PAPER

Failure Microscope: Precisely Diagnosing Routing Instability

Hongjun LIU[†a)], ***Nonmember***, **Baokang ZHAO**[††b)], ***Member***, **Xiaofeng HU**[††c)], **Dan ZHAO**[††d)], ***and*** **Xicheng LU**[††e)], ***Nonmembers***

SUMMARY Root cause analysis of BGP updates is the key to debug and troubleshoot BGP routing problems. However, it is a challenge to precisely diagnose the cause and the origin of routing instability. In this paper, we are the first to distinguish link failure events from policy change events based on BGP updates from single vantage points by analyzing the relationship of the closed loops formed through intersecting all the transient paths during instability and the length variation of the stable paths after instability. Once link failure events are recognized, their origins are precisely inferred with 100% accuracy. Through simulation, our method is effective to distinguish link failure events from link restoration events and policy related events, and reduce the size of candidate set of origins.

key words: *root cause analysis, BGP, event identifying, closed loop*

1. Introduction

Internet routing instability refers to the rapid changes of network reachability and topology information [1], and it increases the risk of packet loss and delay, even leads to loss of connectivity to several networks for prolonged periods of time. So it results in widespread degradation of the network availability and performance. With the increasing demands on fault tolerance and survivability, it is important to identify the origins of routing instability for diagnosing the failures and mitigating the impact.

Although previous works [2]–[13] have aimed to track down the origin of routing instability based on the Border Gateway Protocol (BGP) update messages, the inferred results are of low accuracy, which limits the ability to put the results into practice. The so called accuracy is referred to the probability of identifying the location where the event happens. Most of the previous works finally end up with a candidate set, which includes the possible locations to the event, and the size of which is always larger than 1. If the size of the candidate set is N, so the accuracy is defined as $1/N$. So the accuracy is lower if the size N becomes larger. If the size N equal to 1, we call the accuracy is 100%. Renata et al. has discussed that it is challenge to find the origin of instability through analyzing the BGP update data alone, since many routing changes are not visible in the BGP data and a partial view of the BGP data may lead to inaccurate results [2]. To improve the correctness, works [3]–[6] identified the location where the event happened with a candidate set which is comprised of the AS-inter links and the ASes in changed paths. However, the size of the candidate set is usually larger than 1, for the suspect set size can only be reduced to 4 on average [4]. To precisely pinpoint the location of events, work [7] only consider the link failure events, which is ideal, for recognizing events based on BGP updates is still an open challenge. Some works [8] and [9] identified the origin of events with network-wide analysis based on configurations and logs from network and BGP updates from the Internet. Unluckily, these methods can precisely pinpoint the events in network but not the ones in Internet. Some works [10] and [11] inferred the origins of routing changes using link weights, through which the events that only generate a small number of updates can not be identified. Complementarily, work [12] proposed to deal with events based on preferred path changes. Since the AS responsible for BGP routing change may not appear in the old or new AS path [2], the origin may not appear in the preferred path. Furthermore, this method only finds the origin of link related events, but can not pinpoint the location of events such as policy change. Work [13] presented a statistical approach to accurately identify familiar BGP routing instabilities and pinpoint their locations based on pattern matching. However, if the pattern is not matched, this method is helpless, and the accuracy and correctness of the results can not be guaranteed by a statistics technique.

In this paper we aim to precisely infer the origin of routing instability with high accuracy from the view of a single vantage point. Different from the previous works that only consider the stable paths after the routing is converged, we are the first to distinguish link failure events from policy change events through analyzing the relationship of the closed loops formed by intersecting all the transient paths from one vantage point during instability and the length variation of the stable paths before and after the events. Once recognizing the event that triggers the routing instability, a corresponding method is proposed to pinpoint the origin of the event. Simulation results show that the location of the event is identified with 100% accuracy if the event is recognized as a link failure event. Furthermore, as the scale

Manuscript received May 11, 2012.
Manuscript revised November 5, 2012.

[†]The author is with the School of Biomedical Engineering, Third Military Medical University and Chongqing University, Chongqing, China.

[††]The authors are with the School of Computer, National University of Defense Technology, Changsha, Hunan, China.

a) E-mail: seeker_lhj@163.com
b) E-mail: bkzhao@ nudt.edu.cn
c) E-mail: xfhu@nudt.edu.cn
d) E-mail: danzhao.nudt@gmail.com
e) E-mail: xclluu@163.com

DOI: 10.1587/transinf.E96.D.918

of topology grows, the ratio of identifying link failure events becomes higher.

2. Problem Statement

We call BGP update messages collected at the router in an AS and the router itself as BGP data and a vantage point respectively, while we term the cause which triggers routing instability as an event, and the location where the event occurs is the origin of instability. If a set includes the possible locations of the origin to an event, we refer to it as a candidate set. If the size of the candidate set is 1 and the candidate is the location where the event occurred, the event is considered to be of 100% inference accuracy. The problem to be resolved in this paper are stated as follows: Given a network $G = (V, E)$ and a set of BGP updates from single vantage point, precisely determining the cause which generates the updates and the location where the updates originate based on these updates.

To deal with the problem, we model the Internet as an undirected graph $G = (V, E)$, where V stands for the ASes and E stands for the AS-inter links (short for links) between two ASes, each of which corresponds to one BGP session. In a BGP update, the AS path $p = (v_k, v_{k-1}, \cdots, v_1, v_0)$ about prefix (destination) $d \in v_0$ is a sequence of ASes, where $(v_i, v_{i-1}) \in E$, $v_i \in V$ and v_0 is the originator of d. The number of ASes in the path is the length of the path, which is denoted as $|p|$. If $|p| = 0$, p is called as an empty path. If path satisfies the policy of "valley free" [14], we refer to p as a permitted path. If node v has many permitted paths about d, these permitted paths can be denoted as P^v. Among all the paths in P^v, each one has a preference in BGP decision process, which is denoted as $\lambda(p_i)$, $p_i \in P^v$. The path with the highest preference is the best path of node v reaching prefix d, i.e. $best(v)$. The best path which is used before and after routing instability is called old and new stable path respectively. And the paths that are selected during instability before the new stable path is found are called transient paths. The hops from AS v to prefix d along $best(v)$ are the distance of node v, which is denoted as $Dis(v)$.

To clearly present the basic concept of our approach, we model BGP as a Simple Path Vector Protocol (SPVP) [15]. In SPVP, an AS is considered to be a node, so the path advertised in SPVP is equivalent to AS path in BGP.

To precisely inferring the origins of instabilities, this paper introduces the following assumptions: (1) There is at most one session between any pair of ASes. (2) A routing instability is triggered only by one event. (3) Policy configurations satisfy "customer prefer" policy.

As BGP is modeled as SPVP and each AS in SPVP is viewed as a node, two ASes can only have one session at most. Thus assumption 1 holds and the property of Multi-Exit Discriminator (MED) does not affect the process of selecting new best path. Although many events may occur simultaneously and trigger many instabilities, there have been some works to differentiate instabilities that are triggered by different events, such as works [5], [6], [16]. Therefore, assumption 2 holds if the updates from one event are distinguished from other events using these methods, which is first task before starting our work. For space limitation, we omit this process in this paper, and more information is available in [16]. Assumption 3 ensures the location of event appear in the old or new AS paths.

3. Proposed Method

Our aim is to infer the origin of instability based on BGP data with accuracy as higher as possible. If the cause that triggers the instability is known, it helps to improve the accuracy of inferring the location of routing instability, for explicit cause can eliminate some impossible location. For example, if the cause of instability is policy change, the possible candidates must only include ASes but not any link, since policy change only occurs in an AS and any candidate set of possible origins to an event consists of ASes and links which compose AS path. To recognize the causes, we firstly classify all the events into policy-related events and link-related events, for an event only occur either in an AS or on a link in the modeled Internet. Then we identify link failure events by characterizing the instabilities through considering the relationships of the closed loops formed by intersecting all the transient paths during instability from a single vantage point and the length variation of the stable paths after the instability, and finally, we propose algorithm to pinpoint the location of different types of events.

3.1 Terminology

The cycle of selecting and propagating permitted paths is termed path exploration [17], the process of which in node v is referred to $Explore(v)$. All the selected paths in $Explore(v)$ are called exploring paths $P_{explor}(v)$, which are the transient paths during the instability and the stable paths after the instability. Thus the number of exploring paths in a path exploration process is larger than one, i.e. $|P_{explor}(v)| > 1$. Given two exploring paths $p_i \in P_{explor}(v)$ and $p_j \in P_{explor}(v)$ about prefix $d \in v_0$ in node v, there are at least two common nodes when intersecting the two paths, i.e. $\{v, v_0\} \subseteq p_i \cap p_j$. Among these nodes, if $v_m \in p_i \cap p_j$ and $v_n \in p_i \cap p_j$, the partial path between v_m and v_n in p_i and p_j are denoted as $p_i^{v_m - v_n}$ and $p_j^{v_m - v_n}$ respectively. If $p_i^{v_m - v_n} \cap p_j^{v_m - v_n} = \{v_m, v_n\}$, we call the intersected two partial paths form a *closed loop*, and v_m and v_n are the end nodes of the closed loop. If the hops from v_m to d along path p_i are smaller than that of v_n, the closed loop is denoted as $loop(v_m \rightarrow v_n)$, and $loop(v_n \rightarrow v_m)$ in vice versa. When $v_m = v_0$ in $loop(v_m \rightarrow v_n)$, if $p_i^{v_m - v_n}$ is not the part of the old stable path, i.e. $p_i^{v_m - v_n} \neq best_{before}(v_n)$, $loop(v_m \rightarrow v_n)$ contains this path, i.e. $P_{loop}(v_m \rightarrow v_n) = \{p_i^{v_m - v_n}\}$, the same as $p_j^{v_m - v_n}$. When $m \neq v_0$, this loop contains the two partial paths, i.e. $P_{loop}(v_m \rightarrow v_n) = \{p_i^{v_m - v_n}, p_j^{v_m - v_n}\}$. If another two paths form the same closed loop $loop(v_m \rightarrow v_n)$, just add the different partial paths to $P_{loop}(v_m \rightarrow v_n)$.

By intersecting all the paths from a single vantage point, many closed loops are formed, which are used to identify the nodes experiencing path exploration. Given a node $v \in V$, all the closed loops ending at v are denoted as $Loop^v = \{loop(v_i \rightarrow v) | v_i \in V, v_i \neq v\}$. If $|Loop^v| > 1$, or $|Loop^v| = 1$ and $|P_{loop}(v_i \rightarrow v)| > 1$, there is path exploration $Explore(v)$ at v.

3.2 Classifying Routing Events

According to BGP protocol, there are six path attributes in BGP update message, i.e. ASpath, NextHop, Origin, Local_Pref, MED and Community. Among them, only ASpath and Community are global that the changes of them in some place are visible far away, and the other attributes are local. So only the global attributes are useful to infer the origin of events far away from a vantage point. As a matter of fact, Community impacts Internet routing by changing BGP policy and ASpath. Thus in update messages, only ASpath is useful to analyze root causes. As ASpath is comprised of AS nodes and links, the location of these events can only be either on a link or in an AS. Consequently, all these routing events can be classified into two types:

Link-related events: if a routing event changes the state (failed or restored) of the edge between two ASes, and it results in selecting new best path, this event is called link-related event. It usually includes link failure, link restoration and session reset (restoration after failure quickly). As there are only two link states, i.e. failed and restored, the link-related events can be divided into link failure events and link restoration events.

Policy-related events: if a routing event locates in an AS, and it changes the preference of the permitted paths in this AS to select a new best path, this event is called policy-related event. The policies in BGP include input filter policies, policies in decision process and the output filter policies. As the input filter and output filter policies determine whether a path is permitted to advertise to neighbors over BGP session, these policies has the same function of link-related events. Thus we consider input filter and output filter polices as link-related events. If some ASes are added or removed in the filter at AS v, we view the edges between these ASes and v are failed or restored respectively. The policies in decision process include local preference change, ASpath length change (AS prepending increase and decrease), IGP cost change (inner-link failure and restoration, inner-link weight change), and MED. These policies belong to policy-related events. What is more, if the failure of a router influences multiple inter BGP sessions, we consider it as multiple AS-inter links failed, the same as recovery.

3.3 Distinguishing Routing Events

To distinguish link-related events from policy-related events based on BGP updates, it is vital to exploit the unique traits of the two types of events. The main idea of our method is to characterize the particular traits of path exploration under different routing events by analyzing the relationship of the closed loops and the length variation of the stable paths. The idea lies in that path exploration [17] is very common in inter-domain routing, which is the response to various events. So path exploration may conceal some particular characters which correspond to the specific type of events. Through simulation, the results show that our method distinguishes link failure events from link restoration events and policy related events effectively.

3.3.1 Definitions

For ease of exposition, we first define the relationships of the closed loops.

Definition 1: Loop containment. Given two closed loops $loop(v_1 \rightarrow v_2)$ and $loop(v_3 \rightarrow v_4)$, if the distance of the end nodes satisfies $Dis(v_1) \leq Dis(v_3) < Dis(v_4) < Dis(v_2)$ or $Dis(v_1) < Dis(v_3) < Dis(v_4) \leq Dis(v_2)$, loop $loop(v_1 \rightarrow v_2)$ is called to contain $loop(v_3 \rightarrow v_4)$, the relationship of which is denoted as $loop(v_1 \rightarrow v_2) \supset loop(v_3 \rightarrow v_4)$.

Definition 2: Loop intersection. Given two closed loops $loop(v_1 \rightarrow v_2)$ and $loop(v_3 \rightarrow v_4)$, if the distance of the end nodes satisfies $Dis(v_1) < Dis(v_3) < Dis(v_2) < Dis(v_4)$, the two loops are called intersected, and the relationship is denoted as $loop(v_1 \rightarrow v_2) \cap loop(v_3 \rightarrow v_4)$.

Taking Fig. 1 for example, the updates generated by link failure $(2, 1)$ are propagated to a vantage point, and the paths from the vantage point form three closed loops, that is $loop(2 \rightarrow 8)$, $loop(2 \rightarrow 9)$ and $loop(5 \rightarrow 9)$ which are denoted as loop 1, loop 2 and loop 3 respectively. Loop 2 and loop 3 satisfy the relationship of loop containment, the same as loop 1 and loop 2. The relationship between loop 1 and loop 3 is loop intersection.

Definition 3: First path exploration. Given some path explorations $\{Explore(v_i) | 1 \leq i \leq k\}$, if the distance of node v_h satisfies $Dis(v_h) = \min\{Dis(v_i) | 1 \leq i \leq k\}$, $Explore(v_h)$ is called as the first path exploration. And in which, v_s is termed as the first start node in the first path exploration if $Dis(v_s) = min\{Dis(v_i) | loop(v_i \rightarrow v_h) \in Loop^{v_h}, v_i \neq v_0\}$.

Taking Fig. 1 for example, the path exploration in node 8 is the first path exploration, which forms loop 1. And node 2 is the first start node.

The relationships of the closed loops reveal the information during the process of instability, and the length vari-

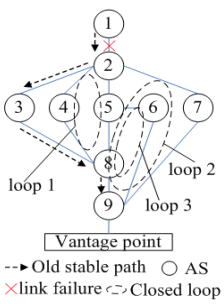


Fig. 1 Example of relationship between closed loops.

ation of stable paths discloses the information after the instability. Under a routing event, given the old and new stable paths about a prefix from a vantage point are p_1 and p_2 respectively, when the length of p_1 is longer than p_2, the length variation is denoted as $|p_1| > |p_2|$. Especially, when a prefix is withdrawn, the length of the path in this update is zero, i.e. $|p_2| = 0$. Here we consider the length variation of stable paths after the instability but not all the transient paths during instability, for the difference of update timing and propagating delays along different paths may result in the sequence arriving at a node out of order [17].

3.3.2 Rules

By correlating the relationship of the closed loops and the path length variation, we find that there are some unique traits of path exploration triggered by link failure events which distinguishes link failure events from policy events and link restoration events. In this consideration, it is vital to pay special attention to the scenario that a routing event increases the number of available paths at node which is far away from the event, i.e. $|P^v_{before}| < |P^v_{after}|$. And we call this kind of scenario as *path increasing instance*. In contrarily, i.e. $|P^v_{before}| > |P^v_{after}|$, this scenario is termed as *path decreasing instance*. As shown in Fig. 2, path $(8, 7, 4, 5, 1)$ is the only one available path at node 8 before the event. If the event is the failure of $(5, 1)$, the recovery of $(2, 1)$ or the local preference to node 3 at node 4 becomes larger than that to node 5, the best path to prefix d is changed to $(8, 6, 4, 3, 2, 1)$, and both of the two paths are available. Therefore, node 8 increases one available path. If tracing back this process, node 8 decreases one available path.

To identify events, we first provide the differentiating rules under no path increasing or decreasing instances, and then propose method to determine whether there is path increasing instances. For ease of exposition, the old and new stable path about prefix $d \in v_0$ from one vantage point are denoted as p_1 and p_2 respectively.

Theorem 1: Given a closed loop $loop(v_3 \rightarrow v_4)$ under an event, if the length variation is $|p_1| > |p_2|$, and there is no path increasing instances, when one closed loop satisfies the relationship of loop containment or loop intersection with $loop(v_3 \rightarrow v_4)$, the event is a link failure.

Proof: The proof is by contradiction. Suppose the event is a policy-related event or link restoration event. According to assumption 2, the single event generates some updates about prefix d and the policies in the ASes propagating the updates stay the same. Since there is a loop $loop(v_3 \rightarrow v_4)$, there must be route change at node v_3. Once receiving the route change, node v_4 will explore paths, or else there is only one update and loop $loop(v_3 \rightarrow v_4)$ is not formed. All the available paths P^{v_4} in v_4 before the event can be divided into two classes $P_1^{v_4}$ and $P_2^{v_4}$ according to the paths pass through node v_3 or not, and $P_1^{v_4} \cup P_2^{v_4} = P^{v_4}$, $P_1^{v_4} \cap P_2^{v_4} = \phi$.

If the new selected best path $best_{after}(v_4)$ in node v_4 passes through node v_3, i.e. $best_{after}(v_4) \in P_1^{v_4}$, the closed loop $loop(v_3 \rightarrow v_4)$ does not intersect with other loops. Since only one single event occurs before node v_3 and there is no path increasing instances during the event, policies between node v_3 and v_4 keep unchanged and the link connections do not increase. For the new selected best paths all pass through node v_3, they have the common ends v_3 and v_4, thus intersecting all the paths forms loop $loop(v_3 \rightarrow v_4)$ between v_3 and v_4. Since updates are generated before v_3, and propagated to v_3 and v_4 successively, the end nodes of the closed loops formed before v_3 do not exceed v_3. Thus $loop(v_3 \rightarrow v_4)$ does not intersect other closed loops before v_3.

If the new selected path does not pass through node v_3, i.e. $best_{after}(v_4) \in P_2^{v_4}$, the new selected best path is not affected by path advertised from v_3. As there is only one event before node v_3 and there is no path increasing instances during the event, the policies in node v_4 keep the same, and the local preferences in v_4 to its neighbors are the same, or else only one path with highest preference is selected and no loop $loop(v_3 \rightarrow v_4)$ is formed. So selecting new best path among the paths P^{v_4} is based on the attribute of shortest AS path. As $|p_1| > |p_2|$, the length variation of the best paths is $|best_{after}(v_4)| < |best_{before}(v_4)|$, which means that $\lambda^{v_4}(best_{before}(v_4)) < \lambda^{v_4}(best_{after}(v_4))$. For $best_{after}(v_4) \in P_2^{v_4}$, the path is available before the event. So there is conflict that $\lambda^{v_4}(best_{after}(v_4)) < \lambda^{v_4}(best_{before}(v_4)) < \lambda^{v_4}(best_{after}(v_4))$. ■

If the path length of and about prefix are the same, link failure events are recognized by theorem 2.

Theorem 2: When a routing event happens, if the path length variation is $|p_1| = |p_2|$, and there is no path increasing and decreasing instance, when one closed loop $loop(v_1 \rightarrow v_2)$ is formed, and $v_1 \in p_1$, $v_1 \neq v_0$, the event is a link failure.

Proof: The proof is by contradiction. Suppose the event is a policy-related event or link restoration. As there is a loop $loop(v_1 \rightarrow v_2)$, the updates about prefix d generated by the event will be propagated to node v_2, and v_2 will explore the available paths to select a new best path. Since there is no path increasing and decreasing instance, v_2 explores paths from the same neighbors before the event. According to assumption 2, the single event generates some updates and the policies in ASes propagating the updates stay the same. The local preferences of v_2 to its neighbors are the same, or else v_2 will select the route from the same neighbor who provides the old stable path to prefix d as the new best path, and thus there is only one update, which is contradicted to that there

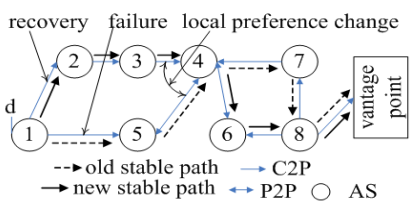


Fig. 2 Path increasing instance.

is a loop $loop(v_1 \rightarrow v_2)$. Therefore, the factor to influence selecting new best path is the attribute of shortest AS path. As the length variation is $|p_1| = |p_2|$ and the policies and connections in v_2 keep unchanged, node v_2 still selects the route from the same neighbor which provides the old stable path as the new best path. Consequently, node v_2 only chooses one path, and there is no path exploration. So there is only one path received in vantage point, and there is no closed loop, which is contradicted to the conditions. ■

When path length becomes longer, i.e. $|p_1| < |p_2|$, both link-related events and policy-related events can have loop containment and intersection. So it is necessary to explore other method to differentiate them. As link failure leads to the withdrawn paths that pass through the failed link, and policy-related events lead to advertise new path to update the existing path, we find that the path between the prefix and the *first start node* in the *first path exploration* illustrated in definition 3 is different between link-related event and policy related event, which is illustrated in theorem 3.

Theorem 3: Under an event, the first path exploration on prefix $d \in v_0$ is $Explore(v)$ and the first start node is v_r. If there is only one path $p(v_r, \cdots, v_0)$ from v_r to v_0 in the first path exploration, when $p(v_r, \cdots, v_0) = best_{before}(v_r)$, the event is a link failure event.

Proof: The proof is by contradiction. When there is only one path from v_r to v_0 in the first path exploration, and $p(v_r, \cdots, v_0) = best_{before}(v_r)$, suppose the event is policy-related or link restoration event. According to definition 3, node v explores paths when the event happens, for $Explore(v)$ is the first path exploration. The event causes node v_r to select a new best path $best_{after}(v_r) \neq best_{before}(v_r)$ to reach d, otherwise node v_r will not propagate updates to node v, and there is no path exploration in v. The exploring paths from node v to prefix d must contain the partial path $p(v_r, \cdots, v_0)$, for the path exploration is the response of advertising the new best path $p(v_r, \cdots, v_0)$ from v_r to v_0 and $p(v_r, \cdots, v_0)$ is the only path from v to prefix d among all the received paths. As a result, $p(v_r, \cdots, v_0)$ meets $p(v_r, \cdots, v_0) = best_{after}(v_r) \neq best_{before}(v_r)$, which is conflicted to the condition that $p(v_r, \cdots, v_0) = best_{before}(v_r)$. As a result, the event is a link failure. ■

If there is more than one path from v_r to v_0 in the first path exploration, it is difficult to distinguish link failure events from link restoration events and policy-related events with the help of path exploration. It is necessary to explore new method to differentiate them. In conclusion, besides of this situation, it is certain to distinguish link failure events from link restoration events and policy-related events based on the theorems 1, 2 and 3.

It is noticeable that theorem 1 and 2 hold when path increasing and decreasing instance are taken into account. It is important to identify whether these instances occur if putting these theorems into practice.

Given two paths p_1 and p_2 from v to prefix $d \in v_0$, the partial paths that belong to p_1 but p_2 are denoted as $sub(p_1/p_2) = \{p_1^{v_i-w_j} | p_1^{v_i-w_j} \in p_1, p_1^{v_i-w_j} \notin p_2\}$, where $\{v_i, w_j\} \in p_1$ and $\{v_i, w_j\} \in p_2$; likewise, the partial paths that belong to p_2 but p_1 are denoted as $sub(p_2/p_1)$. If $Dis(w_h) = \{\min Dis(w_j) | w_j \in p_1^{v_i-w_j}, p_1^{v_i-w_j} \in sub(p_1/p_2)\}$, and $w_h \neq v \neq v_0$, w_h is called as the *nearest partial path node* among all the partial paths. With the partial paths of the stable paths, it is useful to identify path increasing instance, which is shown in theorem 4.

Theorem 4: Under one single event, given the old and new stable path of prefix $d \in v_0$ from node v is p_1 and p_2, if $|p_1| > |p_2|$ and the nearest partial path node is w, when $|p_1^{w-v}| \geq |p_2^{w-v}|$, there is path increasing instance.

Proof: The proof is by contradiction. Suppose there is no path increasing instance. Thus path $p_1^{v_0-w} + p_2^{w-v}$ is available at node v, for p_2 is the new selected best path to replace p_1. As $|p_1| > |p_2|$ and $|p_1^{w-v}| \geq |p_2^{w-v}|$, the length of the available path is $|p_1| \geq |p_1^{v_0-w} + p_2^{w-v}|$. If the local preference of v to its neighbors along p_1 is higher than that along p_2, node v will not select p_2 as the new best path, for the connections and policies of v keep the same under a single event. This is contradicted to the condition. If the local preferences are the same, selecting best path is based on shortest path length, for the length variation is $|p_1| > |p_2|$. As $|p_1| \geq |p_1^{v_0-w} + p_2^{w-v}|$, p_1 will not be selected as the old stable path, which is a conflict. In conclusion, there is path increasing instance under the condition. ■

Theorem 5: Given the old and new stable path of prefix $d \in v_0$ from node v is p_1 and p_2, if $|p_1| = |p_2|$ and there is the nearest partial path node is w, there is path increasing instance or path decreasing instance.

Proof: As node w is the nearest partial path, p_1 and p_2 are divided into four partial paths, $p_1^{v_0-w}$, p_1^{w-v}, $p_2^{v_0-w}$ and p_2^{w-v}, which are denoted as a_1, a_2, b_1 and b_2 respectively. In other words, $a_1 + a_2 = p_1$ and $b_1 + b_2 = p_2$.

If $|a_1| < |b_1|$, the length of a_2 and b_2 satisfies $|a_2| > |b_2|$, for $|p_1| = |p_2|$. As a result, $|a_1 + b_2| < |a_1 + a_2|$. For p_1 is selected as the best path before the event, the possible reason is the local preference of v along a_2 is higher than that along b_2 if $a_1 + b_1$ is a permitted path; or $a_1 + b_2$ is not permitted, but the preference of v along b_2 is higher than that along a_2. In the former scenario, $b_1 + a_2$ will be selected as the new best path, which is a conflict that the new stable path is p_2. The explanation is that $b_1 + a_2$ is not permitted by routing policy. So the path containing a_2 is eliminated from the available paths of v, which is a path decreasing instance. In the latter scenario, if $b_1 + b_2$ is permitted, it is selected as new best path, which results from the higher local preference. So the paths that contain b_2 become available, which are path increasing instances.

Likewise, the situations of $|a_1| = |b_1|$, $|a_2| = |b_2|$ and $|a_1| > |b_1|$, $|a_2| < |b_2|$ have the same results. ■

3.4 Locating the Origin of Instability

The main idea to infer the origins of events is to firstly abstract the candidate set of the origins to the instability by co-

operating with the explicit and implicit information in BGP updates, and then narrow down the set based on the distinguished type of the event.

3.4.1 Inference of Candidates

If the path of prefix d to a vantage point is changed by an event, some updates reflecting the route change of prefix d are generated. These updates are referred as the *explicit information*. The lack of any BGP update message is another information source. It indicates that the current best path is stable and does not suffer any instability [5]. We call this lack of updates as *implicit information*. Collaborating with the explicit and implicit information can abstract the candidates of origins. Taking Fig. 3 for example, the candidates abstracted by clustering across prefixes in [4] based on the explicit information are $\{2, 3, 4, 5, (2, 3), (3, 4), (4, 5)\}$. Since there are no updates reflecting the prefixes in node 3, 4 and 5, the event does not locate in these nodes and the links between them. Thus the candidate set is reduced to be $\{2, 3, (2, 3)\}$ through implicit information.

With the explicit and implicit information, the origin of event is limited to a small scale. As the AS responsible for a routing change appears in the old stable path, the new stable path, or both [3]–[5], intersecting the old and new stable paths will form the similar structure which is shown in Fig. 4. The prefix d is the nearest prefix to the vantage point among the prefixes that are affected by the event. And the dashed line between node 3 and 5 means zero or at least one node connected by link, and the same as the dashed line between node 2 and 4. If there is path exploration in instability, node 6 is the first start node as defined in definition 3. If there is no path exploration in the instability, node 6 is the nearest intersected node to prefix d. When the new stable path is null, which means that the prefix d is withdrawn, we consider the ASes and the links which comprise of the old stable path as the candidates. Under the assumption in this paper, the inferred candidate set includes one failed link (such as $(1, 2)$), one restored link (such as $(1, 3)$), and some

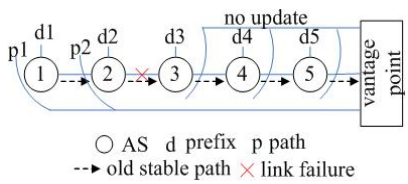


Fig. 3 Rule of abstracting candidates.

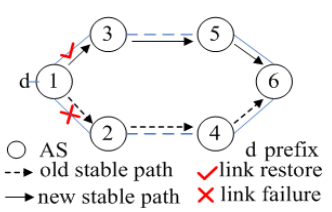


Fig. 4 Location of possible candidates.

ASes (such as 1, 2, 3, 4, 5 and 6), as shown in Fig. 4.

3.4.2 Infer the Origin of Link Failure Events

When the event triggering instability is identified as a link failure event, the location of event is a failed link. As the candidate set includes only one failed link, the failed link is the origin of event. Thus the size of the candidate is 1, and its correctness is discussed in Sect. 5. So we can precisely infer the origin of event with 100% accuracy when the event is identified as link failure event. Once the event is not identified, it is hard to narrow down the candidate set based on single vantage point. Finally, the inferred candidates contain one resorted link, one failed link, some ASes. As the least ASes to form the cycle like Fig. 4 is 3, the smallest size of candidate set is 5 when the event is not identified as link failure event from single vantage point.

4. Simulation

Among the simulators for BGP, we choose SSFNET [18] to generate events. In our simulations, the parameters are set as follows: the MRAI is 30 seconds, the import and export policy is based on "Gao-Rexford" policy, and the other settings are set based on standard BGP.

We perform a large number of simulation runs using Internet-like topology, which is generated as follows: an AS-level topology of the Internet mapped by CAIDA [19] from April 29 2009 has 31212 ASes and 60052 links. Because of the great scale of the topology, we adopt Dimitropoulos [20] to generate small scale Internet-like policy-annotated AS graph. In this paper, three topologies are generated to validate our method, which respectively have 200 ASes and 359 links, 600 ASes and 1124 links, and 800 ASes and 1582 links. As each AS is monitored by a vantage point, the BGP data from each of the vantage points is used to distinguish events.

To validate the method in distinguishing link failure events from policy-related events, 718 simulation runs of policy-related events are performed using topology of 200 ASes. As the change of local preference is a typical policy-related event, in each run, we randomly change the local preference of an AS to its neighbor. From the view of eight randomly selected vantage points that monitor ASes with different degree, i.e. the number of directly connected neighbor ASes, our approach does not identify them as link failure events mistakenly, which is shown in Table 1. For example, AS 35 has 101 neighbors, and it observe 192 events changing the routes in it among the 718 simulation runs, which accounts for 0.27. Analyzing the updates generated by the 192 events respectively, no event is mistakenly identified as link failure event through our method, i.e. False Num is 0. Thus the falsely ratio of identifying event types is 0. Across the eight monitored ASes, our approach have the similar effect in identifying events, for the false percent of the identified events in each AS is 0. This implies that our approach can distinguish link failure events from policy-related events.

Table 1 Identifying policy-related events.

AS	degree	runs	useful runs	useful percent	false num	false percent
178	2	718	193	0.27	0	0.00
179	15	718	199	0.28	0	0.00
35	101	718	192	0.27	0	0.00
47	1	718	199	0.28	0	0.00
56	62	718	180	0.25	0	0.00
62	13	718	199	0.28	0	0.00
73	2	718	192	0.27	0	0.00
82	2	718	189	0.26	0	0.00
average	25	718	193	0.27	0	0.00

Table 2 Identifying link restoration events.

AS	runs	useful runs	useful percent	false num	false percent	error identifies	correct percent
178	359	194	0.54	0	0.00	1	0.99
179	359	199	0.55	0	0.00	2	0.99
35	359	192	0.53	0	0.00	2	0.99
47	359	198	0.55	0	0.00	0	1.00
56	359	187	0.52	0	0.00	0	1.00
62	359	198	0.55	0	0.00	1	0.99
73	359	192	0.53	0	0.00	1	0.99
82	359	192	0.53	0	0.00	0	1.00
average	359	194	0.54	0	0.00	0.88	0.99

Table 3 Identifying link failure events.

AS	runs	useful runs	useful percent	identify events	identify percent	correct percent	result size	compare size
178	359	180	0.50	24	0.13	1.00	4.98	6.99
179	359	196	0.55	66	0.34	1.00	4.39	7.18
35	359	131	0.36	16	0.12	1.00	4.71	6.18
47	359	179	0.50	22	0.12	1.00	5.38	7.55
56	359	109	0.30	6	0.06	1.00	5.17	6.80
62	359	169	0.47	22	0.13	1.00	5.07	6.92
73	359	130	0.36	14	0.11	1.00	4.90	6.44
82	359	144	0.40	20	0.14	1.00	5.49	8.24
average	359	155	0.43	24	0.15	1.00	5.00	7.08

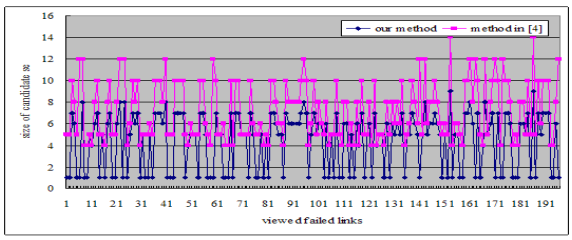


Fig. 5 The size of candidates per each viewed run in AS 179.

Under link restoration events, our method does not mistakenly recognize them as link failure events, which is shown in Table 2. For example, among the 359 simulation runs in the monitored AS 179, each of which is a link restoration event, only 199 runs affect the routes in AS 179, which accounts for 0.55. Among the 199 events, no one is identified as link failure events, i.e. False Num is 0. So the false percent of identifying event types is 0. Since the events are not identified as link failure events, the origin of the events is a candidate set. However, there are 2 runs that the candidate set does not contain the right location where is link restored. Thus the correct percent of inferring the origins of link restoration events is 0.99, which implies that it is rational to assume the location of link restoration events lying in the stable paths. This illustrates that the restored link may not appear in the stable paths, which may lead to error diagnosis.

Under link failure events, our approach identifies about 15% link failure events among the 359 simulation runs on average when using the topology of 200 ASes, each of which is a link failure event, and the origins of the identified events are inferred with 100% accuracy, which is shown in Table 3. Taking the monitored AS 179 for example, among the 196 events which affect the routes in this AS, 66 events are identified as link failure events, which accounts for 34%. The origin of the identified event is precisely inferred to the single link where it is failed, thus the size of the candidate set to these events is 1, so the accuracy ratio of inferring the identified events is 100%, i.e. correct percent is 1.00. This shows that our method can precisely infer the origin of link failure events with 100% accuracy.

Although only part of the events are precisely inferred, our method largely reduces the size of the candidate set comparing to the approach in [4]. For example, under our method, the average size of candidate set in AS 179 among the 196 events is 4.39, and that of method in [4] is 7.18, i.e. result size is 4.39 and compare size is 7.18. Among the 196 runs, the specific size of the inferred candidate set to each useful runs is shown in Fig. 5. If the size is 1, it means that this event is identified as link failure event, and the origin is precisely inferred with 100% accuracy.

Among the 8 ASes, the ASes with smaller degree are of higher proportion to identify link failure events, such as AS 56 with degree of 62 has 6% identify ratio, but AS 82 with degree 2 has 14% identify ratio. This is because the changed routes in edge ASes usually pass through the core ASes in network, so the routes in edge ASes reflect the more connections in the network than that of core ASes. Thus the routes in edge ASes are more likely to satisfy the identifying criteria, such as theorem 1, 2 and 3.

The average percent of identifying link failure events is only 15%, which is seemly not attractive. The reasons underlying this result are following. Firstly, identifying link failure event from various events based on BGP updates is challenge, especially the updates from one single vantage point. Secondly, our work is the first to distinguish link failure events from other events. The previous works have limited ability to recognize the type of the events which trigger routing instability, which means that the "identify percent" is 0 in previous works. Thus the identify percent improved from 0 to 15% is a large step. Thirdly, the topology with 200 ASes is small scale, so the connections of which is limited relative to Internet. Limited connection can not make full use of our criteria. So we perform the same simulations on identifying different events using topology of 600 and 800 ASes respectively, and the identifying ratios under which are shown in Fig. 6.

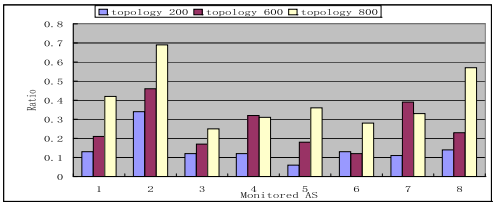


Fig. 6 Identifying ratio under different topology.

As shown in Fig. 6, 8 monitored ASes in each topology are selected based on representative different degree. Under topology 600 and 800 ASes, 1124 and 1582 simulation runs are performed for each monitored AS respectively. And each run corresponds to one link failure event. In average, the ratio of identifying link failure events is 24% in topology 600 ASes, and 40% in topology 800 ASes. This shows that our approach can get higher identifying ratio as the scale of topology grows.

5. Discussion

When inferring the candidates of possible origins to an event, our method builds on the assumption that the AS responsible for a routing change appears in either the old stable path, the new stable path, or both [3]–[5]. However, Renata et al. in [2] have discussed that this assumption is not always true in some situation. As we have classified the events into policy-related events and link-related events, and link failure events are distinguished from other events, we prove that the origin of link failure events is located in the old stable path as shown in theorem 6. Thus the assumption in our method is true when inferring the location of link failure events, and so the inferred origin of the link failure event is the location where the event happens. Thus, our method can narrow down the size of the candidate set to 1, and the candidate is the correct location of the event. So the origin is precisely inferred with 100% accuracy.

Theorem 6: If a link failure event generates many BGP updates about prefix $d \in v_0$, and the updates are received at vantage point o_1, the event locates in the old stable path from o_1 to d when the old stable path is not null.

Proof: The proof is by contradiction. Suppose the single link failure event locates in path p_0 from prefix $d \in v_0$ to vantage point o_1, and p_0 is different from the old stable path p_1, where $|p_1| \neq 0$. So p_1 is the best path from o_1 to d, and the preference of p_1 is higher than that of path p_0 before the link failure event. When a link failed in p_0, the failure will make path p_0 unavailable. However, link failure event can not increase the preference of a path. So this failure event can not make some path become more preferable than the old stable path p_1. According to the BGP decision process, there will be no BGP update advertised if the best path stays unchanged, which is conflicted to that some BGP updates are received. ■

To infer the candidate set of origins to events, we suppose that if there is no update reflecting prefix d received at a vantage point under a routing event, the event will not locate in the path from the vantage point to prefix d. The route flap damping (short for RFD) [21] technique does not affect the correctness of assumption. As illustrated in [22], there are at least four updates advertised before the updates are aborted by RFD. Therefore, RFD mechanism can not damp all the updates generated by one event, and so our assumption holds in inferring candidate set.

6. Conclusion

In this paper, we have proposed a novel method to diagnose the routing instability through characterizing the transient paths from the view of a single vantage point. To improve the accuracy of inferring the origin of instability, we are the first to distinguish link failure events from policy-related events and link restoration events by analyzing the closed loop relationship in path exploration and the length variations of stable paths after the events. After identifying the routing event, we provide an algorithm to infer the origin of routing instability. The simulation results show that our approach effectively distinguishes link failure events from other events and precisely infers the origin of link failure events with 100% accuracy. However, our method can not deal with the scenario of no path exploration in updating messages. It is necessary to introduce other approach to distinguish them, such as link weight change, which is our future work in the next step. For example, the link weight change triggered by link-related event may be observed in opposite direction, but the weight change triggered by policy-related event can only be observed from the same direction.

Acknowledgments

The work described in this paper is supported by the grants of the project of National Science Foundation of China under Grant No.61103189, No.61202488 and No.61070199; and Hunan Province Natural Science Foundation of China (11JJ7003), and the Program for Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province: "network technology", Changjiang Scholars and Innovative Research Team in University (No.IRT1012), and the Doctoral Program of Higher Education of China uner Grant No.20124307120032.

References

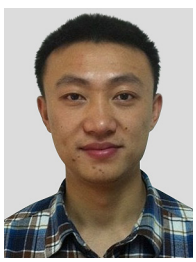
[1] C. Labovitz, G.R. Malan, and F. Jahanian, "Internet routing instability," IEEE/ACM Trans. Netw., vol.6, no.5, pp.515–528, Oct. 1998.
[2] R. Teixeira and J. Rexford, "A measurement framework for pinpointing routing changes," Proc. ACM SIGCOMM Workshop on Network Troubleshooting, Aug. 2004.
[3] D-Fa Chang, R. Govindan, and J. Heidemann, "The temporal and topological characteristics of BGP path changes," Proc. IEEE International Conference on Network Protocols (ICNP), Nov. 2003.
[4] M. Caesar, L. Subramanian, and R. Katz, "Towards localizing root causes of BGP dynamics," Technical Report UCB/CSD-04-1302, U.C. Berkeley, Nov. 2003.

[5] A. Feldmann, O. Maennel, Z.M. Mao, A. Berger, and B. Maggs, "Locating Internet routing instabilities," Proc. ACM SIGCOMM, pp.205–218, Portland, OR, Aug. 2004.
[6] T. Ogishi, Y. Hei, S. Ano, and T. Hasegawa, "Empirical study on inferring BGP routing instability and its location based on single point observation," ICC 2007.
[7] M. Lad, A. Nanavati, D. Massey, and L. Zhang, "An algorithmic approach to identifying link failures," Proc. 10th IEEE Pacific Rim International Symposium on Dependable Computing (PRDDC), pp.25–34, March 2004.
[8] J. Wu, Z.M. Mao, J. Rexford, and J. Wang, "Finding a needle in a haystack: Pinpointing significant BGP routing changes in an IP network," Proc. Networked Systems Design and Implementation (NSDI), May 2005.
[9] Y. Huang, N. Feamster, A. Lakhina, and J. Xu, "Detecting network disruptions with network-wide analysis," Proc. ACM SIGMETRICS, 2007.
[10] M. Lad, R. Oliveira, D. Massey, and L. Zhang, "Inferring the origin of routing changes using link weights," Proc. IEEE ICNP, Oct. 2007.
[11] A. Campisano, L. Cittadini, G. Di Battista, T. Refice, and C. Sasso, "Tracking back the root cause of a path change in inter-domain routing," Proc. IEEE/IFIP NOMS, April 2008.
[12] M. Watari, A. Tachibana, and S. Ano, "Inferring the origin of routing changes based on preferred path changes," PAM 2011.
[13] W. Liang, Y. Li, J. Bi, and G. Zhang, "On the accurate identification of familiar inter-domain routing instabilities," GLOBECOM 2008.
[14] L.X. Gao, "On inferring autonomous system relationships in the Internet," IEEE/ACM Trans. Netw., vol.9, no.6, pp.733–745, 2001.
[15] T.G. Griffin, F.B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," IEEE/ACM Trans. Netw., vol.10, no.2, pp.232–243, 2002.
[16] K. Xu, J. Chandrashekar, and Z.L. Zhang, "A first step toward understanding inter-domain routing dynamics," ACM SIGCOMM MineNet Workshop, Aug. 2005.
[17] R. Oliveira, B. Zhang, D. Pei, and L. Zhang, "Quantifying path exploration in the Internet," IEEE/ACM Trans. Netw., vol.17, no.2, pp.445–458, April 2009.
[18] "SSFNET: Scalable Simulation Framework," http://www.ssfnet.org, 2007.
[19] Caida, The CAIDA AS relationships dataset, AS-rel.20090429.a0.01000.txt [Z]. http://www.caida.org/data/active/as-relationships, 2009.
[20] X. Dimitropoulos, D. Krioukov, A. Vahdat, and G.F. Riley, "Graph annotations in modeling complex network topologies[J]," ACM Trans. Modeling and Computer Simulation (TOMACS), vol.19, no.4, pp.1–2, 2009.
[21] Z. Mao, R. Govindan, G. Varghese, and R. Katz, "Route flap damping exacerbates Internet routing convergence," Proc. ACM SIGCOMM, pp.221–233, 2002.
[22] C. Panigl, J. Schmitz, P. Smith, and C. Vistoli, "RIPE routing-WG recommendations for coordinated route-flap damping parameters," Document ID: ripe-229, Oct. 2001.

Hongjun Liu received his Ph.D. degree in computer science from National University of Defense Technology, China, in 2012. He is currently an Assistant Professor in the School of Biomedical Engineering, Third Military Medical University and Chongqing University. His current research interest includes Internet measurement, network survivability and inter-domain routing.

Baokang Zhao received his B.S. and Ph.D. degrees from the National University of Defense Technology, both in Computer Science. Currently, he is an Assistant Professor in the School of Computer Science, National University of Defense Technology. His current research interests include protocols, algorithms, and security issues in computer networks. Dr. Baokang Zhao has also served as a reviewer for several journals, including Computer Communications (Elsevier), Security and Communication networks (Wiley), Journal of Computer Science and Technology (Springer), etc.

Xiaofeng Hu received his Ph.D. degree in computer science from National University of Defense Technology, China, in 2004. He is currently an associated professor in the School of Computer at the same university. His research interest includes Internet architecture, routing protocol, and high performance router design.



Dan Zhao is currently a Ph.D. student in the School of Computer of National University of Defense Technology in Changsha, China. He received the B.S. and M.S. degree in computer science in the same university, in 2006 and 2008 respectively. His research interest includes network architectures, Internet routing and protocols.

Xicheng Lu received his B.S. degree in computer science from Harbin Engineering Institute, Harbin, China, in 1970. He was a visiting scholar at the University of Massachusetts from 1982 to 1984. He is currently a professor with School of Computer Science of National University of Defense Technology (NUDT), Changsha, China. His research interests include distributed computing, computer networks, and parallel computing. He has served as a member of editorial boards of several journals and has co-chaired many professional conferences. He is an academician of the Chinese Academy of Engineering and a member of the IEEE.