASes along the recovery path.

IV. Evaluation

In this section, we evaluate the performance of IER design. We first introduce the evaluation methodology, and then present the results.

A. Evaluation Methodology

1) Dataset: Our evaluation is based on both the Internet AS-level topology and IXP dataset.

AS topology. A more complete AS graph is always better for our purpose. Therefore, we use the AS topology inferred in a previous paper [8] which contains data sources from both BGP and traceroutes. This topology contains 31845 nodes and 142970 links, which is the best AS topology we can get.

IXP dataset. We use the IXP data from a previous work [6]. They attempted to map all IXPs by using various databases (*e.g.*, IXP databases, IXP websites and IRR [1]), and by looking for IXPs in publicly available datasets through active measurements. Their efforts produce the most complete data about AS member list and AS links in each IXP.

Based on this IXP dataset, we analyze the un-used AS link ratios across all the IXPs. We define un-used AS link ratio in one IXP as $1 - \frac{\#\ of\ existing\ AS\ links}{the\ maximal\ possible\ \#\ of\ AS\ links}$. The result is summarized in Fig. 11. It demonstrates that there are a lot of resources inside the IXPs which can potentially be exploited for emergency recovery. For example, in more than 50% IXPs the un-used AS link ratios are more than 80%.

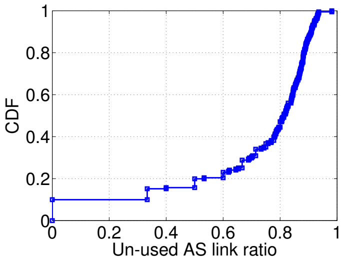


Fig. 11. The CDF of the un-used AS link ratios across all IXPs.

2) Modeling of Internet Routing: In practice, the routing policies used by an AS in the Internet are complex and not publicly known. However, in order to carry out the evaluation, we require a concrete model of routing policies. It is widely believed that business relationships play an important role in determining the policies of an AS [12, 13], we formulate the routing policies based on AS relationships [11].

In general, to select the best AS path to reach a destination, an AS first applies the local preference, then favors these AS paths with the minimal hops (*i.e.*, the shortest AS path) and at last uses a tie breaking method. Since it is hard to know the local preferences for ASes, we follow the widely used rule that an AS always prefers the paths from its customers, then its peers, and at last its providers.

In the modeling of exporting AS paths, we follow the widely used valley-free policy. The policy specifies how an AS exports its routes: 1. to its customers an AS will export all its routes; 2. to its peers or providers the AS only exports the routes from its customers instead of peers or providers. This exporting policy guarantees that any valid AS path should be non-valley.

3) Internet Emergency Scenarios: Our emergency scenarios are based on the real failures happened in the history. Specifically, we consider three failure scenarios[1] as follows:

- Tier-1 AS depeering: AS depeering could be caused by misconfigurations, physical damages or ISP contract termination. As pointed in [27] and evidenced by depeering between Cogent and Level3 [2], the Internet connectivity is significantly affected by Tier-1 AS depeering because the Tier-1 ASes are the core of the Internet.
- Major customer-provider link cut: Customer-provider links connect the networks in different tiers of the Internet. They provide network access and reachability for many low-tier ASes. Tier-1 ASes are the core of the Internet and the top tier ISPs. We therefore study the failures of Tier-1 customer-provider links.
- Regional failure: Regional failures such as the Taiwan earthquake [26], 9.11 event [20] and regional blackout

[1] Note that we do not have physical connectivity between ASes in the AS graph, so the failure of a logical AS link may relate to several physical links. This is a limitation of this paper as well as any previous work in this area.

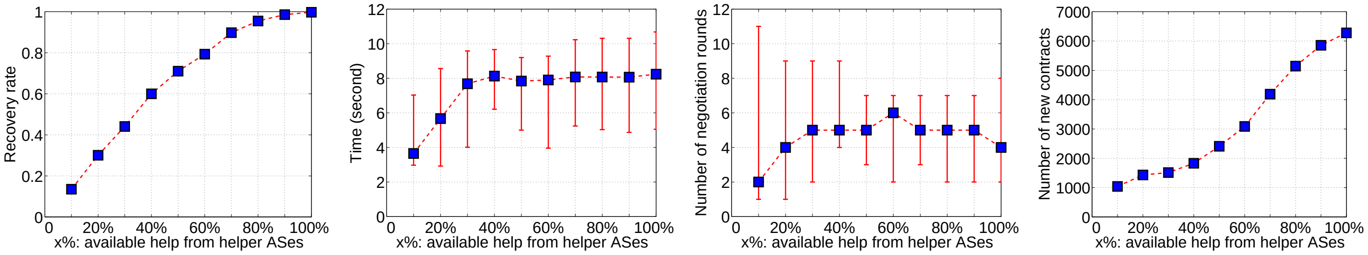


Fig. 12. The results of recovery process for a Tier-1 AS link depeering in LNIX. In our experiment, each affected AS has a time/round value. The error bar shows the max, mean, min values among all affected ASes. Given this is a decentralized and parallel process, the recovery time/round of the whole process is decided by the max value. (382 participant ASes in LNIX, 2414327 disconnected AS pairs can potentially be recovered.)

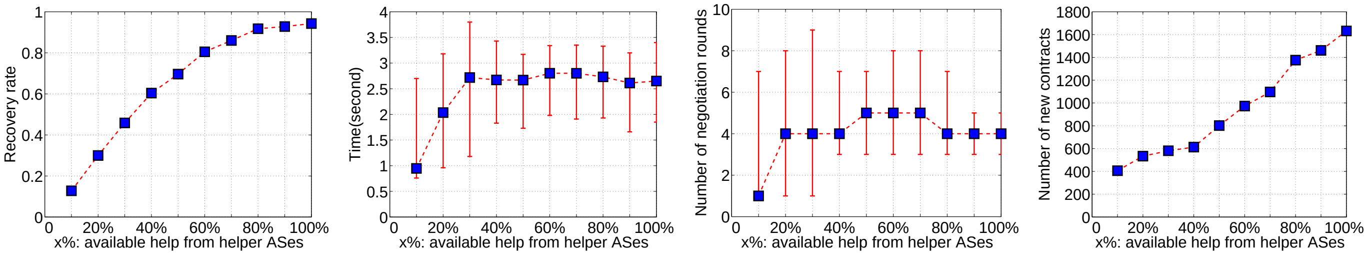


Fig. 13. The results of recovery process for simultaneously 50 critical Tier-1 customer-provider link cut in SIX. (141 participant ASes in SIX, 859120 disconnected AS pairs can potentially be recovered.)

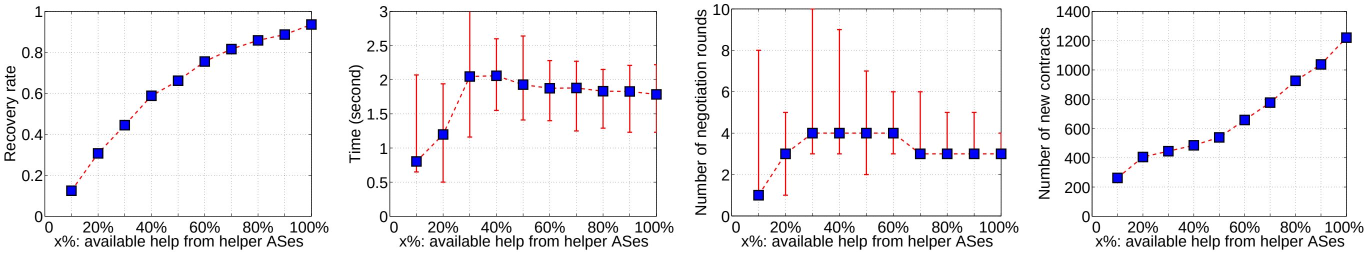


Fig. 14. The results of recovery process for regional failure (*i.e.*, 50% of 1470 AS links with 9 regional ASes are cut) in HKIX. (109 participant ASes in HKIX, 676412 disconnected AS pairs can potentially be recovered.)

can lead to hybrid link breakdown, which may include some peering links and provider-customer links. We study the Taiwan earthquake for such regional failures.

4) Evaluation Metrics: We implement the whole resource allocation procedure and choose LNIX (London), SIX (Seattle) and HKIX (Hong Kong) as representatives to evaluate our algorithms. We assume the above three emergency scenarios and check the following four metrics in each IXP.

- Recovery rate: We define the disconnected AS pairs to be the ones that lose connectivity but cannot be recovered via Internet self-resilience after an emergency. We check how many disconnected AS pairs can be recovered in each IXP with a range of moderate assumptions on the capacity of helper ASes.
- Time: The time consumption of the whole resource allocation process.
- Number of negotiation rounds: The total rounds of negotiation between the affected ASes and helper ASes in the entire resource allocation process.
- Number of new contracts: The number of new AS contracts needed for recovery.

B. Experimental Results

Lacking real traffic demand distribution among ASes and exact bandwidth information, we use number of reachable ASes to quantify the *traffic* of an AS hypothetically. In other words, if an AS can reach n ASes we assume the *traffic* associated with this AS is n. Encouragingly, the current Internet backbone utilization is around 30% [5] which suggests considerable bandwidth available for emergency recovery. Due to this, we assume a helper AS can provide help to reach additional $x\%$ ASes of its current reachable ASes. Thus, $x\%$ defines the available help from the helper ASes. We intentionally vary $x\%$ from 10% to 100% in our evaluation. Note that, for example, $x\% = 50\%$ indicates that some Internet link utilization rate may approximately be $30\% + 30\% \times 50\% = 45\%$.

For failure scale, in Tier-1 depeering case, we randomly depeer 1 Tier-1 link since even a single Tier-1 link depeering is a serious event. Our Tier-1 depeering is generated between 9 well-known Tier-1 ASes [27], which are 174, 209, 701, 1239, 2914, 3356, 3549, 3561 and 7018. In major customer-provider link cut case, we assume 10-50 Tier-1 customer-provider links to be cut simultaneously. In regional failure case, we pick out 9 seriously affected ASes in the Taiwan earthquake [26] (which are 4134, 4755, 4761, 4795, 4837, 7473, 9498, 9929 and 24077) and assume 10%-50% links of these ASes (totally 1470 such links) to be cut simultaneously. Note that all our failure scenarios are fairly serious.

Our evaluation results include the cases for all $x\%$ and all failure scales in each IXP. For space limitation, we only show a small subset of all the results we obtained in the following.

Fig. 12 shows the results of recovery process for a Tier-

AS link depeering in LNIX. Note that there are 382 ASes in LNIX and a total number of 2414327 disconnected AS pairs can potentially be recovered in this IXP. From the pictures, we find that our resource allocation algorithms can recover most disconnected AS pairs with a moderate assumption of available help from helper ASes. For example, with $x\% = 50\%$, our scheme can recover more than 70% of the to-be-recovered disconnected AS pairs. More importantly, our algorithms only consumes a short time. For example, in all the experiments our allocation procedures can be finished in less than 11 seconds and converge in less than 11 rounds. This is important for the emergency response because our goal is to speedup the recovery process. Since our resource negotiation is decentralized and each AS acts independently, we may lose global optimality on recovery rate, however we note that our results are reasonably good and our scheme is time efficient.

Fig. 13 and Fig. 14 show the recovery process for major customer-provider link cut and regional failure in SIX and HKIX respectively. There are several hundreds of thousands of disconnected AS pairs which can potentially be recovered in these two IXPs. All the figures have similar trends as in LNIX except that the time, number of negotiation rounds and number of new contracts in SIX and HKIX are smaller than those in LNIX. This is because they are smaller IXPs with 141 and 109 ASes respectively.

It is interesting to note that while the recovery rate and number of new contracts are increased with the available help from helper ASes (*i.e.*, $x\%$), the time and number of negotiation rounds are not strictly correlated with that. This is because our allocation process terminates either all the requests from the affected ASes are satisfied or no additional resource is available. So the recovery process will terminate earlier either when helper capacity is relatively small or relatively large just as shown in our results.

To summarize, we note that in above evaluation both our demand from affected ASes and help capacity from helper ASes are based on the hypothetical values which may deviate from the real case. However, the key information delivered by our results is that our resource allocation process is fast and is able to produce reasonably good recovery rates in a series of settings (*i.e.*, different IXPs and failure scenarios).

V. Conclusion

Different from most previous efforts using Interest self-resilience, we are the first to explore potential resources in IXPs to recover Internet emergencies. In this paper, we have introduced a detailed IER framework to identify, allocate and reconfigure the potential routing resources in IXPs. Especially, we propose a resource allocation scheme that captures the *general* interests of affected ASes and helper ASes. We abstract the helper AS selection problem to be the Set Cover problem and solve it with an efficient heuristic algorithm. We abstract the affected AS selection problem to be the 0-1 Knapsack problem and solve it with a polynomial dynamic programming algorithm after observing that our inputs are strictly positive integers. In addition, we analyze root causes of the unexpected traffic shifting and propose solutions to this problem accordingly. Finally, we extensively evaluate our resource allocation algorithms using synthetic data generated from realistic AS topology and IXP dataset. Our results suggest that our resource allocation process is fast and is able to deliver reasonably good recovery rates in a series of settings.

Appendix

A. Proof of Theorem 1

Proof: The rationale of our proof is to assign a unit of cost to each helper AS picked out by the heuristic procedure, spread the cost over the destination ASes covered for the first time, and then derive the expected inequality in Theorem 1 between the optimum and the approximation.

First, we define the cost of a destination $d \in H$ such that a cost is assigned only when covered for the first time. If d is covered for the first time by v_j, then we have:

$$c_d = \frac{1}{|H_{v_j} - H_{v_1} \cup H_{v_2} \cup ... \cup H_{v_{j-1}}|}$$

Suppose the algorithm finds a solution C of a total cost of $|C|$, this cost should have been spread out over all the links in H_{all}. In the meanwhile, the optimal solution C^* should also contain all links in H_{all}. Therefore, we have:

$$|C| = \sum_{d \in H_{all}} c_d \leq \sum_{H_x \in C^*} \sum_{d \in H_x} c_d \tag{7}$$

Next, we need a upper bound on the cost of every set H_{v_j}. We define u_i to be the number of elements in H_v remaining uncovered after $v_1, v_2, ..., v_i$ have been selected. Thus,

$$\begin{aligned}
\sum_{d \in H_x} c_d &= \sum_{i=1}^{k} (u_{i-1} - u_i) \frac{1}{|H_{v_j} - H_{v_1} \cup ... \cup H_{v_{j-1}}|}, \\
&\leq \sum_{i=1}^{k} (u_{i-1} - u_i) \frac{1}{u_{i-1}}, \text{(due to the greedy choice)} \\
&= \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{u_{i-1}}, \\
&\leq \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{j}, \text{(because } j \leq u_{i-1}) \\
&= \sum_{i=1}^{k} (\sum_{j=1}^{u_{i-1}} \frac{1}{j} - \sum_{j=1}^{u_i} \frac{1}{j}), \\
&= \sum_{i=1}^{k} (f(u_{i-1}) - f(u_i)), \\
&= f(u_0) - f(u_k) = f(u_0) = f(|H_x|).
\end{aligned} \tag{8}$$

Then combining Eq. 7 and Eq. 8, we have

$$\begin{aligned}
|C| &\leq \sum_{H_x \in C^*} \sum_{d \in H_x} c_d \\
&\leq \sum_{H_x \in C^*} f(|H_x|) \\
&\leq f(max\{|H_x| : H_x \in \mathbb{H}\}) \cdot |C^*|.
\end{aligned} \tag{9}$$

Therefore, Theorem 1 is proved.

■

References

[1] Internet Routing Register. http://www.irr.net.

[2] ISP spat blacks out Net connections. http://www.networkworld.com.

[3] Set Cover Problem. http://en.wikipedia.org/wiki/Set_cover_problem.

[4] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *SOSP*, 2001.

[5] N. Anderson. What exaflood? net backbone shows no signs of osteoporosis. http://arstechnica.com/old/content/2008/09/what-exaflood-net-backbone-shows-no-signs-of-osteoporosis.ars.

[6] B. Augustin, B. Krishnamurthy, and W. Willinger. Ixps: Mapped? In *IMC*, 2009.

[7] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. In *RFC*, 2009.

[8] K. Chen, D. Choffnes, R. Potharaju, Y. Chen, F. Bustamante, D. Pei, and Y. Zhao. Where the sidewalk ends: Extending the internet as graph using traceroutes from p2p users. In *CoNEXT*, 2009.

[9] T. Cormen. Introduce to algorithms. 2001.

[10] B. Fortz and M. Thorup. Internet traffic engineering by optimizing ospf weights. In *IEEE INFOCOM*, 2000.

[11] L. Gao. On inferring Autonomous System relationships in the Internet. *IEEE/ACM Transactions on Networking*, 2001.

[12] L. Gao and J. Rexford. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking.*, 2001.

[13] T. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking.*, 2002.

[14] G. P. Group. GENI: Conceptual design, project execution plan. GENI Design Document.

[15] C. Hu, K. Chen, Y. Chen, and B. Liu. Evaluating potential routing diversity for internet failure recovery. In *mini-INFOCOM*, 2010.

[16] C. Hu, K. Chen, Y. Chen, and B. Liu. Evaluating potential routing diversity for internet failure recovery. In *NANOG'49*, 2010.

[17] N. Kushman, S. Kandula, D. Katabi, and B. M. Maggs. R-BGP: staying connected in a connected world. In *NSDI*, 2007.

[18] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica. Achieving convergence-free routing using failure-carrying packets. In *SIGCOMM*, 2007.

[19] NANOG. North American Network Operators Group Mailing List Archive. http://www.merit.edu/mail.archives/nanog/.

[20] A. Ogielski and J. Cowie. Internet Routing Behavior on 9/11 and in the following weeks. www.renesys.com/tech/presentations/pdf/renesys-030502-NRC-911.pdf.

[21] R. Oliveira, D. Pei, W. Willinger, B.Zhang, and L. Zhang. In Search of the Elusive Ground Truth: The Internet's AS-level Connectivity Structure. In *ACM SIGMETRICS*, 2008.

[22] W. Stanczuk, J. Lubacz, and E. Toczylowski. Trading links and paths on a communication bandwidth market. *Journal of Universal Computer Science*, 2008.

[23] F. Wang and L. Gao. A backup route aware routing protocol: Fast recovery from transient routing failures. In *INFOCOM*, 2008.

[24] H. Wang, Y. R. Yang, P. H. Liu, J. Wang, A. Gerber, and A. Greenberg. Reliability as an interdomain service. In *SIGCOMM*, 2007.

[25] Y. Wang, H. Wang, A. Mahimkar, R. Alimi, Y. Zhang, L. Qiu, and Y. R. Yang. R3: Resilient routing reconfiguration. In *SIGCOMM*, 2010.

[26] S. Wilcox. Quaking Tables: The Taiwan Earthquakes and the Internet Routing Table. http://www.thedogsbollocks.co.uk/tech/0705quakes/AMSIXMay07-Quakes.ppt, 2007.

[27] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin. Internet routing resilience to failures: Analysis and implications. In *CoNEXT*, 2007.

[28] W. Xu and J. Rexford. Miro: multi-path interdomain routing. In *SIGCOMM*, 2006.