# Probing Strategies for Efficient Remapping of Internet Routing Events

Elverton Fazzion, Ítalo Cunha
Universidade Federal de Minas Gerais (UFMG)
elverton,cunha@dcc.ufmg.br

## ABSTRACT

Path changes caused by events such as traffic engineering, changing peering agreements, and link failure impact many routes in the Internet. Topology monitoring platforms perform periodic traceroute measurements toward a large number of destinations. This approach, however, is inadequate to precisely identify the extent of paths involved in the event. For example, a link failure can be restored before all routes are measured. In this work we present measurement strategies that minimize the probing cost for identifying paths impacted by a routing event. Our results show that it is possible to efficiently identify the set of paths impacted by a routing event. Our results also indicate that, when integrated to a state-of-the-art path change tracking system, our strategies significantly increase the number of path changes detected.

## 1 INTRODUCTION

Path changes are caused by *routing events* such as router reconfiguration, link failures, software errors, and scheduled maintenance. Routing events impact multiple paths in the Internet. Current monitoring techniques monitor paths independently: detecting a routing event on one Internet path does not trigger any action on other possibly-impacted paths [2, 3]. This approach leads to (i) outdated routing information, as we delay remapping other paths that have changed due to the routing event, and (ii) prevents us from observing the extent of a routing event, as another routing event might happen before we remap all paths impacted by the first one [4].

We investigate how to use partial information about a detected routing event to efficiently identify which paths it impacted and to quickly remap changes. Whenever a measurement (e.g., traceroute) detects and remaps a change on the path to a destination $d$, we remeasure other paths that *intersect* (traverse) the hops on the path to $d$ impacted by the change. We characterize intersecting paths, and their changes, and propose strategies to efficiently identify possibly-impacted paths in a topology impacted by a routing event.

Our study allows the characterization of routing events with precision previously unreachable, improving the understanding of routing dynamics in the Internet for network operators and researchers. Furthermore, we shown that this precision can be obtained using simple strategies and without a significant increase in probing costs.

## 2 BACKGROUND AND PREVIOUS WORK

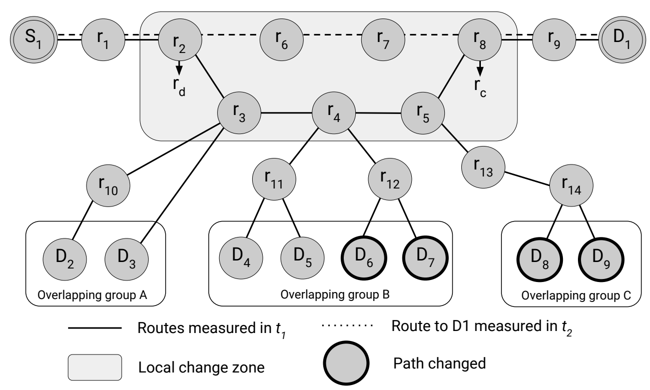


**Figure 1: Example of a route change and definitions.**

We define a local change zone, denoted LCZ, comparing two consecutive measurements (routes) of a path $P$. An LCZ is a sequence of contiguous hops removed from the previous measurement plus the immediately surrounding divergence ($h_d$) and convergence hops ($h_c$) present on both previous and current measurements. Each LCZ is computed minimizing the edit distance between consecutive measurements and comparing the set of interfaces at each hop [1]. The example in Fig. 1 shows two consecutive measurements $R_{d_1}(t_1)$ and $R_{d_1}(t_2)$ of the path from source $S_1$ to destination $D_1$ ($P_{d_1}$) represented by solid and dotted lines, respectively. The two measurements remap a path change, with an LCZ containing hops $\{r_2, r_3, r_4, r_5, r_8\}$, $r_d = r_2$, and $r_c = r_8$.

A LCZ can be intersected by routes from other paths. More precisely, every path $P$ whose *last* known route has a hop in the LCZ, is called an *intersecting path*. We group intersecting paths with identical intersections in *intersecting groups*. In Figure 1, routes $R_{d_2}(t_1)$ and $R_{d_3}(t_1)$ are in the same intersecting group which has two hops in the change zone. Alternatively, $R_{d_8}(t_1)$ and $R_{d_9}(t_1)$ are in another intersecting group which has four hops in the change zone.

**Previous findings:** In a previous work [6], we showed several insights to probe events. First, we can check if an intersecting path has changed (or not) by sending a single probe to any hop of the intersecting path that overlaps with the LCZ, except the divergence hop $h_d$ which rarely detects a path change. Second, we showed the larger the intersection of an intersecting path with the change zone, the larger the probability it changes. Third, we find that paths in the same intersecting group tends to share fate, i.e., when a probe verifies a change in a group, we can consider that there is a high probability that other paths in the same group also changed.

# 3 PROBING EVENTS UNDER BUDGET RESTRICTIONS

Real network monitoring systems control the probing rate to reduce the impact of measurements on traffic [2, 7]. Considering that a monitoring system operates with a probing budget (specified as probes per second), it is necessary to determine a sequence of intersecting paths to be probed that maximizes the utility of the available probes, i.e., a sequence that let us find more changes as possible. We now discuss one probing techniques to probe routing events and a reallocation probing mechanism between routing events.

**Probing by intersecting group:** When a probe verifies a change in a group, we consider that there is a high probability that other paths in the same group also changed and we continue to probe other paths in the group. Thus, when a verification probe does not verify a change, we consider the paths in the group are unlikely to be impacted by the event and we stop probing the current group to start probing the next group. As the size of the intersection with the change zone is related to the probability of change, we probe groups in decreasing order of intersection size. Inside each group we probe intersecting paths randomly.

Figure 1 shows an example of a change zone with eight intersecting paths separated into three groups $A, B$ and $C$ where paths $P_{d_6}, P_{d_7}, P_{d_8}$ and $P_{d_9}$ change. Assume the verification budget for this event is 5 probes. The probing technique by intersecting group starts probing group $C$ and verifies the changes in paths $P_{d_8}$ and $P_{d_9}$ using 2 probes. The technique then probes the intersecting path $P_{d_6}$ in group $B$. As this probe verifies a change, the technique continues in group $B$ and probes path $P_{d_4}$. Since $P_{d_4}$ did not change, the technique move to group $A$ and sends the last probe to path $P_{d_2}$. In this example, we verified 3 of 4 changes (75%) with a verification budget covering 5 of 8 intersecting paths (62.5%).

**Reallocation scheme:** Real route monitoring systems in the Internet work with a limited budget. Considering that the verification of changes has low probability of success [6], our probing strategy ceases the verification phase to save verification probes and use them in future events. In other words, our probing strategy tries to keep a balance of probes in order to maintain the probability of lacking probes below a threshold $L_{\text{deficit}}$. Using a conservative estimate of the change probability of paths, denoted $p_{\text{change}}$, if the balance of probes is sufficient to verify changes in $k$ routing events, the probability of lacking probes is given by a geometric distribution:

$$p_{\text{lackage}} = p_{\text{change}}^{k}(1 - p_{\text{change}}) < L_{\text{deficit}}$$

We keep a sliding window with a history of the last $W$ routing events to estimate $p_{\text{change}}$ and $k$. We calculate $p_{\text{change}}$ as the fraction of probes sent that verified a change. The higher the $p_{\text{change}}$, the higher the balance that needs to be stored to cover frequent (future) events with changes. Note that $p_{\text{change}}$ overestimates the probability of change, since the probing process concentrates probes in groups with many changes. We calculate $k$ dividing the current balance of probes by the average number of probes we need to use from the saved-up budget to verify changes in an event where all paths change (i.e., the difference between the average number of intersecting paths per event and the average verification budget used in the last $W$ past events). The lower the number of probes we need to use from the balance to verify changes in an event, the lower the balance that needs to be stored.

**Discussion:** We compared our probing techniques with machine learning algorithms (e.g., Random Forest and SVM) to predict whether an intersecting paths changes or not based on a set of features (e.g., size of the intersection with the LCZ and size of its intersecting group). We find that our techniques have a similar performance which let us argue for the use of our probing technique, which is significantly simpler, intuitive, does not maintain state and does not require training. Also, we find that our techniques improve the measurements of a routing event when compared to DTRACK, a state-of-the-art measurement tool [5], by verifying a large number of intersecting paths that have changed. More specifically, recent results (omitted due space) show that our techniques finds two and five more intersecting paths that changed in scenarios with low and high probing budget, respectively.

# 4 FUTURE WORK

As our techniques are complementary to other probing mechanisms – our focus is to quickly identify and remap possibly-impacted changes – we plan to enhance our techniques and integrate them to state-of-the-art tools such as DTRACK. Also, we plan to expand our study to understand the relation of a routing event detected in a monitor among the others, helping us to manage probing budget across multiple monitors.

# REFERENCES


[1] B. Augustin, T. Friedman, and R. Teixeira. 2011. Measuring Multipath Routing in the Internet. *IEEE/ACM Trans. Netw.* 19, 3 (2011), 830–840.

[2] K. Claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov. 2009. Internet Mapping: from Art to Science. In *Proc. IEEE CATCH*.

[3] Italo Cunha, Pietro Marchetta, Matt Calder, Yi-Ching Chiu, Bruno V. A. Machado, Antonio Pescapè, Vasileios Giotsas, Harsha V. Madhyastha, and Ethan Katz-Bassett. 2016. Sibyl: A Practical Internet Route Oracle. In *Proc. USENIX NSDI*.

[4] Í. Cunha, R. Teixeira, and C. Diot. 2011. Measuring and Characterizing End-to-End Route Dynamics in the Presence of Load Balancing. In *Proc. PAM*.

[5] Í Cunha, R. Teixeira, D. Veitch, and C. Diot. 2014. DTRACK: A System to Predict and Track Internet Path Changes. *IEEE/ACM Trans. Netw.* 22, 4 (2014), 1025–1038.

[6] Elverton Fazzion, Ítalo Cunha, Dorgival Guedes, Wagner Meira, Jr., Renata Teixeira, Darryl Veitch, and Christophe Diot. 2016. Efficient Remapping of Internet Routing Events. In *SIGCOMM (SRC)*.

[7] Y. Shavitt and U. Weinsberg. 2009. Quantifying the Importance of Vantage Points Distribution in Internet Topology Measurements. In *Proc. IEEE INFOCOM*.