PathCache: A Path Prediction Toolkit

Rachee Singh Phillipa Gill
Stony Brook University
{racsingh, phillipa}@cs.stonybrook.edu

1. INTRODUCTION

Recent years have seen the proliferation of network measurement platforms such as RIPE Atlas [6] and CAIDA's Ark [3], with each platform supporting millions of measurements each day. These platforms represent a valuable, yet limited, resource for network measurements, both in terms of scheduling measurements to probes and maintaining the back end storage systems required to support the platforms. In this poster, we present, PathCache, a system that allows researchers to reuse traceroute measurements that are *already being run* to learn properties of network paths.

Our goal is to support research that is not well-served by existing measurement platforms. We build an interface that can provide path information without having to wait for new measurements to complete. Since PathCache bootstraps off of multiple empirical data sets, it is able to combine them to predict paths that have not been directly measured. Further, PathCache provides higher accuracy path prediction than algorithmic simulations that have been used in prior work [7]. PathCache aims to reduce the load on existing measurement platforms by making it easy for researchers to reuse measurements via an easy-to-use unified interface.

Predicting unmeasured paths. An important feature of PathCache is its ability to predict paths even when a direct measurement was not possible. We leverage a path-stitching approach based on the assumption of destination based routing to combine disparate traceroute measurements to predict paths. We develop a methodology to resolve violations of destination-based routing by assigning a confidence values to computed paths.

SIGCOMM '16, August 22–26, 2016, Florianopolis, Brazil

DOI: http://dx.doi.org/10.1145/2934872.2959053

Improving prediction accuracy. A common approach for predicting paths when a large number of paths need to be computed, or when the paths cannot be measured directly, is to employ algorithmic simulations [4]. However, their accuracy is inadequate for security sensitive applications such as Tor clients that aim to avoid eavesdroppers [1]. PathCache infers high accuracy paths without requiring the client to issue its own measurements. Further, PathCache includes metadata about the measurement and data source that led to a given predicted path, allowing easy validation and evaluation of predicted paths.

Easy interfacing with PathCache. Our goal with PathCache is to make the system as simple to use as possible, so that application developers will leverage reused measurements and avoid issuing new ones whenever possible. For Tor clients that cannot issue queries to a central server–without revealing their intended destination–we implement a downloadable version of PathCache which can be supplied with nightly updates which are on the order of 15MB. We also have exposed a REST API to make PathCache readily accessible to researchers and network operators.

We build and evaluate PathCache using traceroute data from CAIDA's Ark platform [3], RIPE Atlas [6], and iPlane's traceroute data [5]. We further augment the data-plane path data with paths observed via BGP monitors, using the BGPStream API [2]. In the following sections, we describe the design of the PathCache system §2 and demonstrate the benefits of incorporating multiple datasets in terms of coverage and accuracy §3.

2. PATHCACHE SYSTEM DESIGN

PathCache aggregates multiple data- and control-plane data sets from publicly available measurements and BGP route monitors. Traceroutes are converted to AS-level paths which are then combined to produce destination-based AS graphs that are used to derive network paths. These paths are then provided to users of PathCache via a REST API and client side interface. Figure 1 overviews the architecture of PathCache.

From traceroutes to AS-level paths. We take a best-effort approach to converting traceroutes to AS paths (*i.e.,* removing IXPs) but note that improved IP

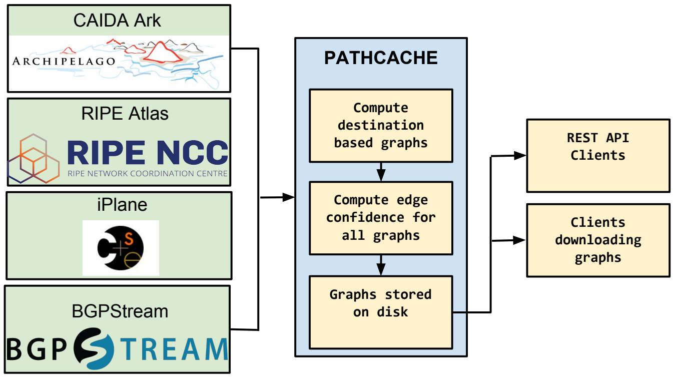

Figure 1: Overall PathCache system design

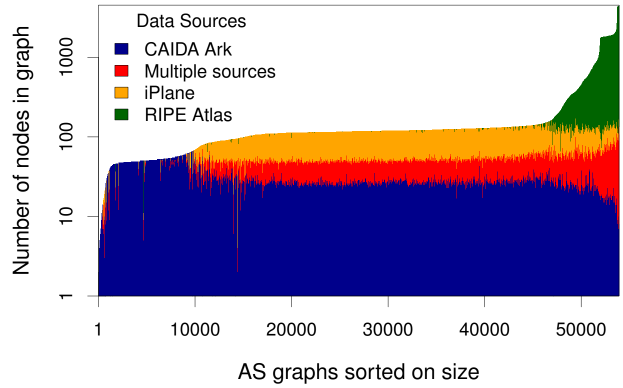

Figure 2: Benefit of different data sources in building destination based graphs

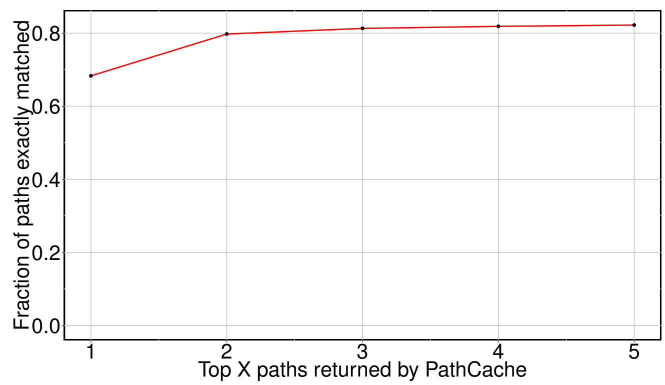

Figure 3: Accuracy of PathCache on "high traffic" paths.

to AS path mapping solutions would only improve the accuracy of PathCache. We create an AS graph for each destination AS including edges used by other ASes to reach the destination.

Merging measured paths to infer new paths. We use these AS graphs to estimate AS paths. Violations of destination based routing present themselves as multiple paths between a source and a destination. We assign a confidence value to each of the paths (a function of the number of times the path is observed in measurements) that can be used to highlight more likely paths.

Usability of PathCache. We expose two interfaces for PathCache. In the first interface, a software client can download the set of destination-based graphs that comprise PathCache from our central server (appx. 125MB). Using this data, the client can then satisfy path queries locally without leaking information to a third party. In addition to the client-side path prediction, we also support a REST API for PathCache. The REST API provides an easy way for researchers to experiment with the platform.

3. EVALUATING PATHCACHE

Benefits of merging data sets. Figure 2 shows the number of ASes in each of the destination-based trees (*i.e.,* the number of ASes we are able to predict paths towards the destination for). The destinations with the most sources are those that are covered by RIPE experiments that can leverage thousands of vantage points. In contrast, iPlane and Ark data that traceroutes increase the breadth of our coverage.

Accuracy of PathCache We evaluate PathCache over a set of paths from top 200 ASes in the world (based on the number of Internet users from an AS) to top 200 Alexa websites. We find that the exact path between the sources and destinations is present in the five highest confidence paths returned by PathCache, 82% of the time (Fig 3). This is nearly 20% higher than simulation based approaches which infer the correct path 65% [1] of the time.

4. REFERENCES

[1] R. Anwar, H. Niaz, D. Choffnes, I. Cunha, P. Gill, and E. Katz-Bassett. Investigating interdomain routing policies in the wild. In *ACM IMC*, 2015.
[2] BGPStream. https://bgpstream.caida.org/.
[3] CAIDA Ark. http://www.caida.org/projects/ark/.
[4] P. Gill, M. Schapira, and S. Goldberg. Modeling on quicksand: Dealing with the scarcity of ground truth in interdomain routing data. *ACM SIGCOMM Computer Communication Review*, 42(1):40–46, 2012.
[5] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An Information Plane for Distributed Services. In *Proc. of Operatings System Design and Implementation*, 2006.
[6] RIPE Atlas. https://atlas.ripe.net/.
[7] O. Starov, R. Nithyanand, A. Zair, P. Gill, and M. Schapira. Measuring and mitigating as-level adversaries against tor. *arXiv preprint arXiv:1505.05173*, 2015.