Tracking Back the Root Cause of a Path Change in Interdomain Routing

Alessio Campisano, Luca Cittadini, Giuseppe Di Battista, Tiziana Refice and Claudio Sasso
Dipartimento di Informatica e Automazione, Università di Roma Tre, Rome, Italy.
{gdb,ratm,refice}@dia.uniroma3.it, {alessio.campisano,claudio.sasso}@gmail.com

Abstract—**Interdomain routes change over time, and it is impressive to observe up to which extent. Routes may change many times in the same day and sometimes in the same hour or minute. Such changes are caused by several types of events, e.g., a routing policy variation in an ISP, a router reboot, or a link fault. In this paper we do a step towards the identification of the cause of route changes, a problem that is attracting increasing attention from both researchers and network administrators. Namely, we propose a methodology for analyzing a given BGP route change in order to, at least partially, locate the event that triggered the change. The methodology is supported by a publicly available on-line service.**

I. Introduction

We consider the following scenario. A network administrator of an Internet Service Provider (ISP) observes that one of the prefixes announced by its Autonomous System (AS) to the Internet had a BGP path change at a certain time. For example, prefix p announced by AS1 usually reaches AS4 passing through AS2 and AS3, while suddenly it started using a different path through AS5 and AS6. The network administrator would like to know why that change happened.

In fact, recent works (e.g., [17]) underline the impact of routing changes in end-to-end performance. Also, this issue becomes much more important as services requiring almost constant delay, limited jitter and packet loss, gain popularity. Hence, many ISPs are interested in understanding what happens to their prefixes in the interdomain routing.

Actually, many research works studied BGP routing dynamics in the last few years. Their contributions can be broadly classified as follows. There are black-box approaches, that apply statistical techniques to group BGP updates into sets that are supposed to be triggered by the same underlying event. Ref. [19] uses the Principal Components Analysis, [22] uses statistics-based anomaly detection, and [21] exploits the wavelet transform. Other authors propose white-box approaches. In [3], [5], [9] streams of BGP updates are analyzed, correlating information across time, topology, collectors, and prefixes. Ref. [13] describes an algorithm, that pinpoints the origin of routing changes due to a link failure or a link restoration, assuming shortest path routing. Finally, some authors (see, e.g., [18]) propose to add an infrastructure to the Internet in order to monitor route changes.

Those contributions generally aim at reporting a full set of events that happened in the network in a given time slice. Roughly, updates are first grouped into clusters, and then events are detected by analyzing multiple clusters. In this modus operandi, the correlation between an update and an event can be biased by the a priori generation of the clusters.

Taking into account the scenario described at the beginning of this section, we propose to tackle the problem from a different perspective. We assume the perspective of an ISP, that is not interested in what happens to the network in general but is rather interested in what happens at a certain time to (some of) its own prefixes. Hence, instead of analyzing a bulk of updates for detecting events in the network, we analyze a specific BGP-update trying to locate its originating event.

In this paper we present the following contributions. In Section II, we show that BGP updates have a flow-based behavior, where the term "flow" is used with its graph-theoretic meaning. The collectors of updates are sources of flow and the ASes originating prefixes are sinks. Exploiting this property, we propose a flow-based model of BGP updates. As far as we know, this is the first time that BGP updates are modelled in terms of a flow system. As a side effect, we put in a flow-based perspective the concept of link-rank, defined in [12]. Further, this section introduces the new concept of global-rank. In Section IV, we propose a methodology for analyzing a given BGP route change c in order to, at least partially, identify and locate the event that triggered c. The cornerstones of the methodology are: (i) A data quality analysis for discarding unreliable data, extending the approach of [20]. (ii) A macro-events detection analysis, focused on local and global ranks. (iii) A fine-grained analysis that analyzes flow changes in a relevant part of the network. The methodology is illustrated by several examples from a reference week.

The effectiveness of the methodology is discussed in Section V by means of simulation experiments and real world data analysis. Our data sources are described in Section III.

The methodology described in Section IV requires the analysis of huge amount of data, and hence it would be unfeasible if not supported by some automatic facility. We developed an on-line service [4] that offers many tools to support the methodology. A prototype version is available at `http://nerodavola.dia.uniroma3.it/rca/`

II. A Flow-based Model

The Internet is divided into administrative domains called *Autonomous Systems* (*ASes*). The *Border Gateway Protocol* (*BGP*) [16] is the routing protocol used to exchange reachability information between ASes. Two ASes having routers that exchange routing information using BGP are said to have a *peering* between them. A BGP router stores in its *Routing Information Base* (*RIB*) the *prefixes* it can reach, and for each of them an *AS-path*. An AS-path, also called *route*, is the sequence of ASes used to reach the destination prefix. Routes are propagated by BGP messages called *updates*. BGP is an incremental protocol, once two BGP routers establish a peering, they exchange their whole RIB each other. This process is called *table transfer*. Further updates are sent only if a route changes, in response to network events (e.g., link failure, router reset, or policy change).

To obtain information about the Internet routing dynamics, projects - such as the RIPE NCC's *Routing Information Service* (*RIS*) [2] and the University of Oregon's *RouteViews Project* (*RV*) [1] - spread around the world several passive collection boxes, called *(Remote) Route Collectors* (*RRCs*). Each route collector has peerings with several BGP routers, called *Collector Peers* (*CPs*), belonging to various ASes. The routing tables of all RRCs and the updates they receive are periodically dumped, permanently stored, and made publicly available. Some collector peers provide information about all the prefixes on the Internet, while others only provide information about a subset of them. We call the former *full collector peers*, the latter *partial collector peers*.

Several models have been proposed to study the evolution of interdomain routing. Most of them assume that each AS can be collapsed into a single router, while others [14] represent the internal structure of each AS with different levels of accuracy. The first approach can be too coarse-grained to capture the impact of the internal routing of an AS on the evolution of the Internet. On the other hand, the second approach contrasts with the fact that the currently available methodologies and data are not able to provide a fine-grained complete and accurate description of the internal structure of an AS.

In this section, we introduce a model based on the concept of flow. The model is shown to be valid not depending on the internal structures of any AS. The validity of the model has the benefit of allowing correct deductions in Root Cause Analysis of interdomain routing. Of course, it also has the drawback of not capturing dynamics internal to an AS.

We consider the following sets. $\mathcal{ASes}$ is the set of all the known ASes, $\mathcal{ASes} = \{1, \ldots, 65535\}$. Since we will consider a graph whose nodes are the elements of $\mathcal{ASes}$, the ASes will also be called *vertices*. $\mathcal{T}$ is the set of all the considered instants of time when a BGP update is received by a RRC from a collector peer. $\mathcal{CP}$ is the set of all the collector peer identifiers.

An *AS-path* (or simply *path*) π is a sequence of ASes such that $\pi = (as_n, \ldots, as_0)$ where $as_i \in \mathcal{ASes}$. as_0 is called *origin*. The empty path is an AS-path and is denoted by ϕ. $\mathcal{AP}$ is the set of all known AS-paths. A pair (as_{i+1}, as_i) of ASes that are consecutive in some AS-path is an *edge*. We consider the edges as directed, i.e. $(v, w) \neq (w, v)$. We say that a path *contains* an edge, $\pi \supseteq (as_{i+1}, as_i)$.

An *update* u is a quadruple (cp, p, π, t) where $u.cp \in \mathcal{CP}$ is the CP that collected the update, $u.p \in \mathcal{P}$ is the prefix contained in the update, $u.\pi \in \mathcal{AP}$ is the route announced by the update, and $u.t \in \mathcal{T}$ is the time when the update is collected. If $u.\pi \neq \phi$ then u is an *announcement*, otherwise it is a *withdrawal*. $\mathcal{U}$ is the set of all known updates.

The *last* update u that collector peer cp received for prefix p, before time t, is denoted $\ell_{cp}(p,t)$; formally, $\ell_{cp}(p,t)$ is such that $\ell_{cp}(p,t).t < t$ and $\nexists u \in \mathcal{U} \;\mid u.cp = cp \;\wedge\; u.p = p \;\wedge\; \ell_{cp}(p,t).t < u.t < t$.

An update u causes a routing *change*. A change c is a quintuple $(u.cp, u.p, \pi_{old}, \pi_{new}, u.t)$ where $\pi_{old} = \ell_{cp}(p,t).\pi$ and $\pi_{new} = u.\pi$.

We now define three concepts that will be crucial for the methodology described in Section IV, called *local rank*, *global rank*, and *origin rank*. While the first has been introduced in [12], the others are, as far as we know, unexplored concepts. Given a collector peer *cp*, the *local rank* of an edge e at time t is defined as the number of prefixes whose path at time t, as observed by *cp*, contains e. Namely, $lrank(cp, e, t) = |P_{cp}(e,t)|$, where $P_{cp}(e,t) = \{p \in \mathcal{P} \mid e \subseteq \ell_{cp}(p,t).\pi \vee \exists u = (cp, p, \pi, t) \in \mathcal{U} \mid u.cp = cp, u.t = t, e \subseteq u.\pi\}$. We define the *global rank* of an edge e at time t as

$$grank(e,t) = |P(e,t)|, P(e,t) = \bigcup_{cp \in \mathcal{CP}} P_{cp}(e,t).$$

Fig. 1 illustrates the values of local and global ranks of the edges of a fragment of Internet, at a certain time t. For example, the label $(1,2,2,3)$ on edge $(as1, as2)$ states that $lrank(cp_1, (as1, as2), t) = 1$ since cp_1 sees just the green dashed prefix traversing $(as1, as2)$. Also, $grank((as1, as2), t) = 3$ since $(as1, as2)$ is traversed by all three prefixes. Note that $grank((as1, as2), t) \neq \sum_{i=1,2,3} lrank(cp_i, (as1, as2), t)$. Observe that, even if cp_2 and as_2 have multiple peerings, according to our definitions, we labelled their pair only once.

Intuitively, while the local rank measures the number of prefixes that are observed passing through an edge by a single *cp*, the global rank measures the number of distinct prefixes that are observed passing through an edge by any $cp \in \mathcal{CP}$.

Finally, given a collector peer *cp* we define the *origin rank* of AS v at time t as $\theta(cp, v, t) = |P(cp, v, t)|$, where $P(cp, v, t) = \{p \in \mathcal{P} \mid \ell_{cp}(p,t).\pi = (as_n, ..., v), n \geq 0 \;\vee\; \exists u \in \mathcal{U} \;\mid\; u.cp = cp, u.t = t, u.\pi = (as_n, ..., v)\}$. Notice that function $\theta(cp, v, t)$ represents the number of prefixes that, at time t, are known by *cp* as originated by AS v. As an example, consider collector peer cp_1 in Fig. 1. $\theta(cp_1, as_0, t) = 3$ and $\theta(cp_1, as, t) = 0 \; \forall as \neq as_0$.

We denote by $\overline{lrank}$ ($\overline{grank}$) the weighted average of the local (global) rank of an edge over time.

Whenever a negative (positive) variation of a local rank is observed during a given time interval, it is interesting to further investigate where prefixes "went to" ("came from"). Intuitively, prefixes move around on the AS graph, as well as water would move in a pipe network. This analogy introduces the concept of flows of prefixes. Tracking the flow of prefixes along different paths can be done by adapting the well-known concept of flow system to the interdomain routing.

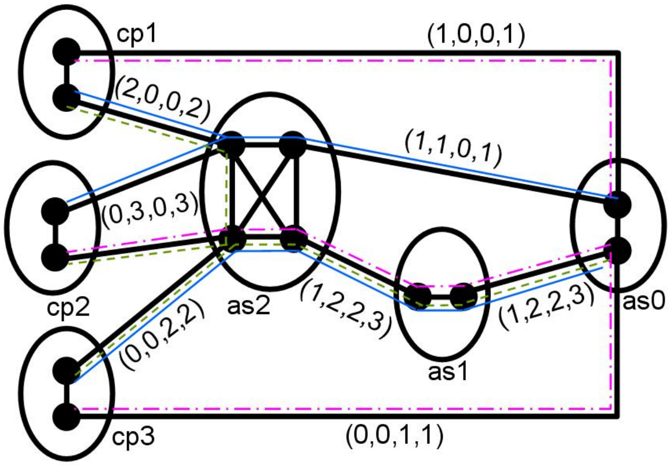

Figure 1. Big points represent routers, thick solid black lines represent IBGP or EBGP peerings between routers, and ellipses represent ASes. ASes cp_i, $i \in \{1,2,3\}$, contain collector peers. Each edge e inter-AS is labelled with a quadruple containing $lrank(cp_1, e, t)$, $lrank(cp_2, e, t)$, $lrank(cp_3, e, t)$, and $grank(e, t)$. AS as_0 originates three prefixes. Thin blue solid, green dashed, and pink mixed lines represent the routes to such prefixes observed by collector peers.

Given a directed graph $G = (V, E)$, a specific vertex as_n called *source*, and a mapping between vertices and flow absorption $g : V \rightarrow \mathbb{Z}$, then a *flow system* is a function $f : E \rightarrow \mathbb{Z}$ where $\forall v \in V, v \neq as_n$,

$$\sum_{(u,v)\in E} f(u,v) - \sum_{(v,w)\in E} f(v,w) = g(v).$$

Theorem 1 shows that functions $lrank(cp, e, t)$ and $\theta(cp, v, t)$ define a flow system at time t. Intuitively, we have that the source of the flow is the AS as_n in which cp is located, and the sinks are all the ASes that originate some prefixes, as observed by cp. A prefix contributing to one unit of the local rank of some edge e contributes also to one unit of the amount of flow traversing e. As an example, consider collector peer $cp_1 = as_n$ of Fig. 1. For instance, for as_2, we have that the sum of the local ranks over incoming edges $(cp_1, as2)$ is 2, and the sum of the local ranks over outgoing edges (as_2, as_0) and (as_2, as_1) is 2. On the other hand, since as_2 does not originate any prefix known to cp_1, $\theta(cp_1, as_2, t) = 0$. Hence, the flow around as_2 is conserved.

Theorem 1. *At a specific time instant, functions* $lrank$ *and* θ *define a flow system.*

Proof: Select a specific instant t and a specific collector peer cp. Consider the value $x = \theta(cp, v, t)$ of function θ for any vertex v. Because of the definition of θ we have that for each unit of flow in x there exists a prefix p such that either $p \in \mathcal{P} \mid \ell_{cp}(p,t).\pi = (as_n, ..., v), n \geq 0$ or $\exists u \in \mathcal{U} \mid u.cp = cp, u.t = t, u.\pi = (as_n, ..., v)$. In both cases we identify an update u that is received from as_n and originates from v. Consider a sequence of two consecutive edges (as_{i+1}, as_i) and (as_i, as_{i-1}) contained in $u.\pi$, u contributes with one unit of flow both to $lrank(cp, (as_{i+1}, as_i), t)$ and to $lrank(cp, (as_i, as_{i-1}), t)$. Hence, for each AS $as_i \neq v$, u does not affect the balance of as_i. This means that for each vertex v, if we consider only paths not ending with v we have

$$\sum_{w \in V} lrank(cp, (w,v), t) = \sum_{w \in V} lrank(cp, (v,w), t).$$

Now, each path ending with v increases both the flow on an incoming edge, $lrank(cp, (w,v), t)$, and $\theta(cp, v, t)$. Then we conclude that, $\forall v \in V$,

$$\sum_{w \in V} lrank(cp, (w,v), t) = \sum_{w \in V} lrank(cp, (v,w), t) + \theta(cp, v, t).$$

■

Observe that Theorem 1 holds even if some ASes perform BGP prefix aggregation. In fact, in this case a collector peer is unable to track all the prefixes contained in the aggregation and the aggregated prefix counts for just one unit of flow.

We stress that functions $grank$ and θ do not define a flow system. As a counterexample, consider again AS as_2 in Fig. 1. We have that the algebraic sum of the global ranks of the edges incident on as_2 is not zero.

Theorem 1 is useful to depict a snapshot of the network at a given instant, while in Theorem 2 we relate the flows of two different instants of time. We define the functions

$$\Delta lrank_t^{t+\tau}(cp, e) = lrank(cp, e, t+\tau) - lrank(cp, e, t)$$

that captures local rank variations (flow variations) between t and $t+\tau$, and the function

$$\Delta\theta_t^{t+\tau}(cp, v) = \theta(cp, v, t+\tau) - \theta(cp, v, t).$$

that accounts for the variation in the number of prefixes that are known by cp as originated by v.

Theorem 2. *Functions* $\Delta lrank_t^{t+\tau}(cp, (v,w))$ *and* $\Delta\theta_t^{t+\tau}(cp, v)$ *define a flow system.*

Proof: For the sake of simplicity, we use $l((v,w), t)$ in substitution of $lrank(cp, (v,w), t)$. $\forall v \in V$:

$$\sum_{w \in V} \Delta lrank_t^{t+\tau}(cp, (w,v)) - \sum_{w \in V} \Delta lrank_t^{t+\tau}(cp, (v,w)) =$$
$$\sum_{w \in V} l((w,v), t+\tau) - \sum_{w \in V} l((v,w), t+\tau) +$$
$$-\sum_{w \in V} l((w,v), t) + \sum_{w \in V} l((v,w), t) =$$
$$\theta(cp, v, t+\tau) - \theta(cp, v, t) = \Delta\theta_t^{t+\tau}(cp, v).$$

■

Observe that, because of the high connectivity of the Internet, a collector peer is likely to be able to reach a constant number of prefixes over time. Also, each of such prefixes is typically announced always by the same origin. Hence, we expect that function $\Delta\theta_t^{t+\tau}(cp, v)$ is zero in most cases. That is, we expect that the flow is overall conserved over time.

III. Data Set

Our work relies on BGP data obtained from RIS [2] and RV [1]. Throughout this paper we use the data collected from 12/26/2006 to 01/02/2007, and we call this time interval *reference week*. We chose this week because it featured massive BGP activity due to Taiwan earthquakes and it preceded the fix of a RIS collectors' bug. During the reference week, there were 526 collector peers, 30% of them were full cps.

Our dataset contains 320,678,893 updates (nearly 46M updates per day on average) with 7,537,378 distinct AS-paths on 70,078 distinct peerings and 24,493 distinct ASes. The number of observed prefixes is 235,725.

Route Collectors Reliability Screening

To check the reliability of route collectors, we periodically perform on all RRCs a preprocessing step, called *RRC Reliability Screening*. The screening of a route collector rrc' over a time interval $[t_{start}, t_{end}]$ is executed as follows: (i) we make a local copy of the RIB of rrc' at t_{start}, (ii) we modify the copy according to the updates collected by rrc' during $[t_{start}, t_{end}]$, (iii) we compare the modified copy to the RIB dumped by rrc' at t_{end}, (iv) we decide if rrc' is reliable by evaluating the ratio between number of mismatches and average size of the RIB.

A route collector can be unreliable because of bugs in routing or collection software (see [10] for details), major asynchronies between route collector and collector peer, non-standard behavior of the collector peer (e.g., some highly recommended timers are not implemented).

Reliability Screenings performed during several experiments led to the detection of a major problem that affected RIS route collectors since May 2005. Overall, the problem affected 44 collector peers, 12 of them were full collector peers. Contacting the RIS maintainers resulted in that problem being fixed by Jan. 2nd, 2007. Since the reference week is before the fix, we are able to assess the impact of the screening step on our dataset.

IV. Analyzing a Route Change

We present a methodology for analyzing a given route change c within the model of Section II. The goal is to identify the portion of the Internet where the event that caused c happened. The methodology consists of three steps. *Collector Peer Check and Selection*: We check the availability of collector peers, and we select a set of collector peers that will be considered in the following analysis. *Macro-Events Detection*: We look for patterns of macro-events, by exploiting the global and local ranks of some edges. This step relies on Theorems 1 and 2. *Fine-Grained Analysis*: If no macro-event has been detected in the previous step, we perform a fine-grained analysis based on several patterns that are consequences of Theorem 2.

Before starting the description of the steps, we underline an issue related to the timing of network events. In several points of the methodology, we analyze what happens in a time interval including the time $c.t$ of the input route change. According to [11], we consider the time interval $[c.t - \Delta, c.t + \Delta]$, with Δ =180 seconds, as a reasonable compromise between accuracy and feasibility and we refer to it as $T_{c.t}$. However, the methodology does not depend on this choice.

A. Collector Peer Check and Selection

Before starting the analysis of the route change c, the methodology requires to execute the RRC Reliability Screening (Section III) in order to discard all the unreliable RRCs.

However, even reliable collector peers sometimes have reboots. If the collector peer $c.cp$ that receives c has a reboot in $T_{c.t}$, we interrupt the analysis because the data collected through $c.cp$ may be too noisy. Moreover, in the analysis of c, we will rely not only on $c.cp$, but also on other collector peers. Hence, in this step we look for all the collector peers that had a reboot in $T_{c.t}$. Information extracted from those collector peers is not further considered. We detect a reboot by either analyzing BGP session state messages, when available, or by seeking for table transfers using the algorithm described in [4].

Among all reliable collector peers without any reboot in $T_{c.t}$, we select those that belong to the ASes of the paths $c.\pi_{old}$ and $c.\pi_{new}$, because they are the most relevant for the subsequent analysis since they provide the closest perspective to analyze c.

B. Macro-Events Detection

In this step we try to relate c to a macro-event by performing first a global rank analysis and then a local rank analysis. We regard as *macro-events* those which affect either the physical or the logical network topology. e.g. an interdomain link fault/restoration, a BGP router fault/restoration, or a BGP session shutdown/setup.

The evolution of the global rank $grank(e, t')$ with $t' \in T_{c.t}$ is considered for each edge e in $c.\pi_{old}$ and $c.\pi_{new}$. Namely, we check if some edge e in $c.\pi_{old}$ or in $c.\pi_{new}$ has a relevant global rank variation and has a value near to zero in $T_{c.t}$. This occurs when no collector peers see any prefix passing through e, and it is a reasonable evidence that e is involved in some way in the event that caused c. We identified three patterns of global rank evolution: (p1) a sudden loss of all prefixes, (p2) a sudden gain of new prefixes starting from 0 prefixes, or (p3) a sudden loss (gain) followed by the resume of the previous situation. Each patter possibly refers to different types of macro-events. E.g. (p1) describes an interdomain link e that fails and loses connectivity to all the prefixes. Once fixed, prefixes might be routed through e again (p2). According to our experience, both gains and losses usually occur within short time periods, due to BGP convergence time [11]. We relate macro-events to the evolution of the global rank of an edge e because it provides a global perspective given by the simultaneous views of e from several collector peers. As the number of collector peers that can see e decreases, this global perspective is more biased. In order to cope with this behaviour, we define the *rank diversity*. The rank diversity of e is a pair $\langle n, \sigma_x/\overline{x}\rangle$, where n is the number of collector peers cp having $\overline{lrank}(cp, e) > 0$, σ_x and $\overline{x}$ are the standard deviation

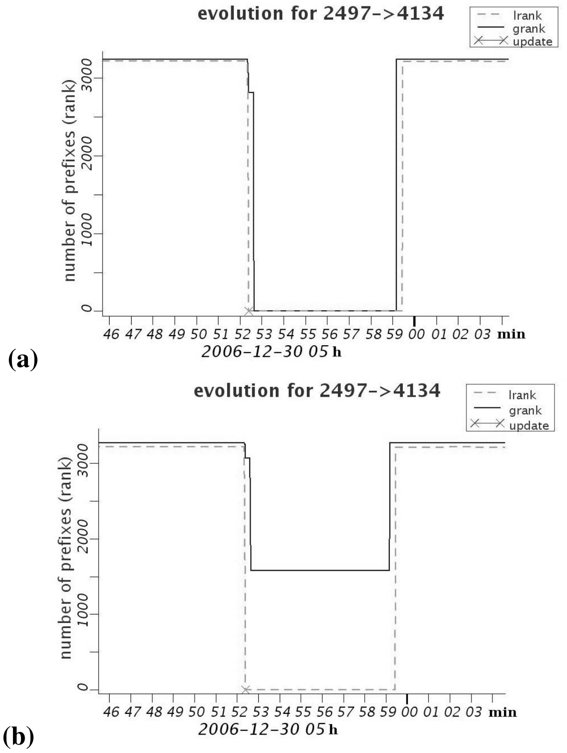

Figure 2. Functions $grank$ (solid black) and $lrank$ (dashed gray) of $(2497, 4134)$. (a) With RRC Reliability Screening. (b) Without RRC Reliability Screening.

and the average, respectively, of such $\overline{lrank}(cp, e)$. We say that the rank diversity is *high* if n is large and $\sigma_x/\overline{x}$ is small. In fact, if n is large we have many collector peers that can see e, and when $\sigma_x/\overline{x}$ is small we have that the collector peers see a similar number of prefixes through e. The global rank analysis provides more valuable information on edges having higher rank diversity.

As an example, we analyze the path change c affecting prefix $c.p$ = 202.41.242.0/24, with $c.\pi_{old}$ = (2497, 4134, 4847, 37942) and $c.\pi_{new}$ = (2497, 2914, 4134, 4847, 37942), observed by $c.cp$ = 198.32.176.24 at time $c.t$ = UTC 30/12/06 05:52:24. First, we check collector peers availability. We identify $\sim$ 20 collector peer resets in $T_{c.t}$ and discard data coming from them. Then, we evaluate the global rank of the edges in $c.\pi_{old}$ and $c.\pi_{new}$. We have that $grank(e) = 0$ in $T_{c.t}$, with $e = (2497, 4134)$. Since $\overline{grank}(e) = 3,166$, edge e has a significant rank variation (Fig. 2.a). The rank diversity of e is $\langle 7, 8.2\% \rangle$. Hence, there are many (7) collector peers that can see e, with a similar number of prefixes. So we consider its global rank trustworthy. This analysis suggests with reasonable confidence that the path change c has been triggered by a macro-event on edge e.

This example also shows the importance of the RRC Reliability Screening. In fact, performing the same analysis skipping such a step, we obtain the evolution of $grank(e)$ shown in Fig. 2.b. In this case, because of the noise generated by the unreliable RRCs, $grank(e)$ never decreases below $1,580$, making the macro-event less visible.

Theorem 1 suggests that whenever there are multiple edges with $grank = 0$ in either $c.\pi_{old}$ or $c.\pi_{new}$ the most likely responsible for the macro event is the edge closest to $c.cp$.

If the global rank analysis ends up with no candidates, we analyse each selected collector peer separately, by looking at the evolution of the local rank in $T_{c.t}$. On edges in $c.\pi_{old}$ and $c.\pi_{new}$, we search for the same patterns as above.

Generally, we trust $grank(as_1, as_2)$ more than $lrank(cp, (as_1, as_2))$, unless cp belongs to as_1 and provides its full routing table. In fact, we consider a collector peer an authoritative source of information on the AS it belongs to. Otherwise, any inference supported only by local rank analysis requires further investigation.

As an example, we analyze the path change c, where $c.p$ = 80.124.192.0/19, π_{old} = (7575, 15557, 8228), π_{new} = (7575, 2914, 3356, 15557, 8228), $c.cp$ = 198.32.176.177, and $c.t$ = UTC 01/01/07 00:04:53.

According to the Collector Peer Check, all the collector peers are available in $T_{c.t}$. We evaluate the global rank of all edges belonging to $c.\pi_{old}$ and $c.\pi_{new}$, and we have that $grank(e) = 0$ in $T_{c.t}$, where $e = (7575, 15557)$. Note that $\overline{grank}(e) = 148$. Unlike the previous example, the rank diversity of e is low ($\langle 2, 0.1\% \rangle$), as the edge is seen by only two collector peers, both belonging to 7575. So its global rank is not worthy. As a consequence, we analyze $lrank$ for $c.cp$. Being $c.cp$ in the left node of e, it is in the best position to observe routing events affecting e. Fig. 3 illustrates the evolution of $lrank(c.cp, e)$, and $grank(e)$ for $e = (7575, 15557)$ and $e' = (3356, 15557)$. It is interesting to notice that a relevant number of prefixes moves from an edge to the other (Theorem 2). From the information extracted, we can deduce with reasonable confidence that the path change c has been triggered by some macro-event affecting edge e.

C. Fine-Grained Analysis

If the Macro-Event Analysis doesn't identify any cause for the route change c, we examine flow changes in order to capture routing events which don't affect the interdomain topology. Namely, we look for events (e.g., BGP policy changes) that in general do not impact all the prefixes passing through an interdomain link, but only a subset of them.

In the Fine-Grained Analysis we investigate flow changes on the whole Internet. However, in our experience, a flow change can spread over a very large portion of the Internet, making the analysis unfeasible. Thus, we focus on a fraction of a flow change, introducing the concepts of path compatibility and restricted flow.

Two paths π_1 and π_2 are *compatible* ($\pi_1 \bowtie \pi_2$) when they share a common left subsequence of at least two ASes (i.e. they share the first edge). A *restricted flow* $\Delta_t^{t+\tau} \hat{f}_P(cp, (u, w))$ is a flow defined on a subset $P \subseteq \mathcal{P}$ of the prefixes. We consider especially interesting the restricted flow on prefixes that experienced, in $T_{c.t} = [t, t+\tau]$, a change whose either the old or the new path is *compatible* with a given path π. In fact, such a restricted flow can be used to study routes coming from (moving to) π. Formally, we evaluate $\Delta_t^{t+\tau} \hat{f}_P(cp, (u, w))$,

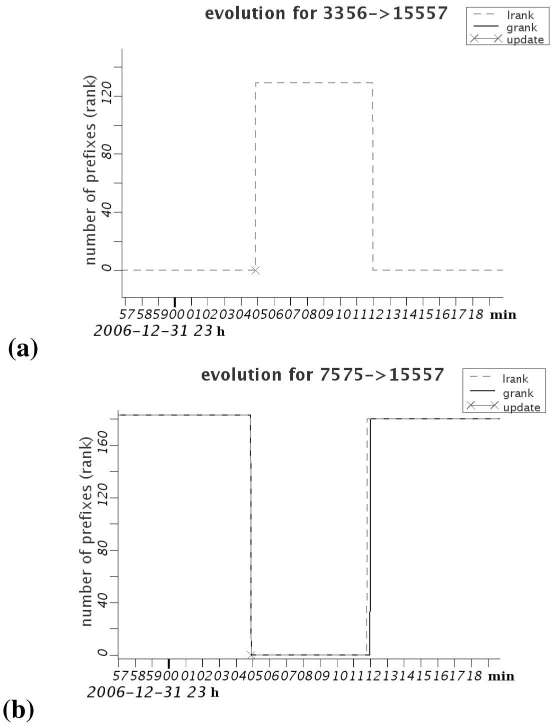

Figure 3. Functions $grank$ (solid black) and $lrank$ (dashed gray) of $(3356, 15557)$ (a), and $(7575, 15557)$ (b).

with $P = \{p \mid \ell_{cp}(p,t).\pi \neq \ell_{cp}(p,t+\tau).\pi \wedge (\ell_{cp}(p,t+\tau).\pi \bowtie \pi \vee \ell_{cp}(p,t).\pi \bowtie \pi)\}$.

For example, we analyze the path change c, where $c.p =$ 202.59.174.0/24, $c.\pi_{old} =$ (16215, 3549, 5511, 4761, 17727), $c.\pi_{new}$ = (16215, 3549, 3320, 4761, 17727), $c.cp$ = 80.81.192.143, and $c.t$ = UTC 12/27/06 10:06:17. After an unsuccessful Macro-Events Detection, we proceed with the present step.

We try to track the rearrangement of the prefixes routed away from $c.\pi_{old}$ (onto $c.\pi_{new}$). Thus, we compute the previously defined flow $\Delta_t^{t+\tau}\hat{f}_P$ where $\pi = c.\pi_{old}$ ($c.\pi_{new}$). In our example we have that prefix 202.57.0.0/24 has, in $T_{c.t}$, a path change from $(16215, 3549, 5511, 4761, 17658)$ to $(16215, 3549, 7473, 4761, 17658)$. The old path is compatible with $c.\pi_{old}$ ($(16215, 3549, 5511, 4761, 17658) \bowtie c.\pi_{old}$). Hence, it is part of the set P (also containing 740 other prefixes) that we use to compute the restricted flow. Observe that, in any restricted flow, edges with a positive flow value describe where prefixes leaving paths compatible with π are re-routed to. Thus, we focus on these edges to analyze prefixes that left $c.\pi_{old}$ ($\pi = c.\pi_{old}$). On the other hand, negative flow values indicate where prefixes moving on paths compatible with π come from. Therefore, we focus on these ones to study prefixes that move onto $c.\pi_{new}$ ($\pi = c.\pi_{new}$).

We build a *restricted flow graph* consisting of edges having $\Delta_t^{t+\tau}\hat{f}_P > 0$ ($\Delta_t^{t+\tau}\hat{f}_P < 0$). Fig. 4 outlines a sketch of the restricted flow graph computed on $c.\pi_{old}$ from our example. The graph visualizes how prefixes in P moved from edges in $c.\pi_{old}$ (red-colored, within the box) to the other edges (green-colored, outside the box). For the sake of clarity, Fig. 4 omits edges with negligible flow values. Notice that most prefixes move away from the first two edges of $c.\pi_{old}$. This is mainly a consequence of flow systems behavior: the flow is more likely to be high on edges closer to the source (collector peer).

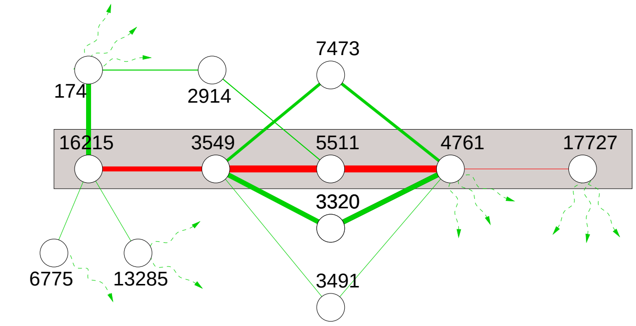

Figure 4. Green edges and their end-vertices are a portion of the restricted flow graph. Path $\pi = c.\pi_{old}$ is also displayed (highlighted in the box) for convenience. Thicker lines represent edges with higher value of $\Delta_t^{t+\tau}\hat{f}_P$

Observe that a lot of prefixes move from $(3549, 5511)$ to edges $(3549, 3320)$, $(3549, 3491)$, and $(3549, 7473)$. Also, those prefixes are still routed through AS 4761 (edges $(3320, 4761)$, $(7473, 4761)$, and $(3491, 4761)$). We argue that this happens due to some event on $(3549, 5511, 4761)$, since multiple events on $(3549, 3320, 4761)$, $(3549, 3491, 4761)$, and $(3549, 7473, 4761)$ are much less likely to occur concurrently. However, we cannot further distinguish if c happens because of a routing event on either $(3549, 5511)$ or $(5511, 4761)$. We generalize the above discussion by considering the nodes m_o and m_i in π having, respectively, maximum outgoing flow and maximum incoming flow in the restricted flow graph. The output set of candidates is the subpath $(m_o, \ldots, m_i)$ of π.

There are some border-line cases to consider. For example, in case two vertices have maximum outgoing (incoming) flow, we can break the tie considering the largest possible candidate set. As another example, there can be many vertices that have similar values of outgoing (incoming) flow. In this case our approach allows to deepen the analysis picking one of the path changes that involve maximum flow vertices and applying the same methodology iteratively on that change. This shift of focus makes our methodology inherently iterative, and allows to cope with the “induced instabilities” problem [9], overcoming a common limitation of inference systems, which are usually able to locate causes of a route change only on the new or the old path.

In order to automatically compute the metrics our methodology relies on, we developed a prototype service. Given a 15-minutes update chunk, the current implementation computes $lranks$, $granks$, and running averages of all the interdomain links in about 1 minute on an entry-level server. Thus, network operators could benefit from a near real-time macro-events detection. The analysis can be selectively refined by applying the fine-grained step to specific path changes, in about 15 seconds per path change.

V. Evaluation

Validating the effectiveness of methodologies for root-cause analysis is a well known hard task both using Internet real data and working on simulations. In the first case, it is difficult to rely on a complete and meaningful set of faults, since producing worldwide outages is of course unfeasible and there is no publicly available history of past faults. In the second case, reconstructing a realistic scenario with the existing platforms is a challenging task since most of them approximate some dimensions (e.g., topology, policies, routing dynamics) of the problem in order to spare resources. We approached the problem from the simulation perspective, supporting it with the analysis of real Internet data to tackle simulation limitations.

We performed extensive simulations, using the state-of-the-art C-BGP platform [15]. Namely, we settled a network with the Internet topology from [8], inferred using RV data [1] collected on August 2007. We set up BGP policies according to the customer-provider relationships provided therein and we successfully checked the validity of the policies with the methodology in [7]. The resulting AS graph consisted of 25,599 ASes and 52,135 interdomain links. In order to account for the impact of collectors' location, we placed collector peers in the same ASes as the full collector peers of RV.

C-BGP handles efficiently only a small number of prefixes on such a huge topology, hence we were able to only deal with about 400 prefixes at the same time. Each prefix was originated by a different AS. To reduce the bias due to the location of the originating ASes, we randomly selected the set of originators in 12 distinct and independent experiments.

In each experiment, we separately generated 36 routing events and gathered all the updates collected by our collector peers. We applied our methodology to this dataset and then we compared the output candidate set with the actual root cause of each event. We simulated 3 different types of events: (i) interdomain link failures/restorations, (ii) routing policy (local-pref) changes, enforced by an hard-reset, and (iii) routing policy (local-pref) changes, enforced by soft-reset [6]. Routing policy changes were configured such that they only affected a subset of the prefixes. We further classified events according to their location in the Internet hierarchy (tier1/transit/stub ASes), choosing 3 distinct affected edges for each class.

Table I summarizes our results. Each entry of the table shows the accuracy of our approach for the given event type. Percentages represent the ratio between the number of input updates for which the methodology correctly returned a candidate set containing the root cause, and the total number of updates triggered by the event. The ratios were averaged over the 3 edges and the 12 distinct choices of the originators. Fig. 5 shows the Cumulative Distribution Function (CDF) of the size of the returned candidate sets. Most sets had 1-2 elements.

Table I shows that the Macro-Events Detection step is quite effective in explaining updates generated by topology changes. The results are also quite satisfactory for the Fine-Grained Analysis. Notice that policy changes requiring hard reset have been approximately seen as topology changes and hence detected by the Macro-Events step. Results show that it is more difficult to detect events which do not alter the network topology (e.g., LP soft), because they can affect only a subset of all the prefixes on a link. As shown by [9], events involving the T1 network are usually harder to locate, due to the huge redundancy within the Internet core.

Table I
Percentage of updates correctly related to an event. LP = local preference change, T1 = tier1 AS, t = transit AS, s = stub AS

Event type	**# of updates**	**Macro**	**Fine**	**Missed**
link down-up, T1-T1	10303	100%	0%	0%
link down-up, T1-t	8898	100%	0%	0%
link down-up, t-t	7358	100%	0%	0%
link down-up, t-s	12855	100%	0%	0%
LP hard, T1-T1	6998	71.28%	27.82%	0.9%
LP hard, T1-t	5882	99.9%	0%	0.1%
LP hard, t-t	4759	100%	0%	0%
LP hard, t-s	8732	99.9%	0%	0.1%
LP soft, T1-T1	8542	0%	92.58%	7.42%
LP soft, T1-t	1340	0%	98.5%	1.5%
LP soft, t-t	1704	0%	100%	0%
LP soft, t-s	1056	0%	95.17%	4.83%

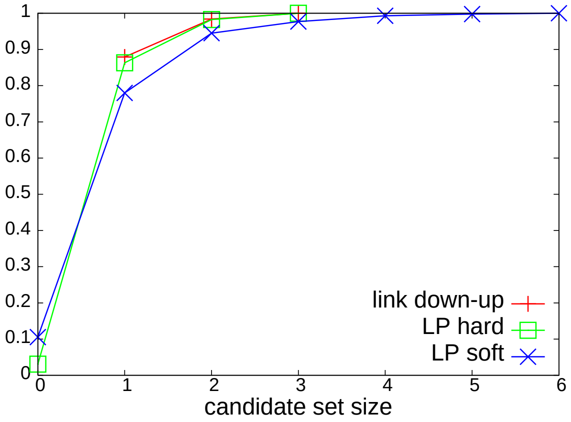

Figure 5. CDF of the size of the candidate sets.

The confidence in our experimental validation is supported by [15], which provides some evidence that C-BGP is a good simulator of real Internet. However, the results are affected by the following limitations: (i) Collector peers were all reliable. This does not allow to understand the impact of the RRC Reliability Screening and of the Collector Peer Check. (ii) C-BGP is optimized to reduce path-exploration updates. This decreases the relevance of the link down-up experiments. In fact, in the real world a negative event (e.g., a link-down) produces several path-exploration updates.

In order to better understand the impact of limitation (i), we analyzed real-world data of the reference week. First, we looked for all collector peers affected by reboots, performing the Collector Peer Check (Section IV). We used only route collectors that successfully passed the RRC Reliability Screening (Section III). Taking into account only full collector peers, we identified 90 table transfers, each one corresponding to a session reset. BGP session state messages, only available for RIS collectors, reported 71 session resets. Overall, the average percentage of time affected by session resets was 0.01% of the reference week per each collector peer. Note that the average increased to 3% if we considered both full and partial collector peers. These results show that the Collector Peer Check step discarded a non negligible portion of the input data.

Afterwards we applied the Macro-Events Detection step on

Table II
NUMBER OF EDGES AFFECTED BY A GRANK0 EVENT.

	reliable cps	**all cps**
all edges	8528	6332 (-25.75%)
edges with $\overline{grank}(e) > 50$	455	372 (-18.24%)
edges with $\langle n \geq 3, \sigma_x/\overline{x} \leq 30\% \rangle$	16	13 (-18.75%)
edges with $\overline{grank}(e) > 50$ **&** $\langle n \geq 3, \sigma_x/\overline{x} \leq 30\% \rangle$	9	7 (-22.22%)

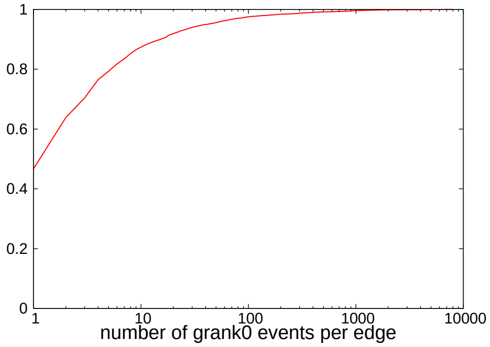

Figure 6. CDF of the number of grank0 events that each edge experienced.

the filtered data. Table II shows the number of distinct edges e affected by at least one *grank0 event*, i.e. with $\overline{grank(e)} > 0$ and $grank(e,t) = 0$ for at least one t in the reference week. We detected 8,528 edges, revealing that grank0 events have been significantly frequent. Among those edges, Fig. 6 shows the CDF of the number of grank0 events experienced by each edge. Out of the detected edges, 455 were relevant edges, i.e. edges with high grank. Only a few edges had a global visibility, i.e. had a significant rank diversity. The second column of Table II points out that some grank0 events were not detectable without eliminating unreliable collector peers.

Concerning limitation (ii), we notice that path-exploration updates are perhaps the less interesting and are quite easy to spot, since they usually can be grouped into sequences of transient path changes that have a very short duration.

Comparison with previous work is not trivial since most of the approaches (e.g., [3], [19]) try to locate events, with little or no interest in the association between an event and the updates it triggered. The simulation in [19] does not realistically represent the Internet (400 ASes, one vantage point, and shortest path routing), so results are mostly uncomparable. Ref. [9] simulates link down-up events, using a setting similar to ours, and obtains 89% accuracy, while we achieve 100% for the same event type.

VI. CONCLUSIONS AND FUTURE WORK

Despite the large amount of efforts, finding the causes of specific interdomain routing changes is extremely challenging. On one hand, nowadays both researchers and network administrators can benefit from large BGP datasets, provided by several BGP collectors spread worldwide. On the other hand, existing approaches exploiting such data strive to identify all network events, disregarding specific changes.

This paper gives three fundamental contributions to face the above problem. First, we show a new, clean model for describing BGP updates based on a flow system. Second, relying on the model, we present a methodology that tackles the root cause analysis problem from a new perspective. We consider the point of view of an ISP that experienced a change affecting (some of) its prefixes, and would like to pinpoint its cause. Namely, instead of searching for all network events, we focus on a single BGP change, combining coarse and fine grained information, in order to track back the event that generated it. We evaluated our approach through Internet scale simulations, and real world data analysis. Results show that this new perspective, in many cases, helps locate the cause of the input change among a reasonably small set of candidates. Third, our methodology is supported by an on-line service.

As future work, we plan to extend the methodology, enlarging the set of patterns we are able to recognize, in order to explain a higher number of changes. We shall also improve the service, to fully support the methodology. Recent work (see, e.g. [14]) analyze public BGP data trying to obtain a partial insight on the internal structure of the ASes. Can this additional information leverage the accuracy of our approach? Also, our flow model is valid for each single collector peer. Are there cases or assumptions when it is valid for multiple collector peers? An answer to these questions could improve our understanding of Internet routing dynamics.

REFERENCES

[1] Oregon Route Views Project. `http://www.routeviews.org/`.
[2] Routing Information System. `http://www.ripe.net/ris/`.
[3] M. Caesar, L. Subramanian, and R. H. Katz. Towards Localizing Root Causes of BGP Dynamics. Technical Report UCB/CSD-04-1302, 2003.
[4] A. Campisano, L. Cittadini, G. Di Battista, T. Refice, and C. Sasso. Update-Driven Root Cause Analysis in Interdomain Routing. Technical Report RT-DIA-117-2007, 2007.
[5] D. Chang, R. Govindan, and J. Heidemann. The Temporal and Topological Characteristics of BGP Path Changes. In *ICNP*, 2003.
[6] E. Chen. Route refresh capability for bgp-4. IETF RFC 2918, 2000.
[7] G. Di Battista, et al. Computing the Types of the Relationships between Autonomous Systems. *IEEE/ACM Transactions on Networking*, 2007.
[8] X. Dimitropoulos, et al. As relationships: Inference and validation. *ACM SIGCOMM CCR*, 37:2007, 2006.
[9] A. Feldmann, et al. Locating Internet Routing Instabilities. In *ACM SIGCOMM*, 2004.
[10] H. Kong. The Consistency Verification of Zebra BGP Data Collection. Technical report, 2003.
[11] C. Labovitz, et al. The impact of Internet policy and topology on delayed routing convergence. In *IEEE INFOCOM*, 2001.
[12] M. Lad, D. Massey, and L. Zhang. Link-Rank: A Graphical Tool for Capturing BGP Routing Dynamics. In *IEEE/IPIF NOMS*, 2004.
[13] M. Lad, A. Nanavati, and L. Z. Dan Massey. An Algorithmic Approach to Identifying Link Failures. In *PRDC*, 2004.
[14] W. Muhlbauer, et al. Building an AS-Topology Model that Captures Route Diversity. In *ACM SIGCOMM*, 2006.
[15] B. Quoitin and S. Uhlig. Modeling the routing of an autonomous system with C-BGP. In *IEEE Network Magazine*, 2005.
[16] Y. Rekhter, et al. A Border Gateway Protocol 4. IETF RFC 4271, 2006.
[17] F. Wang, et al. A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. *ACM SIGCOMM CCR*, 2006.
[18] J. Wuet, et al. Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network. In *NSDI*, 2005.
[19] K. Xu, et al. A First Step to Understand Inter Domain Routing Dynamics. In *ACM SIGCOMM MineNet Workshop*, 2005.
[20] B. Zhang, et al. Identifying BGP Routing Table Transfers. In *ACM SIGCOMM MineNet Workshop*, 2005.
[21] J. Zhang, et al. Learning-Based Anomaly Detection in BGP Updates. In *ACM SIGCOMM MineNet Workshop*, 2005.
[22] K. Zhang, et al. On Detection of Anomalous Routing Dynamics in BGP. In *Networking*, 2004.