

Locating Internet Instability under Simultaneous Events

Hongjun Liu^{1, a}, Xiaofeng Hu^{1, b}, Dan Zhao^{1, c}, and Xicheng Lu^{1, d}

¹School of Computer, National University of Defense Technology, Changsha, Hunan, China

^aseeker_lhj@163.com, ^bxfhu@nudt.edu.cn, ^cdanzhao.nudt@gmail.com, ^dxclluu@163.com

Keywords: BGP, routing instability, instability origin, simultaneous events

Abstract. Locating internet instability is very important to diagnose internet problems. The existing methods are under the assumption that instability is triggered by only one event, and these methods are not applicable for the scenario of simultaneous events. This paper presents the first study on characterizing the simultaneous events, finding out where the multiple events happened could be visible and how many events are simultaneously happened. Furthermore, a novel scheme is proposed to accurately pinpoint the origins of instability under simultaneous events by exploring cycles, which is theoretically proved to be feasible.

1. Introduction

Internet routing instability refers to the rapid changes of network reachability and topology information [1]. Instability increases the risk of packet loss and delay, even leads to loss of connectivity to several networks for prolonged periods of time. Eventually, it will result in widespread degradation of the network availability and performance. With the increasing demands on fault tolerance and survivability, it is crucial important to be able to identify the origins of routing instability. This ability would immensely help in diagnosing the failures and estimating the impact.

A lot of previous works have done to pinpoint the BGP routing instability [2-14] in the last few years. All of these works are under the assumption that each of instability is triggered by only one event. Therefore their targets are to find only one origin per instability. As a matter of fact, given the vast size of the Internet and the high rate of routing events, multiple routing events may simultaneously affect routes to the same prefix. This may cause route advertisements triggered by many events to overlap in time [4]. Especially the large scale natural disasters, such as earthquake, would trigger a large number of simultaneous events. In these scenarios, it is of great necessary to accurately pinpoint all the events for diagnosing and recovering. Utilizing the existing methods to locate the origins of instability may result in inaccurate conclusion. Although Ref. [4] has mentioned this problem, it omits this scenario and infers the origin of instability under one event. To our best knowledge, we are the first to characterize the simultaneous events in a routing instability and locate the origins under this scenario.

To summarize our contributions: this paper presents the first known study of characterizing where the simultaneous events happened could be visible and how many events have happened at the same time in a routing instability in section 2. Then a novel scheme is proposed to locate the origins of instability by exploring cycles under simultaneous events in section 3.

2. Characterizing Simultaneous Events

The simultaneous events referred in this paper are the ones visible to the vantage point. The so called visible means that the vantage point receives and sends path advertisements that reflect the path changes caused by the event. The simultaneous events could be a link failure or restoration, router failure, BGP policy change, BGP session reset, and so on. To understand simultaneous events, it is challenge to find out where the simultaneous events happen could make them be visible and identify how many events have simultaneously happened.

2.1 Where Could the Visible Simultaneous Events Happen

As specified in BGP protocol, only the change of the current best path could trigger update message and propagate them over through internet. This trait could be described by theorem 2.1.

Theorem 2.1: If there is the best path π of the time to reach prefix p from a vantage point, the BGP updates about the change of π must be observed by the vantage point.

Proof: The proof is by contradiction. Given $\pi = (as_n, as_{n-1}, \dots, as_0)$, where as_n is the location of vantage point and as_0 is the originator of prefix p . If the BGP updates about the change of π are not received by the vantage point, it is implied that the update is absorbed by some AS as_i in the spreading process, which means as_i has not advertised this update to its neighbor. According to the BGP protocol, BGP will propagate the update message only when the best path from as_i to as_0 have changed. The update absorption of as_i stands for the best path remains unchanged. The only possible reason is that the change is not on the best path. As a result, this is conflicted with the condition that π is the current best path and has changed.

Based on theorem 2.1, if simultaneous events are visible, it is indispensable that the events should locate on the paths that are the current best path and the ones to be selected as the best path under the effect of simultaneous events. This is expressed in theorem 2.2.

Theorem 2.2: If there are n alternative paths $\{\pi_i | 0 \leq i < n\}$ when simultaneous events happen, and each path has a preference value $pref(\pi_i)$ of the time. If the events make some new paths $\{\pi_{new}\}$ available, only $\max\{pref(\pi_i)\} < \min\{pref(\pi_{new})\}$ could make these events be visible to the vantage point. Otherwise, only when the events locate on the adjacent paths with preference value monotonously descended starting from path π_k with $pref(\pi_k) = \max\{pref(\pi_i)\}$, the simultaneous events could be visible.

Proof: The proof is by contradiction. Suppose some events happen and induce some new paths available, they are not visible to observer. As these events make paths $\{\pi_{new}\}$ become available again, these paths will have their own preferences $pref(\pi_{new})$ according to the BGP policy process. As the assumption holds, these events are not visible, which means that all the new available paths are less preferred to the current best path, so it is concluded that $\max\{pref(\pi_{new})\} < \max\{pref(\pi_i)\}$. This is conflicted to $\max\{pref(\pi_i)\} < \min\{pref(\pi_{new})\}$.

Otherwise, these events don't create new available paths, and they are not visible. This implies that the simultaneous events don't make some path $\pi_h, h \neq k$ acquire the highest preference, so π_k is still the best path staying unchanged, and $pref(\pi_k) = \max\{pref(\pi_i)\}$. This conflicts to the situation that the highest preferences are monotonously descended.

Theorem 2.2 points out that only when an event make the highest preference changed among all alternative paths and the current best path can it be visible. Thus, if the simultaneous events are visible, they are all hidden in the updates of the affected prefixes. So analyzing the updates is a useful way to identify simultaneous events.

2.2 Identifying the Simultaneous Events

When an event happens and it is visible, the vantage point will receive at least one update, which reflects a new valid path to detour the event or the event make the prefix unreachable. The so called valid path means that it is available and the failure event is not located on the path. It is unique per one event, as BGP only advertises the new best path. This is differentiated from the invalid paths in path exploration process [15]. As shown in figure 1, the failure of link $l(1 \leftrightarrow 2)$ induces path exploration. All the paths listed in figure 1 contain the failure, so they are invalid. On considering the old best path before failure and the new valid path or the withdraw update of a certain prefix, we refer to the old best path combines the new valid path or the withdraw update as forming a *cycle*. As the new valid path or withdraw update is unique per one event for a prefix, the formed *cycle* is unique per event. As a result, identifying the events through exploring cycles is a feasible method.

Because each cycle is corresponding to an event one by one, it is intuitionistic to find out the simultaneous events by exploring the cycles in updates. To form a cycle, it is primary to find the new valid path. Especially, when simultaneous events happen, it is critical to find all the new valid paths bypassing the events.

A. Shedding light on the new valid paths

Given the current best path of prefix p is $\pi_1 = (as_n, \dots, as_1, as_0)$, where as_n is the location of vantage point and as_0 is the originator of prefix p . When one link $l(as_{k+1} \leftrightarrow as_k)$ failed, path exploration will advertise many transient paths to detour this failure until a new valid path is found. This implies these transient paths don't actually bypass the failure link $l(as_{k+1} \leftrightarrow as_k)$, otherwise the last one of the transient path would be the new valid path. As a result, all the transient paths would have the common part $\pi_2 = (as_{k+1}, \dots, as_1, as_0)$ that contains the failure link. We refer to the common part as *subpath*. So if many paths have the common subpath, these paths would share a common link failure. The path adjacent to the last transient path is the new valid path.

As shown in figure 1, there may be two subpaths (3,2,1) and (2,1). It is critical to select which of them as the subpath, as different selection would result in different new valid path. For example, if taking both of them as subpaths, there would be new valid paths (6,5,2,1); if only taking (2,1) as the subpath, there would be no new valid path, and finished with a withdraw update. The former scenario would form two cycles, which corresponding to cycle 1 and cycle 2, and the latter would form only one cycle, which corresponding to cycle 2. According to theorem 2.1, there is no update of prefix p_2 , which means that no failure locates on the current best path (6,3,2) of prefix p_2 . So we could only choose (2,1) as the subpath. Thus only one cycle is formed, which is correctly corresponding to the unique link failure $l(1 \leftrightarrow 2)$.

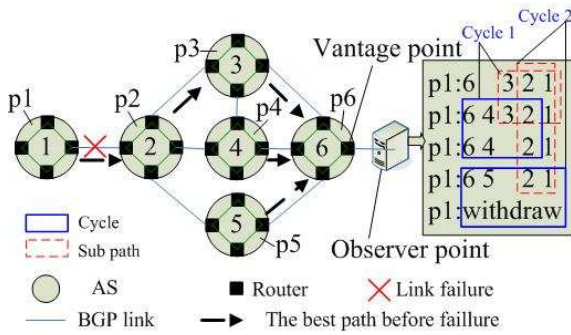


Fig.1. Path exploration under an event

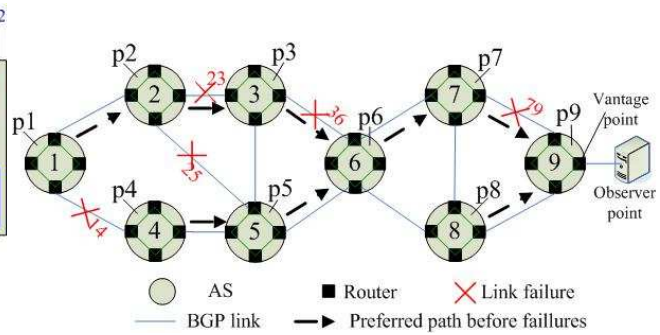


Fig.2. Locations of simultaneous events

B. Identifying the simultaneous events

If there are n simultaneous events happen, it is necessary to find n new valid paths or $n-1$ valid path and a withdraw update. The locations where the events happen would affect the identifying of these events. This is illustrated by figure 2.

As shown in figure 2, multiple events may occur on the current best path simultaneously, such as $l(7 \leftrightarrow 9)$ and $l(2 \leftrightarrow 3)$ or $l(2 \leftrightarrow 3)$ and $l(3 \leftrightarrow 6)$. In the former scenario, two cycles will be formed, each of which corresponding to each link failure. The two cycles shown in figure3 correspondingly have no common AS. This implies that if two cycles only have at most one common AS, it is certain that there are at least two events have happened at the same time. In the latter situation, there will form one or two cycles. When AS 6 selects path (6,5,2) to bypass link $l(3 \leftrightarrow 6)$, there is only one cycle. When AS 6 first selects path (6,5,3) to bypass link $l(3 \leftrightarrow 6)$, there will be two cycle. This implies that if some link events happened on the current best path, and they are adjacent to each other, only part of these events would be visible. The number of formed cycles is constrained by the topology and BGP policy. In this situation, the number of cycles may not completely correspond to the number of events. For example, in the scenarios of link failure $l(2 \leftrightarrow 3)$ and links failure $l(2 \leftrightarrow 3)$, $l(3 \leftrightarrow 6)$ in figure 3, the identified cycles are the same. If link events simultaneously located on both the current best path and the alternative paths, such as $l(2 \leftrightarrow 3)$ and $l(2 \leftrightarrow 5)$, there will form two cycles. If $l(1 \leftrightarrow 4)$ is further failed, the observer will receive a withdraw update. These are illustrated by the corresponding indentifying in figure 3.

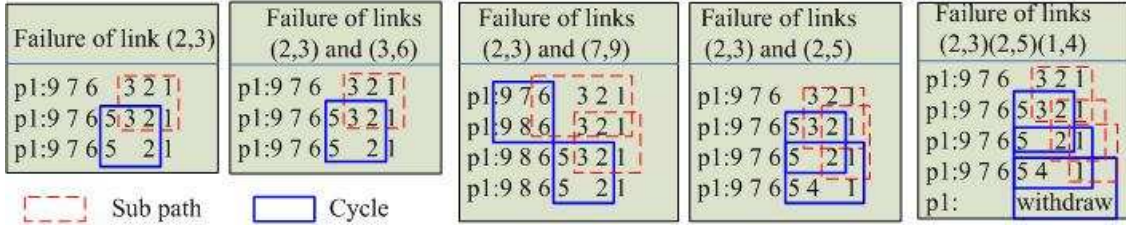


Fig.3. Identify the simultaneous events by cycles

Through the former discussion, we could deduce the following conclusion:

Theorem 2.3: For all the updates of prefix p in a routing instability, if the paths in the updates form n cycles, there are at least n events happened simultaneously.

Proof: As this result is concluded from the previous discussion, the specific proof process is omitted.

3. Pinpointing the Origins of Instability under Simultaneous Events

Although it is difficult to accurately point out how many events are exactly simultaneously happening, we could assert the approximate number of simultaneous events according to theorem 2.3. Under the simultaneous events, the existing methods are not applicable for this scenario, for these methods are used to infer the origin of instability which is triggered by only one event. Taking the failure of $l(2 \leftrightarrow 3)$ and $l(2 \leftrightarrow 5)$ in figure 2 for example, the existing methods only consider the stable paths [5] $(9,7,6,3,2,1)$ and $(9,7,6,5,4,1)$, and they could not pinpoint the failed link $l(2 \leftrightarrow 5)$, because the failure of link $l(2 \leftrightarrow 5)$ doesn't contained in the two stable paths. As a result, the existing methods would infer only part or error origins of the instability under simultaneous events. Therefore, we propose a new scheme to pinpoint the origins of instability under simultaneous events by exploring the cycles, and it is compatible to the existing methods, as shown in figure 4.

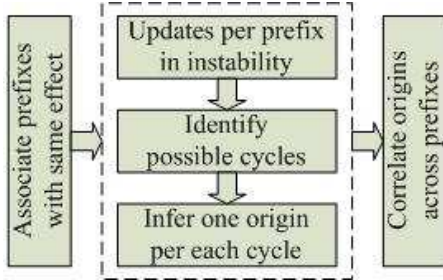


Fig.4. Locating instability under simultaneous events

```

New valid paths  $Vp = \emptyset$ , Last valid path  $r = \pi_s$ 
Foreach AS path  $\pi$  of prefix  $p$ 
  If  $\pi = \emptyset$  ## it is a withdraw
     $Vp = withdraw$ ,  $r = withdraw$ 
  else
     $\pi_p = subpath(\pi, r)$ 
    If  $\pi_p \in Sp$ , ##  $Sp$  is the set of subpaths
      continue;
    Else
       $Sp = \pi_p$ ,  $r = \pi$ ,  $Vp = \pi$ 
Foreach subpath  $\pi_p \in Sp$ 
  Valid subpath  $Pr = \text{examin}(\pi_p)$ 
  For each new valid path  $\pi_p \in Pr$ 
    Paths for cycle  $Pc = \text{mapValidSubpth}(\pi_p, Vp)$ 

```

Fig.5. Identifying cycles

As discussed in section 2.A, it is necessary to explore the new valid paths per prefix. As a matter of fact, there are tens of thousands prefixes in Internet nowadays. It is of a great volume of workload to explore cycles for all prefixes. In fact, the best paths of many prefixes share a common subpath to reach a vantage point. For example, prefix $p1$, $p2$ and $p3$ in figure 2 share the common link $l(3 \leftrightarrow 6)$ to reach the vantage point. So when link $l(3 \leftrightarrow 6)$ failed, it is only necessary to consider the prefix $p3$ which is nearest to the failure among all the prefixes that their best paths share the failure link.

With the technique described in section 2, it is useful to explore the cycles hidden in updates to identify the simultaneous events. The concrete algorithm is specified in figure 5, which shows exploring cycles per prefix in a routing instability under simultaneous events. And π_b is the stable path of prefix p before the instability, function $subpath(\pi, r)$ finds out the subpath of the two paths, $\text{examin}(\pi_p)$ check whether the sub path is available, and $\text{mapValidSubpth}(\pi_p, Vp)$ maps the subpath to the new valid path. As each cycle is corresponding to an event, we could make use of the existing methods to infer the origin of this event, and this make our scheme is compatible.

Through these efforts, a lot of origins are found per prefix. If some of the cycles from different prefixes are intersected, the cycles may be triggered by the same event. So it is necessary to correlate the origins across the prefixes. There are some existing correlation methods, so we could transplant them to correlate the failure events and omit the specification for the restriction of space in this paper.

4. Conclusions

Locating the internet instability is very important to diagnose the internet problem. However, the existing methods are only used to infer the origin of instability which is triggered by one event. As a matter of fact, there are many simultaneous events happen in real world, such as earthquake, large scale black out, and so on. The existing methods are not applicable for the scenario of simultaneous events, for these methods would result in partial or error inference conclusion. This paper is the first study to characterize the simultaneous events, and propose a new scheme to pinpoint the origins by exploring cycles under this situation. Furthermore, this scheme is compatible to the existing methods.

* This work is supported by the National Natural Science Foundation of China under Grant No. 61070199.

References

- [1] Craig Labovitz, G. Robert Malan, Farnam Jahanian, "Internet routing instability," IEEE/ACM Transactions on Networks, Vol.6, No.5, October 1998.
- [2] R. Teixeira and J. Rexford, "A Measurement Framework for Pin-Pointing Routing Changes," In Proceedings of 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," in Proc. IEEE International Conference on Network Protocols (ICNP), November 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," in Proc. ACM SIGCOMM, pages 205–218, Portland, OR, Aug. 2004.
- [6] M. Lad, A. Nanavati, D. Massey, and L. Zhang, "An algorithmic approach to identifying link failures," in Proc. 10th IEEE Pacific Rim International Symposium on Dependable Computing (PRDDC), March 2004, pp. 25-34.
- [7] 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," in Proc. Networked Systems Design and Implementation (NSDI), May 2005.
- [8] M. Lad, R. Oliveira, D. Massey and L. Zhang, "Inferring the Origin of Routing Changes using Link Weights," In Proceedings of IEEE ICNP, Oct. 2007.
- [9] Y. Huang, N. Feamster, A. Lakhina, and J. Xu. "Detecting Network Disruptions with Network-Wide Analysis," in Proc. of ACM SIGMETRICS, 2007.
- [10] 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
- [11] A. Campisano, L. Cittadini, G. Di Battista, T. Refice, C. Sasso, "Tracking Back the Root Cause of a Path Change in Inter-domain Routing," In Proceedings of IEEE/IFIP NOMS, Apr. 2008.
- [12] Wei Liang, Ye Li, Jingping Bi and Guoqiang Zhang. "On the Accurate Identification of Familiar Inter-Domain Routing Instabilities," GLOBECOM 2008.
- [13] Kuate, A.M., Teixeira, R., Feamster, N., and Meulle, M. "Joint analysis of network incidents and intradomain routing changes," CNSM 2010, pp.198-205.
- [14] M.Watari, A.Tachibana, and S.Ano. "Inferring the Origin of Routing Changes based on Preferred Path Changes," PAM 2011.
- [15] R. Oliveira, B. Zhang, D. Pei, L. Zhang, "Quantifying Path Exploration in the Internet," IEEE/ACM Transactions on Networking, vol. 17, no. 2, pp. 445-458, Apr. 2009.