Detecting and Localizing Large-Scale Router Failures Using Active Probes

Qiang Zheng*, Guohong Cao*, Tom La Porta*, and Ananthram Swami†
*Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA
†Army Research Laboratory, Adelphi, MD
* {quz103, gcao, tlp}@cse.psu.edu † ananthram.swami@us.army.mil

***Abstract*—Detecting the occurrence of large-scale router failures and localizing the failed routers are critical to enhancing network reliability. We propose a two-phase approach for detecting and localizing large-scale router failures using traceroute-like active probes. To detect large-scale router failures, the detection phase is periodically invoked to probe all routers. When detecting large-scale router failures, the localization phase is triggered to identify the failed routers. We reduce the probing cost by avoiding three types of useless probes. For the routers whose status cannot be identified by probes, we develop a distance based method to estimate their failure probability. Experimental results based on ISP topologies show that the accuracy of our approach is higher than 96.5%, even when only 10% of routers are connected by end systems for probing. Compared with prior works, the proposed approach achieves much higher accuracy with lower probing cost.**

I. Introduction

IP networks have been widely deployed and have become significant communication infrastructures for a broad range of services. Many applications, such as financial transactions, online games, Voice over IP, and video services, require networks to be highly reliable. Some special networks like military networks also demand high reliability. Current network protection mechanisms, such as IP fast reroute [1], can only deal with sporadic link failures. To recover networks from large-scale router failures, service providers have to identify the failed routers and then send network operators to repair the network [2]. Consequently, detecting and localizing large-scale router failures are particularly important to enhancing network reliability.

Many events can cause large-scale router failures, including natural disasters and intentional attacks. For example, Hurricane Katrina [3], the Taiwan earthquake in December 2006 [4], and the Wenchuan earthquake in May 2008 [2] destroyed a large portion of the Internet near the disaster location. Additionally, intentional attacks like terroristic events (e.g., 911 attack [5]) and attacks by weapons of mass destruction (e.g., ElectroMagnetic Pulse attacks [6]) can also lead to large-scale router failures. The existing works on detecting and localizing network failures only consider small-scale failures [7], [8], and thus are not suitable for large-scale router failures. Some other prior works focus on discovering routing disruption [9], [10], but do not address how to identify the failed routers.

This work was supported in part by the Defense Threat Reduction Agency under grant HDTRA1-10-1-0085.

We propose an approach for detecting and localizing large-scale router failures using traceroute-like active probes sent from end systems. Generally, there are two important considerations in active probe based network failure detection and localization: *probing cost* and *accuracy*. Accordingly, minimizing the probing cost (i.e., the number of probing messages) and accurately localizing the failed routers are two major challenges addressed in our approach.

To minimize the probing cost, our approach consists of a periodic detection phase and a localization phase that is triggered on demand. We carefully choose probing paths for the two phases. For the detection phase, we aim at using minimal number of probing messages to probe all routers. We formalize this problem as a 0-1 integer programming problem and prove that it is NP-hard. Hence, we propose a greedy algorithm to solve it. For the localization phase, we discover three types of probes that do not provide useful information. We avoid these probes during the localization phase.

For large-scale router failures, active probes may be unable to identify the status of some routers, which will be explained in detail in Section IV. We propose a novel distance based model to estimate the failure probability of those routers. The basic idea is that large-scale router failures are usually within a geographically contiguous area. Hence, a router close to the failed routers may also fail with high probability. Through the estimated failure probability, we can identify routers that are highly likely to have failed.

Experimental results on ISP topologies show that the accuracy of our approach is higher than 96.5%, even when only 10% of routers are connected by end systems for probing. Besides, the probing cost of our approach is very low and is not affected by the number of end systems used for probing. Compared with prior works, it achieves higher accuracy with much lower probing cost.

The rest of this paper is organized as follows. In Section II, we present the network model and failure model, and introduce our approach. Section III and Section IV present the detection phase and the localization phase of the proposed approach. Section V evaluates the performance of our approach. Finally, Section VI concludes the paper.

II. Overview

In this section, we first introduce the network model and failure model, and then outline our approach.

A. Network Model and Failure Model

Similar to prior works on IP network monitoring and failure diagnosis [11]–[13], we model the network under study as a connected undirected graph $G(V, E)$, where $V = \{v_i | 1 \leq i \leq n\}$ is the set of nodes (IP routers) and E is the set of edges (bidirectional communication links between IP routers). The failure area is modeled as a geographically contiguous area in the network. Routers within it all fail. We do not make any assumption for the shape and location of the failure area, because in practice the failure area can be anywhere with any shape. Currently, we focus on the scenario of one failure area. Fig. 1 shows an example of large-scale router failures.

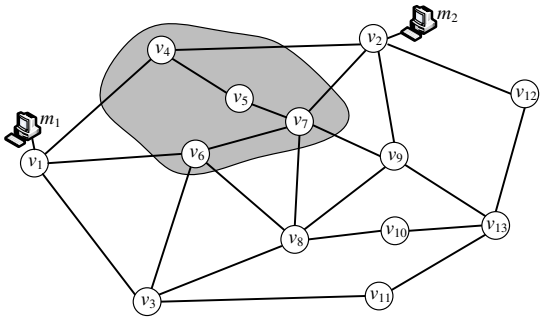


Fig. 1. An example of large-scale router failures. The shaded region denotes the failure area. Routers within it fail.

The proposed approach is built on a traceroute-like active probe, which is widely used in diagnosing network failures, e.g., [8], [12], [14]. Suppose n_p routers can be directly connected with end systems. We connect an end system with each of them and call these end systems the *probers*. Thus, we have n_p probers $m_1, \cdots, m_{n_p}$. In Fig. 1, there are two probers m_1 and m_2. A prober can issue traceroute-like active probes towards all routers. The probe from the prober m_i towards the router v_j follows the routing path $p_{i,j}$ from m_i to v_j. A probing path *covers* a router if it traverses this router.

The central Network Operations Center (NOC) controls probers to issue active probes and collects the probing result from them for failure analysis. When probing a path $p_{i,j}$, the prober sends κ probing messages to each router along $p_{i,j}$ (The default κ in traceroute is 3.). The *probing cost* of $p_{i,j}$ is defined as $\kappa h_{i,j}$, where $h_{i,j}$ is the number of hops in $p_{i,j}$.

When a router receives a probing message, it sends a reply message to the prober and hence we know that this router is alive. If a probing path contains failed routers, the probe can identify the first failed routers along the path. Besides, it can determine that the routers between the prober and the first failed router are alive. However, for the routers along the probing path but behind the first failed router, the probe cannot determine whether they are alive or failed. In Fig. 1, the probe from m_1 to v_5 and the probe from m_1 to v_2 follows $m_1 \rightarrow v_1 \rightarrow v_4 \rightarrow v_5$ and $m_1 \rightarrow v_1 \rightarrow v_4 \rightarrow v_2$, respectively. They identify that v_1 is alive and v_4 fails. Since v_4 fails, the probing messages cannot reach v_5 and v_2. Here, v_5 fails but v_2 is alive.

TABLE I
TABLE OF NOTATIONS

Symbols	Meaning
v_j	the jth router in the network
m_i	the ith prober
$p_{i,j}$	routing path from m_i to v_j
$h_{i,j}$	the number of hops in $p_{i,j}$
$a^k_{i,j}$	1 if $p_{i,j}$ traverses v_k, 0 otherwise
$d_{i,j}$	geographic distance between v_i and v_j

B. Design Overview

Intuitively, probers should periodically probe every path to identify the status of all routers. However, the probing cost is very high. To minimize the probing cost, we divide the whole task into two parts: detecting the occurrence of large-scale router failures and localizing the failed routers. Accordingly, our approach consists of a detection phase and a localization phase. Since we cannot foresee when large-scale router failures occur, the detection phase is periodically invoked to probe all routers. We seek to choose probing paths so as to cover all routers with minimal number of probing messages. The formulation and solution of this problem will be introduced in detail in Section III.

If the detection phase discovers that the number of failed routers exceeds the predefined threshold, the localization phase is immediately triggered to identify all failed routers. We may need to probe some additional paths to check the status of routers. We discover three types of probes that do not provide useful information, and avoid these probes during the localization phase. It is possible that the status of some routers cannot be identified by probes. Hence, we develop a distance based model to estimate the failure probability of these routers. Then we determine if they have failed based on the failure probability. The detail of the localization phase will be presented in Section IV.

III. The Detection Phase

We first define the problem and present the problem formulation, and then propose a greedy algorithm to solve it.

A. Problem Formulation

The objective is to choose probing paths to cover all routers with minimal number of probing messages. We define this problem as follows.

Definition 1 (Problem of probing paths selection): Given the network $G(V, E)$ and the set of probing paths $P = \{p_{i,j} | 1 \leq i \leq n_p, 1 \leq j \leq n\}$, the objective is to select probing paths from P that satisfy: (1) every router is covered; (2) the number of probing messages is minimal.

Before presenting the problem formulation, we define variable $a^k_{i,j}$ in Eq. (1) for the coverage relation.

$$a^k_{i,j} = \begin{cases} 1 & \text{if } p_{i,j} \text{ covers router } v_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Besides, we define variable $c_{i,j}$ in Eq. (2) to denote that probing path $p_{i,j}$ is chosen.

$$c_{i,j} = \begin{cases} 1 & \text{if } p_{i,j} \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Our problem can be formalized as a 0-1 integer programming problem as in Eq. (3)–Eq. (5).

$$\text{minimize} \quad \sum_{i=1}^{n_p}\sum_{j=1}^{n} \kappa c_{i,j} h_{i,j} \tag{3}$$

subject to

$$\forall\ 1 \le k \le n \quad \sum_{i=1}^{n_p}\sum_{j=1}^{n} c_{i,j} a_{i,j}^{k} \ge 1 \tag{4}$$

$$\forall\ 1 \le i \le n_p, 1 \le j \le n \quad c_{i,j} \in \{0,1\} \tag{5}$$

The objective function Eq. (3) minimizes the number of probing messages and the constraint in Eq. (4) means that the selected probing paths must cover every router.

B. The Bipartite Model and Greedy Algorithm

We model the above 0-1 integer programming problem with a bipartite $G_B = \{P, V, A\}$. Each vertex in the upper part P and lower part V denotes a probing path and a router. The vertex of $p_{i,j}$ has the attribute $\kappa h_{i,j}$, which is the probing cost of $p_{i,j}$. Here, we set κ to 3, i.e., the default value in traceroute. The set $A = \{a_{i,j}^{k} | p_{i,j} \in P, v_k \in V\}$ denotes the edges between two parts of the bipartite. Fig. 2 shows the bipartite model for the example in Fig. 1. There are $2\times13 = 26$ probing paths in the upper part and we only show 5 of them.

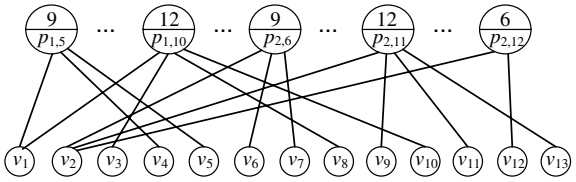


Fig. 2. The bipartite model of the problem of probing paths selection. In each vertex of the upper part, the value above $p_{i,j}$ denotes the attribute $\kappa h_{i,j}$, where κ is set to 3.

With the bipartite model, our problem can be stated as: selecting a set of vertices P_{det} from the upper part such that: (1) every vertex in the lower part has neighbors in P_{det}; (2) $\sum_{p_{i,j}\in P_{det}} \kappa h_{i,j}$ is minimal.

Theorem 1: The problem of probing paths selection is NP-hard.

Proof: By setting the attribute $\kappa h_{i,j}$ to 1 for every probing path $p_{i,j}$, the problem of probing paths selection is the same as the classic set cover problem. This means that the set cover problem is a special case of our problem. Since the set cover problem is NP-hard, our problem is also NP-hard. ■

We propose a greedy algorithm `PathSelection` shown in Algorithm 1 to solve our problem. The algorithm repeatedly selects a vertex from P and removes the corresponding vertices from V, until V becomes empty. Each time, it chooses $p_{i,j}$ with the smallest $\frac{\kappa h_{i,j}}{N_{i,j}}$ (line 4), where $N_{i,j}$, the number of neighbors that are currently uncovered, is computed in line 3. If multiple vertices have the same smallest $\frac{\kappa h_{i,j}}{N_{i,j}}$, we choose $p_{i,j}$ such that v_j is the leaf node of the routing tree of m_i. It helps reduce the probing cost of the localization phase, which will be explained in detail in Section IV-B. If the destination of these probing paths with the same smallest $\frac{\kappa h_{i,j}}{N_{i,j}}$ are all non-leaf nodes, we randomly choose a vertex from the candidates. Then, we remove $p_{i,j}$ from P (line 5) and remove the neighbors of $p_{i,j}$ from V (line 7–9). Because of removing $p_{i,j}$ from V, some vertices in P may have no neighbors; thus we remove them from P (line 10–12). When the algorithm terminates, the set P_{det} contains the selected probing paths.

Algorithm 1 PathSelection

Input: The bipartite $G_B = \{P, V, A\}$
Output: The set of probing paths P_{det}
Procedure:
1: $P_{det} \leftarrow \emptyset$
2: **while** $V \neq \emptyset$ **do**
3: $N_{i,j} \leftarrow$ the number of neighbors of vertex $p_{i,j} \in P$
4: $p_{i,j} \leftarrow$ the vertex in P with the smallest $\frac{\kappa h_{i,j}}{N_{i,j}}$
5: $P \leftarrow P - p_{i,j}$
6: $P_{det} \leftarrow P_{det} \cup p_{i,j}$
7: **for** each neighbor v_k of $p_{i,j}$ **do**
8: $V \leftarrow V - v_k$
9: **end for**
10: **for** each vertex $p_{a,b} \in P$ with no neighbors **do**
11: $P \leftarrow P - p_{a,b}$
12: **end for**
13: **end while**

The algorithm `PathSelection` is invoked when our approach is first deployed and when the network topology or routing configuration changes. In each detection round, probers issue active probes along the selected probing paths and then send the probing result to the NOC.

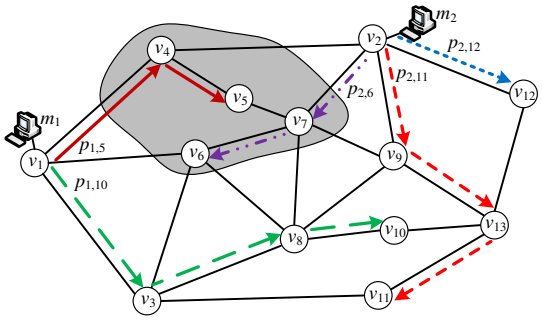


Fig. 3. The probing paths selected for the detection phase.

For the example in Fig. 1, the algorithm `PathSelection` chooses 5 probing paths $p_{1,5}$, $p_{1,10}$, $p_{2,6}$, $p_{2,11}$, and $p_{2,12}$ as shown in Fig. 3. Probing these paths classifies the status of routers into three categories, i.e., *live*, *failed*, and *unknown*, as shown in Table II.

TABLE II
THE PROBING RESULT OF THE DETECTION PHASE

Status	Routers
live	v_1, v_2, v_3, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}
failed	v_4, v_7
unknown	v_5, v_6

IV. The Localization Phase

We first discuss when to trigger the localization phase, and then describe three types of useless probes. Finally, we propose

a distance based model for estimating the failure probability of routers.

A. Triggering the Localization Phase

We use a simple and practical criterion to determine if the localization phase should be triggered. The localization phase is triggered if the detection phase discovers that the number of failed routers exceeds the predefined threshold τ. This criterion is based on the fact that most failures in the Internet are sporadic link failures [15]. If several router failures are detected, it is likely that large-scale router failures have occurred. We set the threshold τ to 2 in this paper. As shown in Table II, the detection phase discovers that at least two routers have failed. Hence, the localization phase is triggered.

B. Avoiding Useless Probes

Intuitively, each prober needs to probe paths towards all routers to identify their status. In Fig. 4, m_1 needs to issue 13 probes, and so does m_2. We identify the following three types of useless probes. Avoiding them can save many probing messages.

1) The prober only needs to probe the paths towards the leaf nodes of its routing tree, because the paths towards non-leaf nodes are included in the paths towards the leaf nodes. In Fig. IV-B, probing paths $p_{1,3}$ and $p_{1,11}$ are a part of $p_{1,13}$, and hence we do not need to probe them.
2) We do not need to probe the paths that are probed in the current detection phase. Unlike temporary link failures, large-scale router failures are usually long lasting. Thus, we can make use of the probing result in the current detection phase. This is the reason that we prefer the probing paths towards the leaf nodes of routing trees in Algorithm 1.
3) For a probing path $p_{i,j}$ containing a failed router v_k, if the routers between m_i and v_k are all live, we do not need to probe $p_{i,j}$, because probing it can only identify the failed router v_k. In our example, the detection phase identifies that v_4 and v_7 fail. As shown in Fig. 4, probing paths $p_{1,5}$, $p_{1,9}$, $p_{1,12}$, $p_{2,3}$, $p_{2,5}$, $p_{2,6}$, and $p_{2,10}$ satisfy the requirement, and thus we do not need to probe them.

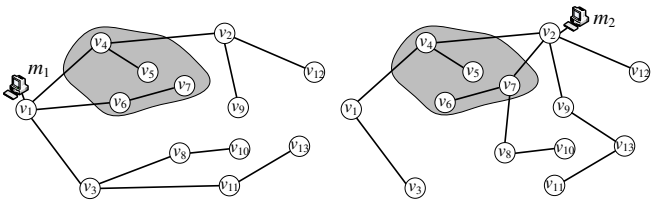


Fig. 4. The routing tree of the two probers.

After excluding the above three types of useless probes, the localization phase needs to probe only two paths $p_{1,7} : m_1 \rightarrow v_1 \rightarrow v_6 \rightarrow v_7$ and $p_{1,13} : m_1 \rightarrow v_1 \rightarrow v_3 \rightarrow v_{11} \rightarrow v_{13}$ with $(3+4)\kappa = 7\kappa$ probing messages. Therefore, excluding useless probes can save $\frac{26-2}{26} = 92.3\%$ probes and $\frac{74\kappa-7\kappa}{74\kappa} = 90.5\%$ probing messages. From the probing result of the two phases, we can obtain the status of routers as shown in Table III.

TABLE III
THE STATUS OF ROUTERS IDENTIFIED BY THE PROBES IN TWO PHASES

Status	Routers
live	v_1, v_2, v_3, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}
failed	v_4, v_6, v_7
unknown	v_5

C. Distance Based Router Failure Probability Estimation

It is possible that probes cannot identify the status of some routers (e.g., v_5 in Table III), especially when probers are very few or the failure area is large. We propose a distance based model to estimate the failure probability of the routers with unknown status. Since large-scale router failures are usually in a geographically contiguous area, a router close to the failed routers may also fail with high probability. The basic idea is to map the distance to a failure probability.

Let $d_{i,j}$ be the geographic distance between routers v_i and v_j. The probes in two phases identify some failed routers and the routers with unknown status, which are denoted by the set V_F and V_U, respectively. For $v_i \in V_U$ and $v_j \in V_F$, $P(v_i|v_j)$ is the conditional failure probability of v_i, when we know v_j has failed. We compute $P(v_i|v_j)$ with Eq. (6) by mapping $d_{i,j}$ to a failure probability with a function $\mathcal{F}$.

$$P(v_i|v_j) = \mathcal{F}(d_{i,j}) \tag{6}$$

There are two requirements for the mapping function $\mathcal{F}$. First, it needs to map $d_{i,j}$ to a real number between 0 and 1, i.e., $\mathcal{F} : \mathbb{R}^+ \rightarrow (0, 1)$. Second, it should be a strictly decreasing function, and thus a larger distance is mapped to a smaller failure probability. Many functions satisfy these requirements and can be chosen as $\mathcal{F}$. We compared several functions and choose $\mathcal{F}(x) = 0.9^{\sqrt{x}}$ in this paper.

Based on the conditional failure probability $P(v_i|v_j)$, we calculate the failure probability of v_i with Eq. (7).

$$f_i = 1 - \prod_{v_j \in V_F} (1 - P(v_i|v_j))) \tag{7}$$

In our example, the probes identify that v_4, v_6, and v_7 fail, and the status of v_5 is unknown. The calculation of the failure probability of v_5 is shown in Fig. 5.

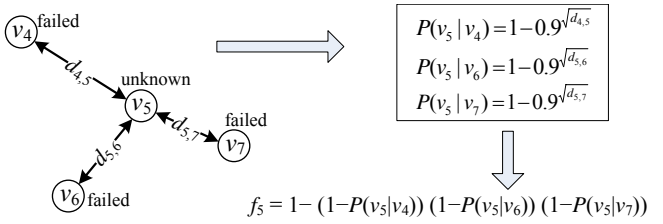


Fig. 5. The calculation of the failure probability of v_5.

Let $\gamma \in (0, 1)$ be the predefined threshold. A router v_i is said to fail if its estimated failure probability f_i is larger than γ. The threshold γ should be selected according to the mapping function $\mathcal{F}$ and the range of the distance $d_{i,j}$. In Section V, we will investigate the accuracy of our approach with different γ. In summary, the localization algorithm `LocalizeFailure` is shown in Algorithm 2.

Algorithm 2 LocalizeFailure

Input: The set V_F and V_U
Output: The set V_F
Procedure:
1: **for** each $v_i \in V_U$ **do**
2: $\quad f_i \leftarrow 1 - \prod_{v_j \in V_F}(1 - \mathcal{F}(d_{i,j}))$
3: **end for**
4: **for** each $v_i \in V_U$ **do**
5: $\quad$ **if** $f_i > \gamma$
6: $\quad\quad V_F \leftarrow V_F \cup v_i$
7: **end for**

V. Performance Evaluation

In this section, we evaluate the performance of our approach and compare it with prior works [14], [16].

A. Simulation Setup

The simulation is based on ten ISP topologies derived from the Rocketfuel project [17], which are summarized in Table IV. For each topology, we randomly place nodes in a 2000×2000 area. All topologies adopt the shortest path routing calculated based on hop count. To simplify the simulation, the failure area is a circle randomly placed in the 2000×2000 area with the radius randomly selected between 400 and 700. Nodes within the circle all fail.

TABLE IV
Summary of topologies used in simulation

Topology	# Nodes	# Links
AS209	58	108
AS701	83	219
AS1668	53	64
AS2914	70	111
AS3257	41	87
AS3320	70	355
AS3356	63	285
AS3549	61	486
AS3561	92	329
AS4323	51	161

Generally, most routers in the Internet cannot be directly connected with probers. Hence, the percentage of routers connected with probers is varied from 2% to 20% in increments of 2%. We randomly select routers and connect probers with them. The parameter κ is set to 3, i.e., a prober needs three messages to probe a router. The threshold τ used in triggering the localization phase is set to 2. We choose $\mathcal{F}(x) = 0.9^{\sqrt{x}}$ as the mapping function of Algorithm 2. The threshold γ in Algorithm 2 is varied from 0.1 to 1.0 with steps of 0.1. We run each simulation 1,000 times and report the average across the simulation set.

We compare the accuracy and probing cost of our algorithm `LocalizeFailure` with the *prefix probing* [14], [16], in which each prober sends probes towards the leaf nodes of its routing tree.

B. Accuracy

First we investigate the accuracy of our approach. The detection and localization result has three possibilities.

1) A live router is identified as live, or a failed router is identified as failed. Suppose the status of x routers are correctly identified. Then, the *accuracy* is defined as the ratio $\frac{x}{n}$, where n is the total number of routers.
2) A live router is identified as failed. Suppose y routers are in this case. The *false positive rate* is the ratio $\frac{y}{n}$.
3) A failed router is identified as live. Suppose z routers are in this case. The *false negative rate* is the ratio $\frac{z}{n}$.

The accuracy of our approach with varying threshold γ is shown in Fig. 6 and Fig. 7. Due to space limits, we only show the result when 10% and 20% of routers are connected with probers. The highest accuracy is achieved when $\gamma = 0.7$, which is higher than 96.5% in all topologies.

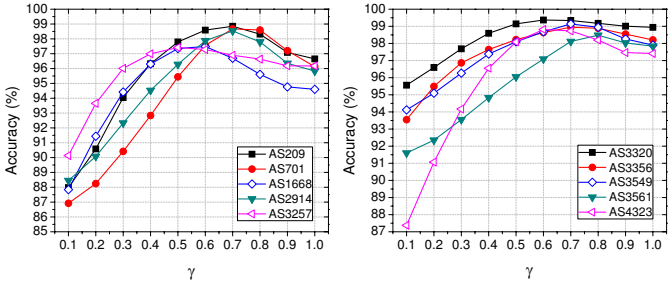


Fig. 6. The accuracy when 10% of routers are connected with probers.

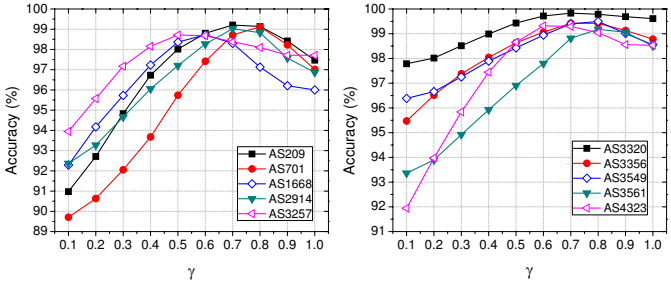


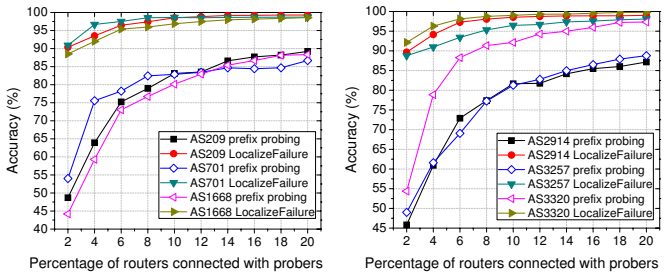
Fig. 7. The accuracy when 20% of routers are connected with probers.

The false positive rate and false negative rate when $\gamma = 0.7$ are shown in Table V. Due to space limits, we only show results for three topologies. The high accuracy together with the low false positive rate and false negative rate show the effectiveness of our approach.

TABLE V
The false positive rate (%) and false negative rate (%) when $\gamma = 0.7$ (Pos: false positive; Neg: false negative)

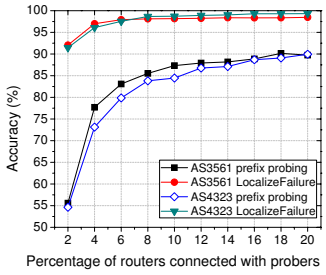
Prober ratio (%)	AS209		AS2914		AS3549	
	Pos	Neg	Pos	Neg	Pos	Neg
6	0.47	3.08	0.69	2.03	0.50	1.48
8	0.44	2.14	0.51	1.42	0.53	0.83
10	0.51	0.96	0.68	0.79	0.66	0.25
12	0.46	0.69	0.54	0.56	0.48	0.39
14	0.46	0.24	0.56	0.53	0.43	0.18

We compare our algorithm `LocalizeFailure` ($\gamma = 0.7$) with the prefix probing and show the result in Fig. 8. Our approach achieves much higher accuracy than prefix probing. Active probes may not identify the status of some routers, especially when there are few probers. The prefix probing cannot deal with these routers, while our approach uses the distance based method to estimate the failure probability of these routers. As a result, our approach achieves higher accuracy and the number of probers has little affect on the accuracy of our approach.

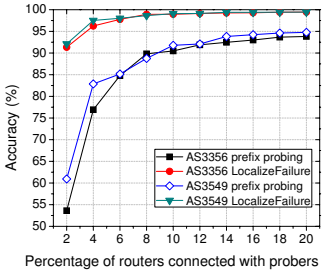


(a) AS209, AS701, and AS1668

(b) AS2914, AS3257, and AS3320

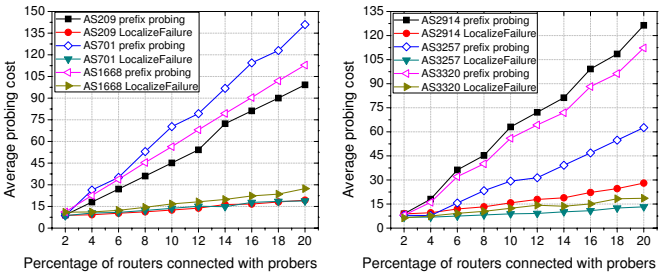


(c) AS3561 and AS4323



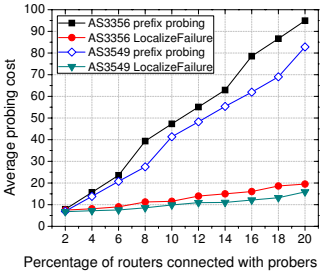
(d) AS3356 and AS3549

Fig. 8. The accuracy of the prefix probing and the algorithm `LocalizeFailure` with $\gamma = 0.7$.

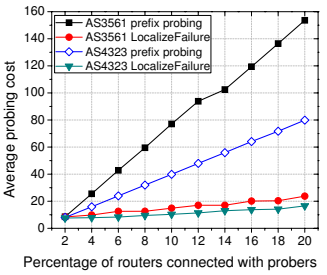


(a) AS209, AS701, and AS1668

(b) AS2914, AS3257, and AS3320



(c) AS3356 and AS3549



(d) AS3561 and AS4323

Fig. 9. The average probing cost of the prefix probing and the algorithm `LocalizeFailure`.

C. Probing Cost

Next we compare the probing cost of our algorithm `LocalizeFailure` and that of prefix probing. The performance metric *average probing cost* is defined as the average number of probing messages per router. The probing messages in our approach includes the probing messages in the detection phase and the localization phase. The evaluation result is shown in Fig. 9. The average probing cost of our approach is much lower than that of prefix probing. In prefix probing, the average probing cost is proportional to the number of probers, because each prober sends probes towards the leaf nodes of its routing tree. Our approach excludes three types of useless probes. Thus, the average probing cost is very low and increases quite slowly when the number of probers increases. Hence, our method has much better scalability than prefix probing.

VI. Conclusions

We proposed a two-phase approach for detecting and localizing large-scale router failures. For the detection phase, we propose an algorithm to choose probing paths which can cover all routers with minimal number of probing messages. For the localization phase, we identify three types of useless probes. Avoiding these probes helps reduce the probing cost significantly. For the routers whose status cannot be identified by probes, we develop a distance based method to estimate their failure probability. Experimental results based on ISP topologies show that the accuracy of our approach is higher than 96.5%, even when only 10% of routers are connected by end systems for probing. Compared with prior works, our method achieves much higher accuracy and with considerably lower probing cost.

References

[1] M. Shand and S. Bryant, "IP fast reroute framework," RFC5714, January 2010.

[2] Y. Ran, "Considerations and suggestions on improvement of communication network disaster countermeasures after the wenchuan earthquake," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 44–47, 2011.

[3] J. Cowie, A. Popescu, and T. Underwood, "Impact of Hurricane Katrina on Internet infrastructure," http://www.renesys.com/tech/presentations/pdf/Renesys-Katrina-Report-9sep2005.pdf, 2005.

[4] S. LaPerriere, "Taiwan earthquake fiber cuts: a service provider view," http://www.nanog.org/meetings/nanog39/presentations/laperriere.pdf, 2007.

[5] A. Ogielski and J. Cowie, "Internet routing behavior on 9/11 and in the following weeks," http://www.renesys.com/tech/presentations/pdf/renesys-030502-NRC-911.pdf, 2002.

[6] C. Wilson, "High altitude electromagnetic pulse (HEMP) and high power microwave (HPM) devices: threat assessments," http://www.fas.org/man/crs/RL32544.pdf, 2004.

[7] P. Barford, N. Duffield, A. Ron, and J. Sommers, "Network performance anomaly detection and localization," in *IEEE INFOCOM*, 2009.

[8] L. Cheng, X. Qiu, L. Meng, Y. Qiao, and R. Boutaba, "Efficient active probing for fault diagnosis in large scale and noisy networks," in *IEEE INFOCOM*, 2010.

[9] Y. Zhang, Z. M. Mao, and M. Zhang, "Effective diagnosis of routing disruptions from end systems," in *USENIX NSDI*, 2008.

[10] A. Haeberlen, I. Avramopoulos, J. Rexford, and P. Druschel, "NetReview: Detecting when interdomain routing goes wrong," in *USENIX NSDI*, 2009.

[11] K. Suh, Y. Guo, J. Kurose, and D. Towsley, "Locating network monitors: complexity, heuristics, and coverage," in *IEEE INFOCOM*, 2005.

[12] H. X. Nguyen, R. Teixeira, P. Thiran, and C. Diot, "Minimizing probing cost for detecting interface failures: Algorithms and scalability analysis," in *IEEE INFOCOM*, 2009.

[13] Q. Zheng and G. Cao, "Minimizing probing cost and achieving identifiability in network link monitoring," in *IEEE ICDCS*, 2010.

[14] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin, "Internet routing resilience to failures analysis and implications," in *ACM CoNEXT*, 2007.

[15] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *IEEE INFOCOM*, 2004.

[16] Y. Zhang, Z. M. Mao, and J. Wang, "A framework for measuring and predicting impact of routing changes," in *IEEE INFOCOM*, 2007.

[17] R. Sherwood, A. Bender, and N. Spring, "Measuring ISP topologies with Rocketfuel," in *ACM SIGCOMM*, 2002.