Coordinated Remapping of Internet Routing Events

Elverton Fazzion[1], Ítalo Cunha[1], Dorgival Guedes[1], Wagner Meira Jr.[1]
Renata Teixeira[2], Darryl Veitch[3], Christophe Diot[4]

[1]Universidade Federal de Minas Gerais [2]Inria [3]Dept. Elec. and Elec. Eng. University of Melbourne [4]Safran

ABSTRACT

A number of systems rely on traceroute to track a large number of Internet paths as they change over time. Our previous work designed DTRACK, a system that optimizes probing to track Internet path changes. DTRACK only remaps a path, i.e., probes every hop from the source to the destination of the path, after it detects that the path has changed. This paper shows that such complete remapping is inefficient, because most path changes are localized in a few hops of a path. We develop a new version of DTrack with local remapping, which just probes the (often few) hops that have changed. Our evaluation with trace-driven simulations and in a PlanetLab deployment shows that local remapping reduces by 73% the number of probes issued during remapping when compared with complete remapping. Yet, local remapping has little impact on the accuracy of inferred paths.

1. INTRODUCTION

A number of distributed services and applications need to measure Internet paths to maintain an up-to-date view of the underlying infrastructure [?, ?, ?, ?, ?, ?]. All these systems use some version of traceroute to repeatedly measure a large number of Internet paths. Traceroute sends probes to every hop between a source and a destination, so the measurement of a single path often requires tens of probes. The number of probes required to measure a path is even larger if one wants to measure all paths between a source and a destination when routers perform load balancing [?].

Network measurement and monitoring platforms should not impact network traffic. As such, measurement platforms operate on a limited probing budget [?, ?, ?, ?]. For example, CAIDA's Ark [?] measures routes to all routable /24 prefixes probing from around 18 monitors at 100 packets per second per monitor and RIPE Atlas [?] collects path and ping measurements from more than 4,000 devices at an average rate of 6kbps per device.

As a result of limited budget, path measurements take time. Discovering a (multi)path with Paris traceroute, which is a traceroute version that discovers all paths under load balancing, requires hundreds of probes and takes tens of seconds [?].[1] Because measuring each path takes time, a source cannot measure paths frequently and as a result it may miss path changes. For instance, topology mapping systems can take from several minutes to a few days to measure all required paths [?, ?, ?]. In between measurements, paths may be outdated or inconsistent.

Our previous work showed that Internet paths being mostly stable, measuring all paths at the same frequency wastes probes in paths that are not changing [?]. We developed DTRACK, a system that optimizes probing to track Internet path changes [?]. DTRACK splits the task of tracking paths into two sub-tasks. *Path change detection* sends a single probe per path at any given time, probing more frequently paths that are more likely to change. *Path remapping* responds to a path change detection by immediately sending probes to discover all interfaces of the new multipath. Although DTRACK detects twice as many path changes as traditional probing [?], remapping a path remains costly. DTRACK simply runs Paris traceroute to remap the entire end-to-end path. Such complete remapping ensures the accuracy of inferred paths, but incurs a large probing overhead.

In this paper, we show that complete remapping wastes probes because most path changes are localized in a few (consecutive) hops of a path (Sec. ??). We build on this observation to develop a more efficient remapping method for DTRACK (Sec. ??). Given knowledge of the path before the change and the hop where the change was detected, we first send probes to precisely locate the change and then just remap the (generally few) hops that have changed. We call this method *local remapping* as opposed to the complete remapping originally implemented in DTRACK. Local remapping still uses Paris traceroute's multipath detection algorithm [?] for discovering all interfaces at a given hop to guarantee accuracy, but we no longer have to remap every single hop of the path.

Our evaluation via trace-driven simulations shows that local remapping reduces the remapping probing cost of 88% of path changes in our dataset by more than half, and reduces overall cost by 73% (Sec. ??). Local remapping has two limitations: (i) it may only remap part of the change when paths change in multiple locations; and (ii) it does not work when a path change reorders the interfaces of the path.

[1]It is essential to accurately measure paths under load balancing to avoid traceroute errors and misinterpretation of path changes [?].