Computer Networks 56 (2012) 1029–1040

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A structural approach for PoP geo-location ☆

Dima Feldman, Yuval Shavitt, Noa Zilberman *

School of Electrical Engineering, Tel-Aviv University, Israel

ARTICLE INFO

Article history:
Received 23 February 2011
Received in revised form 13 September 2011
Accepted 25 October 2011
Available online 16 December 2011

Keywords:
PoP
Geolocation
Internet topology

ABSTRACT

Inferring PoP level maps is gaining interest due to its importance to many areas, e.g., for tracking the Internet evolution and studying its properties. In this paper we introduce a novel structural approach to automatically generate large scale PoP level maps using traceroute measurement from multiple locations. The PoPs are first identified based on their structure, and are then assigned a location using information from several geo-location databases. We discuss the tradeoffs in this approach and provide extensive validation details. The generated maps can be widely used for research, and we provide some possible directions.

1. Introduction

Mapping the Internet and studying its evolution has become an important research topic. Internet maps are presented in several levels of aggregation: from the AS level, which is the most coarse, to the finest level of routers, each level of abstraction is suitable for studying different aspects of the network. The Autonomous Systems (AS) level is most commonly used to draw Internet maps, as it is relatively small (tens of thousands of ASes) and therefore relatively easy to handle. The disadvantage of using AS information for Internet evolution study is that AS sizes may differ by orders of magnitude. While a large AS can span an entire continent, a small one can serve a small community. Obviously, it is hard to correlate large ASes to geographic location due to their span, but network evolution is triggered by economic factors that may be restricted to much smaller areas than those spanned by large ISPs. Router level maps represent the other extreme: they contain too many details to suit practical purposes, and the large number of entities makes them very hard to handle.

Service providers tend to place multiple routers in a single location called a Point of Presence (PoP), which serves a certain area. Thus for studying the Internet evolution and for many other tasks, PoP maps give a better level of aggregation than router level maps with minimal loss of information. PoP level graphs provide the ability to examine the size of each AS network by the number of physical co-locations and their connectivity instead of by the number of its routers and IP links, which is an important contribution. The points of presence are not only counted, but also provided with a geographical location and information about the size of the PoP. Using PoP level graphs one can detect important nodes of the network, understand network dynamics, examine types of relationships between service providers as well as routing policies and more.

This paper focuses on PoP level map generation, based on an algorithm described in Section 3. The traceroute measurements used in this work were generated by DIMES, a highly-distributed Internet measurements infrastructure [1]. DIMES achieves high distribution of vantage points by employing a community based distribution methodology that uses Internet users' PCs for measurements.

2. Related work

While aggregating IPs to AS is a fairly simple task, PoP level maps are more difficult to create. Andersen et al. [2]

☆ This work was partially funded by the Israeli Science Foundation's center of knowledge Grant 1685/07.

* Corresponding author.

E-mail addresses: shavitt@eng.tau.ac.il (Y. Shavitt), noa@eng.tau.ac.il (N. Zilberman).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.10.029

used BGP messages for clustering IPs and validated their PoP extraction based on DNS. Rocketfuel's [3] generated PoP maps using tracers and DNS names. The iPlane project also generates PoP level maps [4] by first clustering router interfaces into routers by resolving aliases, and then clustering routers into PoPs by probing each router from a large number of vantage points and using the TTL value to estimate the length of the reverse path, with the assumption that reverse path length of routers in the same PoP will be similar.

Assigning a location to an IP address, let alone a PoP, is a complicated task. The most common way to do so is using a geolocation service. Geolocation services range from free services to services that cost tens of thousands of dollar a year. The most basic services use DNS resolution as the basis for the database [3], while others use proprietary means such as random forest classifier rules, hand-labeled hostnames [5], user's information provided by partners [6] and more. IP2Geo [7] was one of the first to suggest a measurement-based approach to approximate the geographical distance of network hosts. A more mature approach is constraint based geolocation [8], using several delay constraints to infer the location of a network host by a triangulation-like method. Later works, such as Octant [9] used a geometric approach to localize nodes within a 22 miles radius. Katz-Bassett et al. [10] suggested topology based geolocation using link delay to improve the location of nodes. Yoshida et al. [11] used end-to-end communication delay measurements to infer PoP level topology between thirteen cities in Japan. Laki et al. [12] increased geolocation accuracy by decomposing the overall path-wise packet delay to link-wise components and were thus able to approximate the overall propagation delay along the measurement path. Eriksson et al. [13] applied a learning based approach to improve geolocation. They reduced IP geolocation to a machine learning classification problem and used Naive Bayes framework to increase geolocation accuracy.

In this paper we present a structural approach for creating large scale PoP maps with geographic information. We study the effect of the volume and quality of the data on the algorithm and provide detailed validation of the algorithm and its results.

3. PoP discovery

3.1. PoP extraction algorithm

We define a PoP as a group of routers which belong to a single AS and are physically located at the same building or campus. In most cases [14,15] the PoP consists of two or more backbone/core routers and a number of client/access routers. The client/access routers are connected redundantly to more than one core router, while the core routers are connected to the core network of the ISP. Fig. 1(a) shows a simple interconnection of four routers with a small number of interfaces. Assuming that during traceroute measurements ICMP replies are received from the incoming interfaces of the routers, the graph shown in Fig. 1(b) is obtained. For example a traceroute measurement that enters our network through *interface A* on *router a* and leaves the network from *interface L* on *router b* will create an $A \rightarrow I$ path on the graph. In a similar way a measurement that enters the network from *interface L* on *router b* and leaves it from *interface W* on *router c* will create a $L \rightarrow C \rightarrow Y$ path on the graph. At the core of the Interface graph, which results from performing many traceroute measurements through a PoP, there is clearly a bi-partite graph. We look for this specific structure when trying to discover PoPs. Alon et al. [16] showed that many complex networks have repetitive patterns of interconnections, called 'network motifs', which became a standard term in the networks analysis community. Their work showed that real-world networks outside the communication field are not purely random, but have a higher than (or lower than) expected number of specific motifs. We have used their *mfinder* [17] package to search for motifs in graphs obtained by the DIMES measurements. In order to show the significance of a specific motif, the software uses the Z-score measure, which is calculated according to Eq. (1).

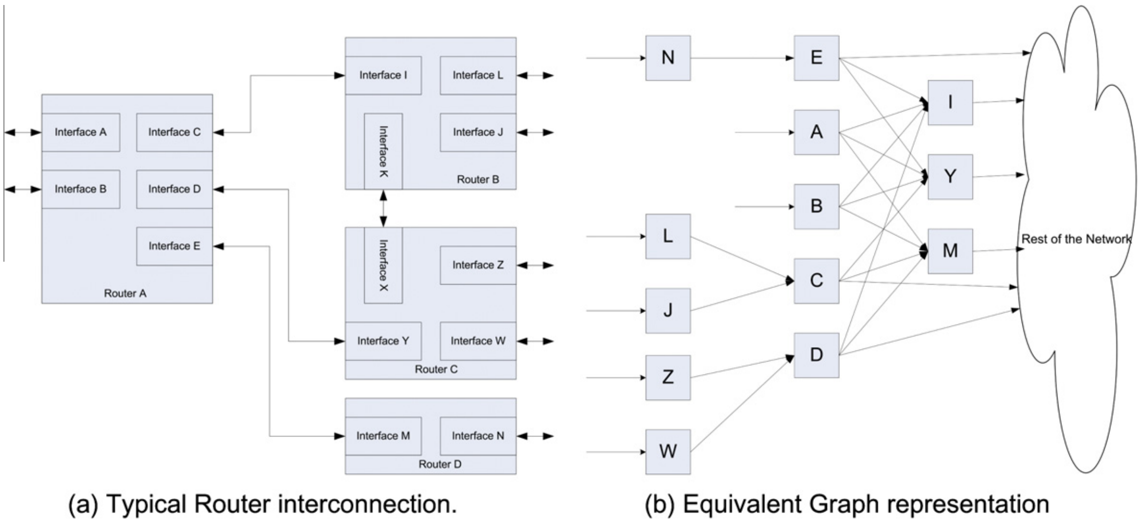
$$Z = \frac{X - \mu}{\sigma}, \tag{1}$$

where X is a number of a motif occurrences in a specific network, and μ and σ are the mean and standard deviation of the motif occurrences within a certain random network. The number of motif appearances in a random network is a stochastic function with mean and variance. The Z-score reveals how many units of the standard deviation a specific count of a motif is above or below the mean. Unsurprisingly, we have found a number of motifs with a high Z-score across all AS networks in the graph; partial results displayed in Table 1 show the clear dominance of the 'bi-fan' motif (number 204) in three large providers, Global Crossing, France Telecom and Broadwing (now Level3). Note that motif 460 is bi-fan with one additional measurement in the reverse direction and motif 206 is a bi-fan with an additional measurement.

Although *mfinder* [17] is a very useful tool for identification of important motifs, it is not designed to be used for network clustering. In our work we do not look for a specific motif in the network, but for highly connected clusters as described in the previous chapter. However, we do search for 'bi-fan's (id204) repetitions under certain weight constrains as cores of the PoPs. The cores are extended with other close by interfaces. The following steps, introduced in [18], are used to reduce the IP level graph G (V, E) to a PoP level network:

Initial Partition. Remove all edges with a delay higher than PD_{max_th}, the PoP maximal diameter threshold, and edges with number of measurements below PM_{min_th}, the PoP's edges measurements threshold. PM_{min_th} is introduced in order to consider only links with a highly reliable delay estimation to avoid false indication of PoPs. As a result, a non-connected graph G' is obtained. Then, for each connected component of G' an induced sub graph is built by adding back all the edges that connect nodes of the connected component. Each connected group is a candidate to become one or more PoPs.

There are two reasons for a connected group to include more than a single PoP. First and most obvious is geographically adjacent PoPs, e.g., New York, NY and Newark,

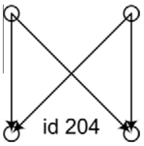
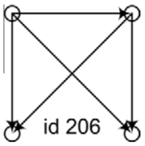
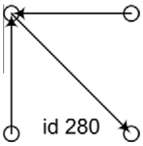
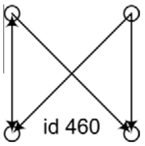
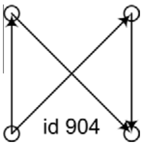


(a) Typical Router interconnection.

(b) Equivalent Graph representation

Fig. 1. Typical network connection.

Table 1
Common network motifs in IP interconnections networks of three ASes.

					
AS number	Z-score				
AS6395	377	–	9.51	43.84	148.39
AS5111	329.29	36.42	–	74.63	73.57
AS3549	154.8	5.38	37.87	19.51	–

NJ. Second is wrong delay estimation of a small number of links. For instance a single incorrectly estimated link between Los Angeles, CA and Dallas, TX might unify the groups obtained by such a naive method.

Refined partition.

(*a*) *Parent–child classification*. The next stage in the algorithm uses a classification to *parent pairs* and *child pairs*.

Definition 3.1. A pair of nodes is marked as *parent pair* if **both** of them point to two or more nodes.

Definition 3.2. A pair of nodes are marked as *child pair* if **both** of them are pointed to by at least two nodes.

All *parent pair* nodes are assigned to groups by pairwise unifying *parent pair* nodes. For example in Fig. 3, nodes {1,2}, {2,5} and {3,4} are defined as *parent pair*, thus we obtain two *parent pair* groups {1,2,5} and {3,4}. The groups of *child pair* nodes are created according to the same process as defined for *parent pair* groups. Some nodes might belong to both categories and it is allowable for a node to belong to one *parent pair* group and to one *child pair* group. By definition, if a node belongs to two or more groups of the same kind, these groups are unified. Fig. 2 shows an example of *parent/child* classification.

The PoP algorithm checks for each connected group extracted in the initial partitioning of the algorithm, if it contains more than one possible PoP. Note that each candidate partition looks like a collection of highly connected bipartite graphs with rich connectivity between them. The considered partition of parents and children is then divided according to the measurement direction in the bipartite graph (each node or a group of nodes simultaneously can be a parent of one bipartite and a child of another). In this operation the weights of the edges are ignored. The minimal size of each group is two nodes.

(*b*) *Localization*. Dividing the parents and children groups into physical collocations using the high connectivity of the bipartite graph. The input for the localization stage algorithm is a highly connected bipartite graph $G(V,E)$ with a weight function $W : E \rightarrow \mathbb{R}$ representing the estimated physical link delay, as shown in Fig. 3. The other input to the algorithm is a partition of the graph to the *parent/child* groups as previously described. The localization algorithm checks whether nodes of the same type (*parent/child*) belong to the same physical collocation. For this task the algorithm takes advantage of the topological structure of the group. For instance, if we check the parent group

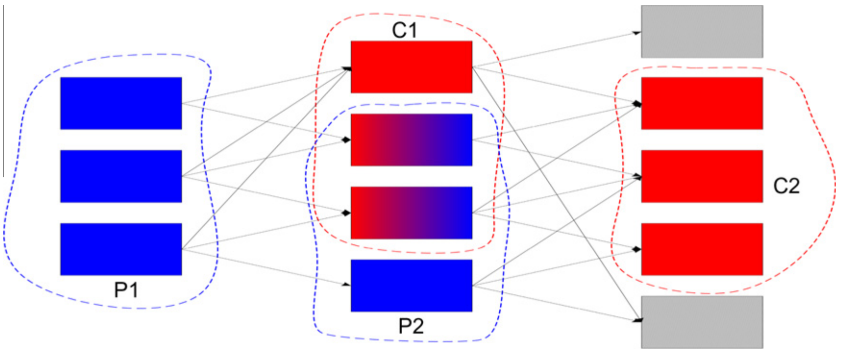


Fig. 2. Parent–child classification: blue nodes (left) – *parent pair*, red nodes (right) – *child*, blue and red nodes (middle) – both parent and child, gray stripes nodes (right) – not classified.

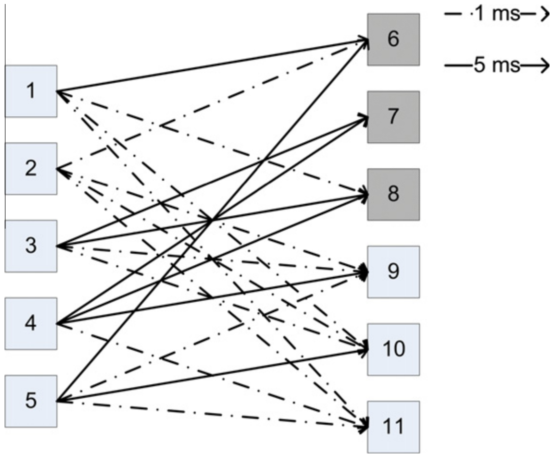


Fig. 3. Bipartite graph example, on the right side dark and bright nodes belongs to different collocation.

P we note that each child node of the group is pointed to by at least two parent nodes. Comparing the delays from the *child pair* nodes we can partition nodes of the *parent pair* group to one or more geographic collocations.

Formally, we represent each member of a group of two or more nodes (either *parent pair* or *child pair* group) in a coordinate space of the nodes that points to them using the weight of the edges. Next, we check the distance between each pair of nodes in that coordinate space. We assume that the link delay estimation errors in [19] are caused mainly by an impulse noise, i.e., most of the measurements are fairly precise or have only small noise, while a small portion of the measurements may have large errors. Therefore, unlike the Gaussian noise case, where Euclidean distance is used as a representation of the distance between nodes, we compare the similarity over the coordinates.

An example of the difficulties in determining geographic co-location is shown in Fig. 3. By looking at the delay spread, one can easily determine that nodes 6–8 (darken) are not co-located with nodes 9–11. Looking at the distance between nodes 1–3 and nodes 9–11 it becomes clear that the former are also co-located. However deciding whether node 5 is also collocated with nodes 1–3 is not straightforward. Examining the delay spread between nodes 5 and 1–3 to nodes 9 and 11, gives a positive answer for collocation, while the measurement to node 10 that puts node 5 away from nodes 1–3 might be discarded as noise. The existence of yet another group of measurements to node 6, which is indecisive in its results, complicates the picture, and shows the difficulties in automating these decisions.

We propose the following deterministic algorithm to classify the locations of nodes in the bipartite graph. For each pair of parent nodes $(u,v) \in P$, $u \neq v$, we define the 'common children' group, *CC* by

$$CC(u,v) = \{w \in G | (u,w) \in E \bigcap (v,w) \in E\}. \tag{2}$$

We denote the members of $CC(u,v)$ as $\{cc_1, cc_2, \ldots, cc_m\}$. Then using the weights of the edges from the pair of parent nodes to the 'common children', $W(u,cc_i)$ and $W(v,cc_i)$, we calculate the 'Error Ratio' vector, *ER*:

$$\overline{ER(u,v)} = \left[\frac{W(u,cc_1)}{W(v,cc_1)}, \frac{W(u,cc_2)}{W(v,cc_2)}, \ldots, \frac{W(u,cc_m)}{W(v,cc_m)}\right]. \tag{3}$$

The selection between (u,v) and (v,u) for a numerator and a denominator results in identical results when observing $|\log(\overline{ER(u,v)})|$ due to the properties of logarithms. Another important property of $|\log(\overline{ER(u,v)})|$ is that for coordinates with a small relative error, the values of the elements in $ER(u,v)$ will be rather small, and will increase with a loss of the accuracy. Therefore comparing $er(u,v) = median(|\log(\overline{ER(u,v)})|)$ to a certain threshold gives a proper indication of the accuracy in the majority of measurements.

We use the *er* values for the parents, to partition parents groups into smaller parent groups which are geographically collocated. To this end, we produce a weighted clique of all the parent nodes in a group, where the weight of the edge (u,v) is $er(u,v)$. We remove all the links with a weight above a certain small threshold. Each connected component in the remaining graph becomes a parent group for the next step. To summarize, we partitioned the parent group to several parent groups that are geographically co-located.

The same process is repeated for child groups, where the error vectors are calculated by the distances to the common parents.

This kind of localization helps us to overcome a relatively large number of errors. However, if more than half of the measurements to a certain node are incorrect, the algorithm may fail to determine its location. Otherwise, there is no impact on the overall performance. Those 'badly' measured nodes might not became a part of the correct PoP, but the PoP map will be formed correctly in spite of them, i.e., no new PoPs will be created.

(*c*) *Unification*. Unifying *parent*/*child* group to the same PoP. If a *parent pair* and a *child pair* groups are connected, then the weighted distance between the groups is calculated (if they are connected, by definition more than one edge connects the two groups); if it is smaller than a certain threshold, PPC_{max_th}, the pair of groups is declared as part of the same PoP.

Final refinements.

(*a*) *Unification of loosely connected components*. In some cases, e.g., due to insufficient measurements, different parts of a PoP are only loosely connected in a way that does not form even a 2×2 bi-partite; in the extreme case only a single link connects two parts of a PoP. This will not allow the unification process, just described above, to identify the parts as belonging to the same PoP. Thus, the algorithm looks for connected components (PoP candidates) that are connected by links whose median distance is very short (below PD_{max_th}). Note that at this point, due to the unification process, the graph has shrunk considerably, and thus the search for 'close' components is inexpensive.

(*b*) *Singleton Treatment.* At the end of the process, the ISP graph has evolved through the multiple node unifications described above into a graph that is comprised of several multi-nodes (the PoPs) and a larger number of nodes (IP interfaces) that were not assigned to any PoP. Typically, these nodes have only one or two links connecting them to the rest of the graph, and the path from a node to the closest PoP is in most cases one hop and sometimes two. This final step assigns many of these nodes to existing PoPs. The assignment is conducted by running a Dijkstra shortest path algorithm from a node to all PoPs, and connecting a singleton to the closest PoP, providing the distance (in mSec) is below a given threshold PD_{max_th}.

While this step has some advantages, it typically degrades the algorithm accuracy and does not add to the number of discovered PoPs. Therefore, unless noted differently, it is eliminated in most presented results. We discuss the effect of Singletons in Section 3.2.

3.2. PoP extraction validation and results

Following, we present our validation tests and the results of a full implementation. The validation is then extended to discuss tradeoffs in the algorithm's implementation and their effect on result's accuracy.

Two collected datasets for PoP extraction are taken from DIMES [20]. One is from 2009, with a focus on weeks 27 to 30 for specific examples, and the other taken from weeks 42 to 43 of 2010. The database from weeks 27 to 30, 2009 includes 56 million traceroute measurements, collected by 1415 agents. The 2010 database, from weeks 42 to 43, has a total 33 million measurements, an average of 2.35 million measurements a day. The measurements were collected by 1308 agents, which were located in 49 countries around the world.

First, we examine the best time period length for collecting measurements for PoPs, and select it to be two weeks. DIMES produces five to six million daily measurements, both traceroute and ping, meaning thirty to forty million measurements per week, which typically result in 5.5 M to 6.5 M distinct IP edges being discovered. The selection of a two weeks time period balances between two delicate tradeoffs: the number of distinct edges used for the PoP construction and the sensitivity to changes in the network. A time frame of a single week is too short, with considerably fewer distinct edges than those from two weeks. A month, on the other hand, does add many more edges, but it is insensitive to changes in the network, which we would like to track. In addition, the algorithm runs considerably slower on such large data sets. Table 2 shows the changes in PoP maps between different time frames. The first row in the table shows the difference in PoP maps between two consecutive weeks. The second row refers to a one week period compared to two weeks, and the last row compares two to four weeks measurements collection periods. The columns "#PoPs" and "#IPs in PoPs" refer to the change in number of discovered PoPs and IPs included in these discovered PoPs over the compared periods. "#Distinct Edges" refers to the change in distinct IP edges measured by DIMES. This number is independent of the PoP algorithm.

We set PM_{min_th}, the minimal number of node's measurements, to be 5. This threshold was found to be optimal over many heuristic test cases, cleaning noisy measurements while filtering out only a small number of edges. We then ran the median algorithm described in [19] to find the delay between two adjacent nodes.

The resulting IP address to PoP mapping table typically consists of over 50,000 IP addresses, in about 4000 different PoPs. The average size of a PoP is 16 IP addresses, with a median of 6. The largest PoP size observed was 2500. The size of the discovered PoPs depend both on our measurement method and the ISP's policies. When a PoP is measured from many different agents or there are many paths between the source and destination nodes, the size of the PoP will be larger. However, measuring from one direction or if there is a relatively small number of

Table 2
Changes in PoP maps between different time frames.

Compared time frame	#PoPs	#IPs in PoPs	#Distinct edges
1 week to 1 week	<1%	<1%	±20%
1 week to 2 weeks	+58%	+79%	+43%
2 weeks to 4 weeks	+10%	+15%	+59%

alternative routes, the size of the discovered PoP will be small. The policies of the ISP can cause nodes inside the PoP to not answer traceroute messages and become anonymous or transparent e.g., due to use of MPLS.

On a single day, DIMES may run several experiments in parallel, however, the vast majority of the measurements performed over a week belong to the DIMES default experiment where a set of roughly 2.5 million target IP addresses, selected to cover all the allocated IP address prefixes, are cyclically sent to the agents. To test whether the target set limits us from discovering more PoPs, 2.5 million IP addresses were added to this basic experiment, identified by the iPlane project [4] as belonging to PoPs. The addition of the iPlane IP addresses increased the number of PoPs discovered by less than 20%, yet did not reach the numbers in iPlane. We believe that the immense number of IPs grouped by iPlane into PoPs partly represent IPs which are not part of the PoP.

The number of PoPs found in an AS network correlates with its measured size. Fig. 4 shows that the number of PoPs discovered per AS depends logarithmically on the number of IP edges measured. Fig. 5, showing the number of IPs included in PoPs compared to the number of IPs edges measured, demonstrates even better the logarithmic relation between the number of measurements and the discovered PoPs. As the number of IP edges reflects measurements through unique IPs and not PoPs, this is an expected outcome.

Figs. 6–9 explore the PoP extraction algorithm's sensitivity to its two parameters PD_{max_th} and PM_{min_th}. In each figure five ISPs are explored: Level 3, ATT, Comcast, MCI, and Deutsche Telekom. In Fig. 6 the number of discovered PoPs is compared with PD_{max_th}, the maximal delay threshold. Fig. 7 presents the number of IPs included in these PoPs under these conditions. Neither the number of discovered PoPs nor the number of IPs within the PoPs are sensitive to the delay threshold, as long as the threshold is $3mS$ or above. PD_{max_th} was therefore selected to be $3mS$, as it presents a good tradeoff between delay measurement error and location accuracy. Figs. 8 and 9 show the effect of PM_{min_th}, the minimal number of measurements threshold,

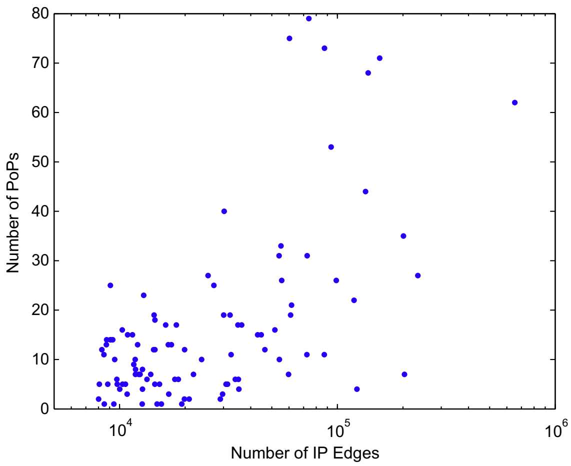


Fig. 4. Number of Discovered PoPs vs. number of measured IP edges.

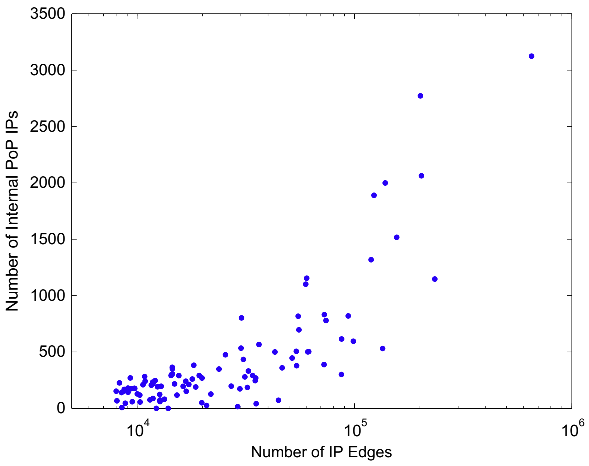


Fig. 5. Number of IPs in PoPs vs. number of measured IP edges.

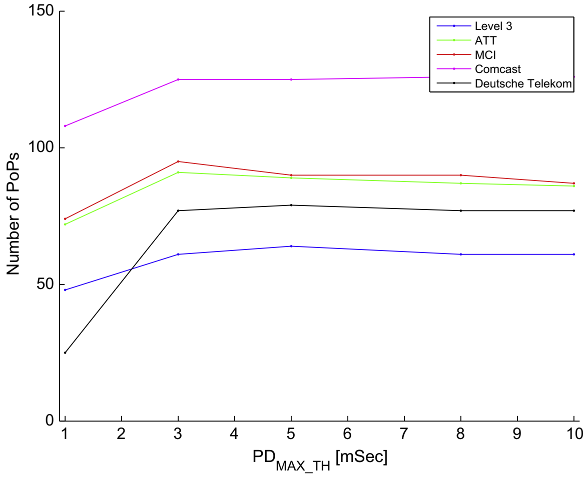


Fig. 6. Number of PoPs vs. maximal delay.

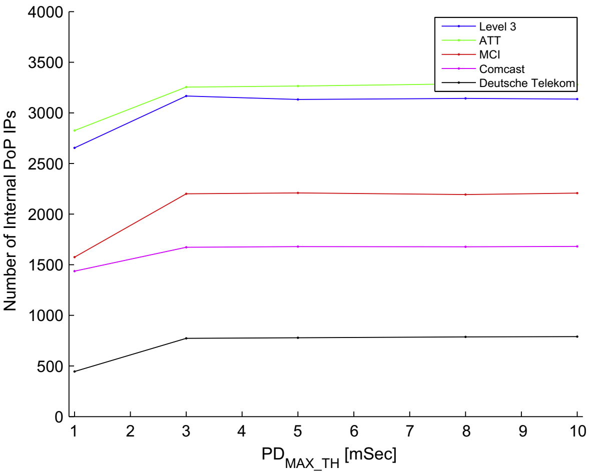


Fig. 7. Number of IPs in PoPs vs. maxi-mal delay.

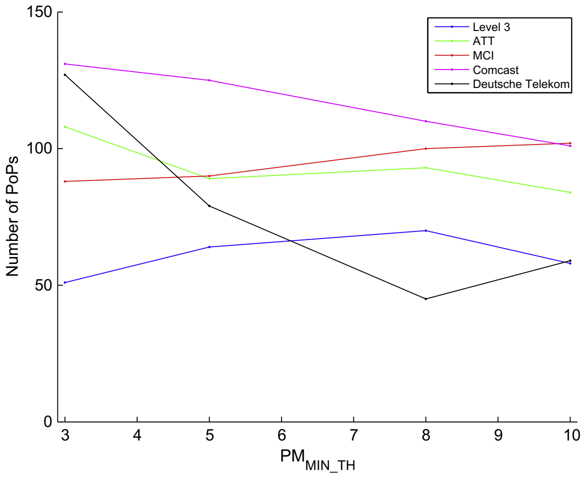


Fig. 8. Number of PoPs vs. minimal number of measurements.

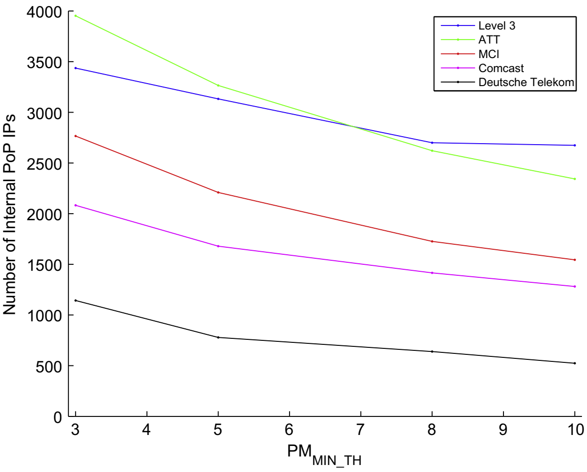


Fig. 9. Number of IPs in PoPs vs. minimal number of measurements.

on the number of discovered PoPs and the number of IPs included in them. The number of IPs included in PoPs clearly decreases as the minimal number of required measurements increases, as can be expected. The number of discovered PoPs shows a mixed behavior as the reduction of IP level links may have two conflicting outcomes; An increase is caused by a loss of connectivity inside a PoP which in turn causes it to split to several PoPs located at the same place, while a decrease is caused by the loss of the ability to identify a PoP. In our experiments, PM_{min_th} was selected to be 5.

Additional validation tests repeatedly targeted previously identified PoP IP addresses within several large ASes, such as Level3, ATT and MCI, from agents within the AS. They did not increase the number of discovered PoPs, but proved that discovered PoPs are stable. To show that the PoP algorithm succeeds when enough measurements are provided, two ASes were taken as an example: GEANT, the pan-European academic network, and Proxad, a French ISP. Both were selected since their PoP topology is public and since DIMES did not have many measurements in them by default. Comparing the amount of PoPs and IPs within PoPs discovered based on default DIMES measurements and directed measurement tests, the number of discovered PoPs more than doubled and the number of IPs within PoPs grew by a factor of ten. In both cases, the directed tests doubled the number of distinct measured edges within the AS, thus increasing the connectivity required to discover PoPs. We conclude that increasing the number of measurements improves the algorithm's performance.

Other stability tests examined the IP addresses identified as part of PoPs and found 85% similarity between consecutive fortnights. The difference between PoPs was due to lack of measurements through the PoP connecting nodes, rather than the PoP extraction algorithm. In addition, not all the traceroutes are identical every week, due to the community based nature of DIMES. Additional validation actions taken are detailed in Section 4. Validation of PoP maps was always an issue in related work, e.g., in iPlane [4] or RocketFuel [3], and we find that the level of validation introduced in this work is at least at the level of previous efforts.

4. PoP geolocation methods

Automatically assigning every discovered PoP to a geographical location is the second contribution of this work. We use geolocation services in order to find the PoP's geographic coordinates. Geolocation services provide location information regrading a given IP address, including country, city, longitude and latitude.

In the past, as Katz-Bassett et al. [10] indicated, geolocation databases were not highly reliable: They were combined from multiple sources, such as DNS hostname parsing rules, whois registration and DNS LOC records. Due to the sources of information, many of them were outdated as well. In recent years geolocation services have been widely used to countermeasure Internet frauds, for marketing, publicity and conditional access. This led to an immense effort to improve the database quality, yet not resulted in a great deal of accuracy. While some location services do not reveal their level of accuracy, country-level assignment is typically over 99% accurate, as the IP assignments to ASes are in most cases bounded within a single country. MaxMind GeoIP service [21] provides with its database accuracy information on city level, within a radius of 25 miles of true location, which ranges from 40–44% (Nigeria, Tunisia) to 94–95% (Georgia, Singapore). The United states, for example, has 83% accuracy at the city level. A further assessment of the geolocation information is therefore required. We present such an evaluation in [22], based on PoP and IP level analysis.

We use several geolocation services to maximize the accuracy of our PoP location. The initial results from 2009 used MaxMind GeoIP [21], IPligence [23], and Hostip.info [24]. The results from 2010 were extended to use also f IP2Location DB5 [25] and GeoBytes [26]. Information from Netacuity [6] and Spotter [27] was used to some extent as well.

To identify the geographical location of a PoP, we use the geographic location of each of the IPs included in it. As all the PoP IP addresses should be located within the same campus, or within its vicinity if singletons are considered, the location confidence of a PoP is significantly higher than the confidence that can be gained from locating each of its IP addresses separately. The algorithm, introduced in [28], operates as follows:

Initial location. Each of the geolocation databases used is queried for the location (longitude, latitude) of each IP included in the PoP. Next, the center of weight of the PoP location is found by calculating the median of all PoP's IP locations. Unlike average calculation, where a single wrong IP can significantly deflect a location, the median provides a better suited starting point, but does not guarantee good results if there is complete disagreement between geolocation databases. For example, Fig. 10 shows a single PoP in the UUNET network, which is located by different geolocation databases in six locations spread in 4 countries and two continents. However, since geolocation databases are typically reliable in country-level assignment, such examples are rare.

Location error range. Every PoP location is assigned a range of convergence, representing the expected location error range based on the information received from the geolocation databases. For every IP address in a PoP and for every geolocation database we collect the geographic coordinates. Thus if there are N IP addresses and M databases, for each of the IP addresses we get at most (if all are resolved) $N \times M$ location votes. The algorithm finds the smallest radius which has at least 50% of the votes, with 1 km granularity. If the radius is above a given threshold, typically 100 km or 500 km, the algorithm outputs the threshold radius and the percentage of location votes within it. If one of the geolocation databases lacks information on an IP address, this IP element is not counted in the majority vote.

Location refinement. After a range of convergence is found, the PoP location accuracy is further improved. The new PoP location is set to the median of the location votes inside the range of convergence. This ensures that deviations in the PoP location caused by a small number of IP elements outside the range of convergence are discarded, and the PoP is centered based only on credible IP addresses locations.

To summarize, the PoP geolocation algorithm provides per PoP longitude, latitude, range of convergence and the percentage of location votes within its convergence range.

4.1. Geolocation results

The geolocation algorithm has two interesting outcomes. First, it validates the PoP extraction algorithm by showing that PoPs are indeed scattered geographically, and locates points of presence around the globe. Second, it examines the quality of the geolocation services and finds their faults.

The algorithm converges successfully based on its validation's results. 70% percent of the PoPs have a range of convergence of ten kilometer or less. Although 89% of the PoPs have more than the minimal requirement of 50% of the IP location votes within the convergence range, for only 9.1% of the PoPs have over 90% of the location votes within the convergence range, indicating inaccuracies in some of them. To strengthen this point, when requiring the PoP location to be agreed upon by any three geolocation

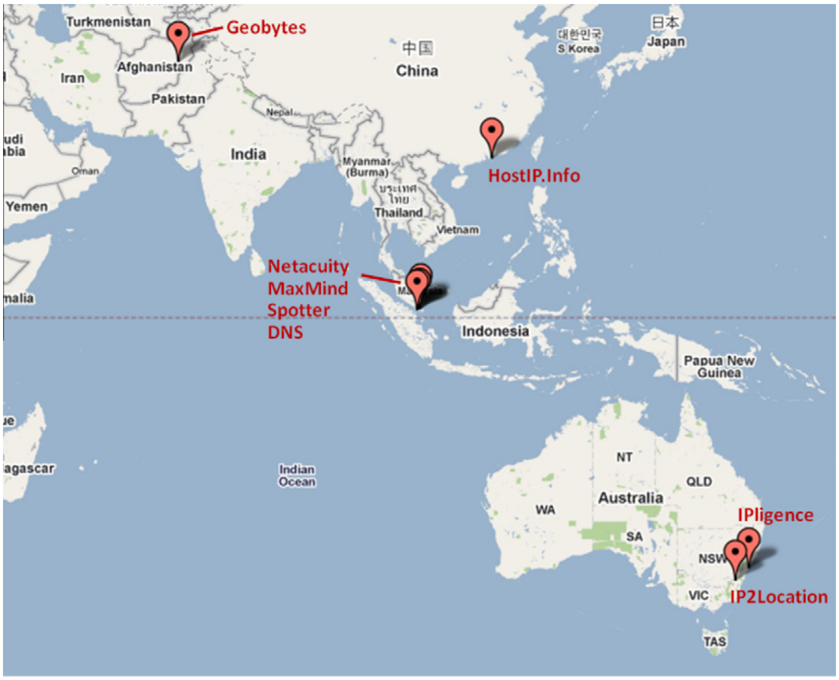


Fig. 10. Mismatch between databases – UUNET.

databases instead of five, over 90% of the PoPs converge within ten kilometers range, which comes to show that the disagreement between the geolocation database is the cause to the above.

Fig. 11 shows the discovered PoPs located on a world map. Clearly, the US and Europe have very good coverage. In East Asia many PoPs are discovered as well, but only a few are found in South America and Africa.

We then proceed and generate a PoP location map per Internet service provider. The maps display the PoPs of all the ASes residing under the same provider (sibling ASes), to provide a full picture of the vendor's network. The provider maps also show the connectivity between the different PoPs, as measured by DIMES. Fig. 12 shows as an example provider map of Qwest with its internal network connectivity.

To validate our generated maps we compare them against the PoP maps published by the ISP, such as Sprint [29], Qwest [30], Global Crossing [31], British Telecom [32], ATT [33] and others. The PoP algorithm detects most of the large points of presence, but it detects very few small, local PoPs. There are several explanations for this behavior. First, we measure mainly to and through nodes that pass a lot of traffic, and filter out edges that were hardly measured, in order to filter out noise. Even when we add the PoP IPs discovered by iPlane, most of these small PoPs are still not found. This leads us to the second reason some PoPs are not discovered: due to security reasons, many routers do not answer traceroute ICMP packets, which reduces the algorithm's ability to discover the PoP structure. Last, some of the vendors employ encapsulating protocols such as MPLS, which may hide most of the routing path. Luckily, as our results show, these protocols are not deployed widely enough to harm our measurements.

As another method of validation, fifty PoPs that belong to universities around the globe were selected, and the location given to them by the algorithm was compared against the institute's actual location. For 49 out of 50 universities, the location was accurate within a 10 km radius. The last PoP, belonging to The University of Pisa, was located by the algorithm in Rome instead (330 km away), due to an inaccuracy in the MaxMind and IPligence databases. Only Hostip.info provided the right coordinates for this PoP. Each PoP location was also validated against its DNS name, yet many interfaces had no DNS name assigned to them.

We compare our PoP geolocation also for GARR, the Italian research network. In weeks 42–43, 2010 we found eight PoPs in GARR, containing 99 IP addresses. GARR has a total of fifty-eight PoPs in Italy; however in several cases a few PoPs are located in a small area. For example, there are eight PoPs in Milan's area, and six in Florence's vicinity. Our extraction algorithm thus merges such PoPs into a single entity. Checking the assignment of PoPs to locations, based on DNS, information provided on GARR's website [34] and information from users, we successfully geolocate five of the PoPs in their correct location based on 100% of the IP locations. In two PoPs, the PoP is located correctly, however it seems to include a single IP address which is supposed to reside in a different location. In both cases we observe that the edge delay to other IP addresses included in this PoP is less than 2mS. For the last PoP, the PoP is located correctly in Milan, however it includes several IP addresses that are supposedly part of different PoPs. We note that the geolocation databases are also missing information for many of these IP addresses – only 55% of the IPs which are part of the PoP have location information, and the agreement level that we assign for the PoP is low as well: 66%.

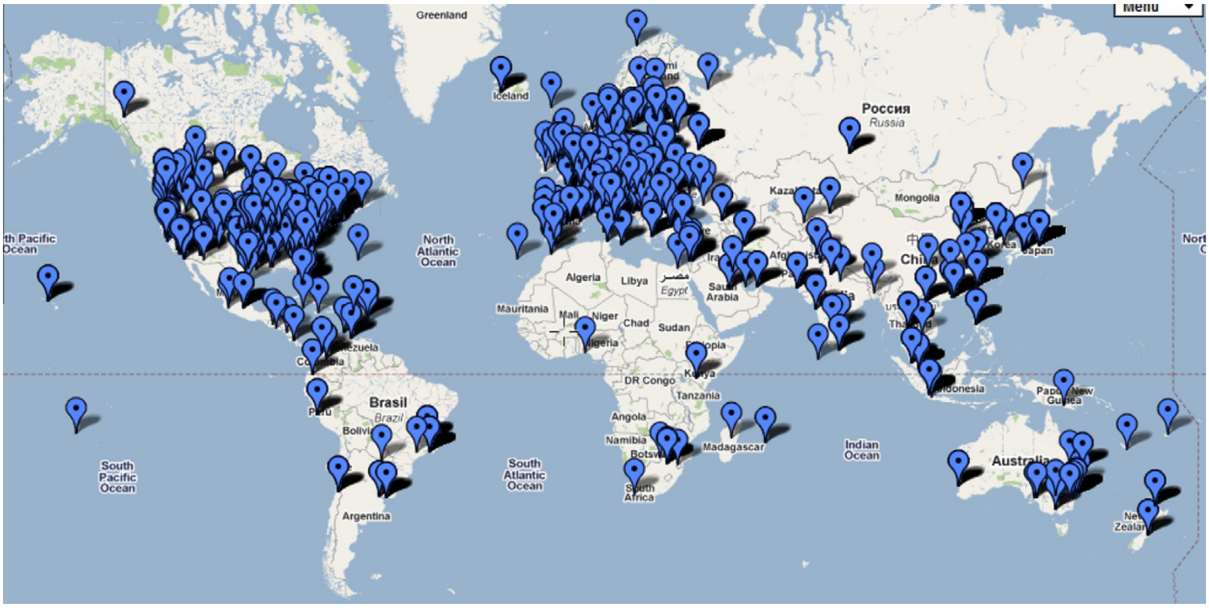


Fig. 11. PoPs world map.

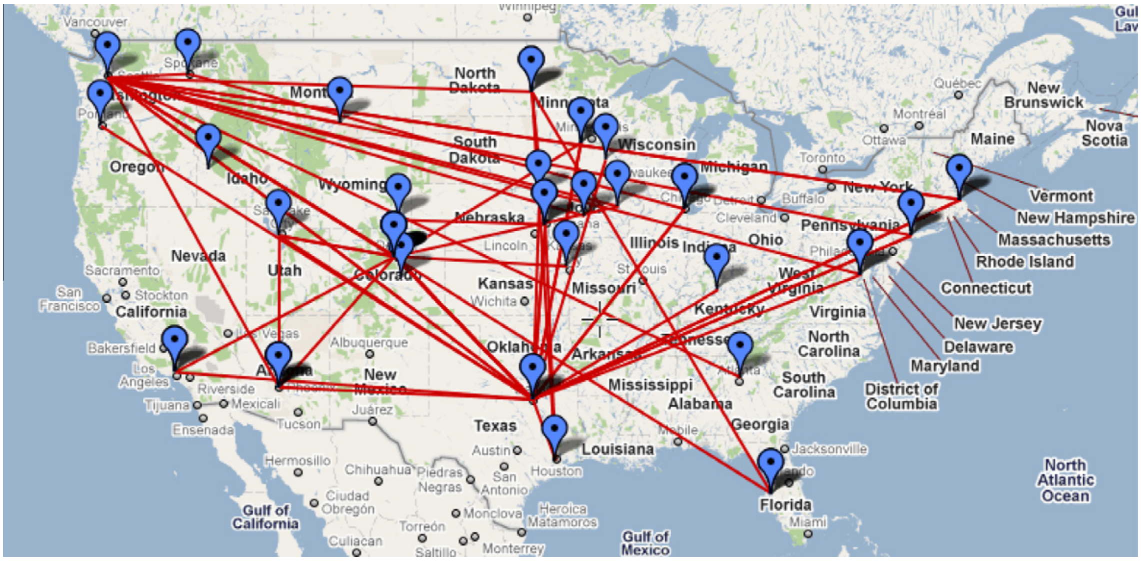


Fig. 12. QWEST US PoP map.

For less than 10% of the PoPs we fail to find the location with high confidence using five geolocation databases. In almost all these PoPs the cause is lack of location information in the databases, mostly in HostIP.Info, GeoBytes and MaxMind (MaxMind provides country level information). When a majority is requested only amongst three databases, more than 99% of the PoPs are located with high confidence. When IP location information is available, the main cause of PoP location failure is due to disagreement between the location services. To summarize, while in some cases the disagreement is a result of incorrectly estimated links, as suggested in subsection 3.1, the majority is caused by geolocation database inaccuracies.

5. Discussion

5.1. Issues in PoPs discovery

The extraction of PoPs and assignment to geolocation based on active measurements requires careful data filtering. Previous [3,4,11] PoP discovery algorithms were based on methods such as RTT measurements, Interface aliasing, and DNS entries; all three are known to inflict errors. In particular, the delay measurement inaccuracy is a known problem [10,35], and clustering by the delay from a limited number of vantage points is prone to errors in distinguishing short distances. Internet aliasing to routers was shown to be problematic, as well as the use of DNS [36].

Our PoPs extraction algorithm takes several precautions. First, at least PM_{min_th} measurements are required per IP level edge in order for it to be considered by the PoP extraction algorithm, and a median algorithm [19] is applied in order to reduce the delay measurement error. Second, the distribution of the DIMES vantage points results in the measurement of an IP edge being made by different agents from different locations, thus reducing the inherited measurement error of a specific path. Last, when DIMES measures a certain path, it sends four consequent traceroutes per destination. We considered the median of both, the average of two middle delay results time measured and minimum delay, across all edges and studied the tradeoffs between the two. Fig. 13 shows a CDF of median edge delay, based on best (least) and average traceroute measurements, over one million edges. As can be seen, both graphs follow the same trend, with about 1mS shift between the two plots at the small delay values (e.g., the probability of getting 2mS delay using average delay measurements equals the probability to get 1mS delay using best delay measurements). Looking at an edge delay of 3mS, the value set for the PD_{max_th} threshold in our evaluation, the best (least) delay CDF probability is 0.43, while the average delay CDF probability is close to 0.36. As there may be a variance between networks, we compare the edge delay of five service providers: ATT, Sprint, Cogent, Level3 and France Telecom. Fig. 14 shows for each of the

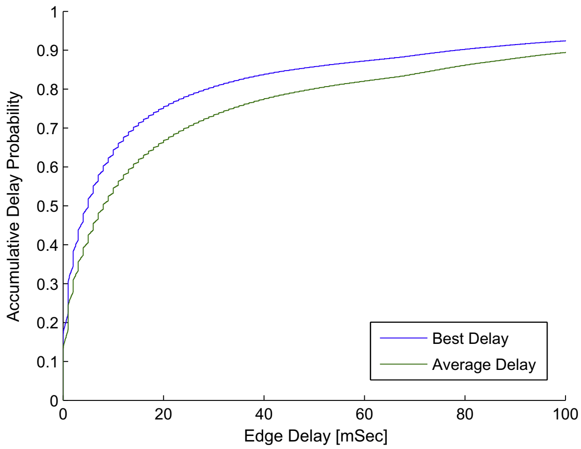


Fig. 13. CDF of best and average edge delays, one million measurements.

providers the CDF of best (least) and average edge delay. As can be seen, the best edge delay curves (top) overlap for all ISPs, and the same applies for the average edge delay (bottom). We thus take the best time per quartet of measurements for our edge delay calculations.

5.2. Geo-location results

Validating the geolocation results is problematic [22] due to the need for ground truth which is hard to obtain. Our validation is based on two methods. First we point to coherence in the data from multiple databases. If the radius of convergence between five different databases for a large majority of the PoPs is small, it is strong evidence for the validity of the results. The advantage of our geolocation method is that the returned location comes with a radius of convergence which serves as a confidence measure. In the future, we plan to use an iterative algorithm that will start by locating the PoPs with the highest confidence values and then based on triangulation (using the PoP to PoP delay estimations) will continue to locate PoPs with decreasing confidence. The second validation we used is by comparing our results to data available on the Web by service providers. Some ISPs provided feedback on the PoP maps as well. Overall, we believe our validation shows a high confidence in the results, but of course we do not claim of 100% accuracy.

5.3. Leveraging PoPs for network properties study

PoP-level maps can be used in diverse ways to study the Internet. Beyond providing geographical information on service providers' equipment spread, additional information can be obtained on the connectivity within the AS network, and more importantly, the connectivity between service providers. While most of the studies until today focused on types of relationships (ToR) between service providers on the AS level, a study of ToR on the PoP level can provide much more information, such as how ToR between a pair of ISP changes between locations over the globe. This will help us understand routing in the Internet.

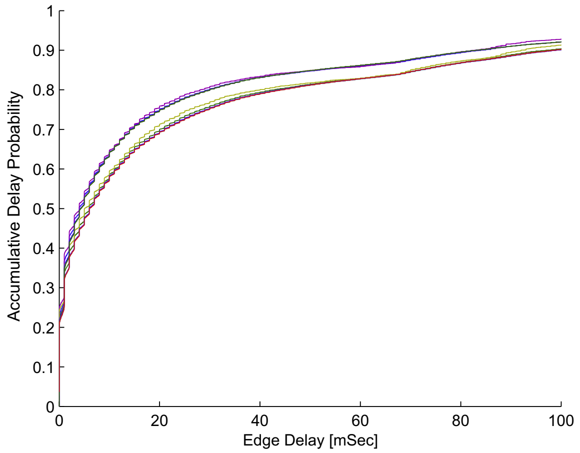


Fig. 14. CDF of best and average edge delays, different ISPs.

Analyzing PoP level maps from geographic and demographic standpoints can be leveraged to design an evolution model of the network. An advance modeling framework may also take into account the combined PoP/AS level to create evolutionary models coupling various socio-economic datasets to the growth of the Internet capability.

Another application of PoP-level maps is evaluation of geolocation databases. The fact that a PoP groups IPs with a locality property allows to check consistency within the database. Another option is to check the spread radius of IPs within the same PoP according to a single database and to compare different databases' range of convergence. By placing PoPs on a map according to different geolocation databases, it is also possible to find anomalies in the database. We discuss this topic thoroughly in [22].

The PoP level maps, as well as source measurements and derived tables are all available for the research community from the DIMES Web site at www.netDimes.org.

6. Conclusion

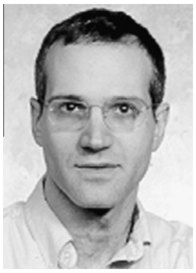
In this paper we presented a novel structural approach to automatically generate world-wide PoP maps using the DIMES project infrastructure. The extraction algorithm is based on detection of a network motif, and we discuss at length the theoretical background supporting this scheme. The generated PoP maps have location information for each PoP, deduced from geolocation databases and using a geolocation algorithm which increases the PoP location accuracy. An extensive validation of both PoPs extraction and geolocation algorithms is provided, studying different aspects of the approach. We recognize that many PoPs, mainly small ones, are not discovered due to insufficient measurements. To make the map richer we believe one should improve DIMES's spread, adding more vantage points and increasing the number of measurements. The generated PoP maps can be used for purposes such as the study of type of relationships (ToR) between service providers on PoP level, geolocation databases evaluation [22], distance estimation, and more.

References

[1] Y. Shavitt, E. Shir, DIMES: Let the Internet measure itself., in: ACM SIGCOMM Computer Communication Review, vol. 35, 2005.
[2] D.G. Andersen, N. Feamster, S. Bauer, H. Balakrishnan, Topology inference from BGP routing dynamics, in: Internet Measurement Workshop, 2002, pp. 243–248.
[3] N. Spring, R. Mahajan, D. Wetherall, Measuring ISP topologies with Rocketfuel, in: ACM SIGCOMM, 2002, pp. 133–145.
[4] H.V. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, A. Venkataramani, A structural approach to latency prediction, in: IMC'06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, 2006, pp. 99–104.
[5] Quova, 2010, <http://www.quova.com>.
[6] Digital Envoy, NetAcuity Edge, , 2010, <http://www.digital-element.com/our_technology/edge.html>.
[7] V.N. Padmanabhan, L. Subramanian, An investigation of geographic mapping techniques for Internet hosts, in: SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, 2001, pp. 173–185.
[8] B. Gueye, A. Ziviani, M. Crovella, S. Fdida, Constraint-based geolocation of Internet hosts, IEEE/ACM Trans. Netw. 14 (6).
[9] B. Wong, I. Stoyanov, E.G. Sirer, Octant: A comprehensive framework for the geolocalization of Internet hosts, in: NSDI, 2007.

[10] E. Katz-Bassett, J.P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, Y. Chawathe, Towards IP geolocation using delay and topology measurements, in: The 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06), 2006, pp. 71–84.
[11] K. Yoshida, Y. Kikuchi, M. Yamamoto, Y. Fujii, K. Nagami, I. Nakagawa, H. Esaki, Inferring PoP-level ISP topology through end-to-end delay measurement., in: PAM, vol. 5448, 2009, pp. 35–44.
[12] S. Laki, P. Mátray, P. Hága, I. Csabai, G. Vattay, A model based approach for improving router geolocation, Computer Networks 54 (9) (2010) 1490–1501.
[13] B. Eriksson, P. Barford, J. Sommers, R. Nowak, A learning-based approach for IP geolocation, in: Passive and Active Measurement, 2010, pp. 171–180.
[14] A. Sardella, Building next-gen points of presence, cost-effective PoP consolidation with juniper routers, White paper, Juniper Networks, June 2006.
[15] B.R. Greene, P. Smith, Cisco ISP Essentials, Cisco Press, 2002.
[16] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Network motifs: simple building blocks of complex networks, Science 298 (5594) (2002) 824–827.
[17] Mfinder – network motifs detection tools, <http://www.weizmann.ac.il/mcb/UriAlon/>.
[18] D. Feldman, Y. Shavitt, Automatic large scale generation of Internet PoP level maps, in: GLOBECOM, 2008, pp. 2426–2431.
[19] D. Feldman, Y. Shavitt, An optimal median calculation algorithm for estimating Internet link delays from active measurements, in: IEEE E2EMON, 2007.
[20] DIMES, Distributed Internet Measurements and Simulations, <http://www.netdimes.org/>.
[21] MaxMind LLC, GeoIP, 2010, <http://www.maxmind.com>.
[22] Y. Shavitt, N. Zilberman, A geolocation databases study, IEEE Journal on Selected Areas in Communications 29 (9).
[23] IPligence, IPligence Max, 2010, <http://www.ipligence.com>.
[24] hostip.info, hostip.info, 2010, <http://www.hostip.info>.
[25] Hexsoft Development, IP2Location, 2010, <http://www.ip2location.com>.
[26] Geobytes, GeoNetMap, 2010, <http://www.geobytes.com/>.
[27] S. Laki, P. Mátray, P. Hága, T. Sebök, I. Csabai, G. Vattay, Spotter: A model based active geolocation service, in: IEEE INFOCOM 2011, Shanghai, China, 2011.
[28] Y. Shavitt, N. Zilberman, A structural approach for PoP geolocation, in: Infocom Workshop on Network Science for Communications (NetSciCom), 2010.
[29] Sprint, Global IP network, <https://www.sprint.net/network_maps.php>.
[30] Qwest, IP network statistics, <http://66.77.32.148/index_flash.html>.
[31] Global Crossing, Global Crossing network, http://www.globalcrossing.com/html/map062408.html.
[32] BT Global Services, Network maps, http://www.bt.net/info/europe.shtml.
[33] AT& T Global Services, AT& T Global Services global network map, http://www.corp.att.com/globalnetworking/media/network_map.swf.
[34] GARR, The Italian academic and research network, http://www.garr.it/eng/index.php.
[35] D. Lee, K. Jang, C. Lee, S. Moon, G. Iannaccone, Path stitching: Internet-wide path and delay estimation from existing measurements, in: IEEE Infocom mini-conference, 2010.
[36] M. Zhang, Y. Ruan, V. Pai, J. Rexford, How DNS misnaming distorts Internet topology mapping, in: ATEC '06: Proceedings of the annual conference on USENIX '06 Annual Technical Conference, 2006, pp. 34–34.

Dima Feldman received his B.Sc. in Electrical Engineering from the Technion-Israel Institute of Technology, Haifa, Israel in 2000, and his and M.Sc. in Electrical Engineering from Tel-Aviv University, in 2007. His re- search focused on Internet measurements,mapping and characterization. He currently serves in a managerial position in the telecommunications industry.



Yuval Shavitt received the B.Sc. in Computer Engineering (cum laude), M.Sc. in Electrical Engineering and D.Sc. from the Technion-Israel Institute of Technology, Haifa, Israel in 1986, 1992, and 1996, respectively. After graduation he spent a year as a Postdoctoral Fellow at the Department of Computer Science at Johns Hopkins University, Baltimore, MD. Between 1997 and 2001 he was a Member of Technical Stuff at Bell Labs, Lucent Tech- nologies, Holmdel, NJ. Starting October 2000, he is a Faculty Member in the School of Electrical Engineering at Tel-Aviv University, Israel. His research interests include Internet measurements, mapping, and characterization; and data mining peer-to-peer networks.

Noa Zilberman received her B.Sc. and M.Sc. (both magna cum laude) in Electrical Engineering from Tel-Aviv University, Israel in 2003 and 2007, respectively. Since 1999 she has filled several development, architecture and managerial roles in the telecommunications industry. She is currently a Ph.D. candidate in the School of Electrical Engineering at Tel-Aviv University. Her research focuses on Internet measurements, mapping, and characterization.