A Structural Approach for PoP Geo-Location

Yuval Shavitt
School of Electrical Engineering
Tel-Aviv University, Israel
Email: shavitt@eng.tau.ac.il

Noa Zilberman
School of Electrical Engineering
Tel-Aviv University, Israel
Email: noa@eng.tau.ac.il

***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 then are assigned a location using collaborated information from several geo-location databases. Using this approach, we could evaluate the accuracy of these databases and suggest means to improve it. The PoP-PoP edges, which are extracted from the traceroutes, present a fairly rich AS-AS connectivity map.**

I. INTRODUCTION

A. Internet Maps

Mapping the Internet and study its evolution have become an important research topic. Internet maps are presented in several layers: 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. 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 is that AS sizes may be different 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. 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 Point of Presence (PoP). 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, not only by its amount of routers and connectivity, but also by the number of physical co-locations of the network, which is an important contribution. Using PoP level graphs one can detect important nodes of the network, understand network dynamics and more.

This paper focuses on PoP level map generation, based on an algorithm described in Section II. 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.

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

In the recent years several works have tried to cope with the question of location accuracy. Octant [5] uses a geometric approach to localize a node within 22 mile radii. Katz *et al.* [6] suggested using link delay to improve the location of nodes. Yoshida *et al.* [7] used end-to-end communication delay measurements to infer PoP level topology between thirteen cities in Japan.

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. We also explore the generated maps properties and provide a comparison between several commercial location services. Last, we provide some data on our global AS Geo-PoP connectivity map.

II. POP DISCOVERY

A. 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 [8], [9] 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 core routers are connected to the core network of the ISP. The algorithm we use for PoP extraction was first suggested by Feldman and Shavitt in [10]. The algorithm looks for bi-partite subgraphs with certain weight constraints in the IP interface graph of an AS; no aliasing to routers is needed. The bi-partites serve as cores of the PoPs and are extended with other close by interfaces. Feldman and Shavitt suggested the following steps to reduce the IP level graph G(V,E) to a PoP level network:

[†]This work was partially funded by the OneLab II and the MOMENT consortia that are part of the EU FP7; The Israeli Science Foundation's center of knowledge grant 1685/07; and 2008 IBM Faculty Award.

Initial Partition. Remove all edges with delay higher than PD_{max_th}, PoP maximal diameter threshold, and edges with number of measurements below PM_{min_th}, PoP measurements threshold. PM_{min_th} is introduced in order to consider only links with a high 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. Induced subgraph is one that consists of some of the vertices of the original graph and all of the edges that connect them in the original graph. 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. The first and most obvious reason is geographically adjacent PoPs, e.g., New York, NY and Newark, NJ. The other is caused by wrong delay estimation of a small amount 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. Next, the algorithm checks if each connected group has more than one PoP. Note that each candidate partition looks like a collection of highly connected bipartite graphs with rich connectivity between them. The entire partition to parents and children is then divided according to the measurement direction in the bipartite graph(each node or groups of nodes simultaneously can be parents of one bipartite and children 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.

(c) Unification. Unifying *parent/child* group to the same PoP. If *parent pair* and *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 the pair of groups is declared as part of the same PoP.

Final Refinements.

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 2x2 bi-partite, in the extreme case only a single link connects two parts of a PoP. This will not allow the unification process, which is 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 shrank considerably, and thus the search for 'close' components is easy.

In the original algorithm [10], an additional step was implemented, called Singleton Treatment, in which nodes with only one or two links are assigned to PoPs. This part was eliminated in this work, as it degrades algorithm stability and does not add to the number of discovered PoPs.

Compared Time Frame	#PoPs	#IPs in PoPs	#Distinct Edges
1 Week to 1 Week	$< 1\%$	$< 1\%$	$\pm 20\%$
1 Week to 2 Weeks	$+58\%$	$+79\%$	$+43\%$
2 Weeks to 4 Weeks	$+10\%$	$+15\%$	$+59\%$

TABLE I
CHANGES IN POP MAPS BETWEEN DIFFERENT TIME FRAMES

B. PoP Extraction Validation and Results

Feldman and Shavitt [10] made only a limited attempt to validate the algorithm output. Here we present our validation tests and the results of a full implementation. The dataset in use is 2009 measurements, with a focus on weeks 27 to 30 for specific examples.

First, two weeks were selected as the best time period for collecting measurement for PoPs. DIMES produces over 5 million daily measurements, meaning thirty to forty million measurements per week, which typically result in 5.5M to 6.5M distinct IP edges being discovered. Approximately 1000 agents in about 200 ASes are used for the measurements, targeting 2.5M destination IPs in over twenty six thousand ASes. 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 less distinct edges than 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 set. Table I 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 accordingly over the compared periods. "#Distinct Edges" refers to the change in distinct edges measured by DIMES. This number is independent from 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 run the median algorithm described in [10] to find the delay between two adjacent nodes.

The resulting IP address to PoP mapping table typically consists of 50K IP addresses, in about 4400 different PoPs. The average size of a PoP is 15, 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 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 be 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 belongs to the DIMES default experiment where a set of roughly 1 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. This increases the measurement cycle to over 2 days. The addition of the iPlane IP addresses increased the number of PoP 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 user IPs.

The number of PoPs found in an AS network correlates with its measured size. Figure 1 shows that the number of PoPs discovered per AS depends logarithmically on the number of IP edges measured. Figure 2, showing the number of IPs included in PoPs 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.

Figures 3 to 6 explore the PoP extraction algorithm's sensitivity to PD_{max_th} and PM_{min_th}. In each figure 5 ISPs are explored: Level 3, AT&T, Comcast, MCI, and Deutsche Telekom. In Figure 3 the number of discovered PoPs is compared with PD_{max_th}, the maximal delay threshold. Figure 4 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 $3mSec$ or above. PD_{max_th} was selected to be $5mSec$, as it presents a good tradeoff between delay measurement's error and location accuracy. Figures 5 and 6 show the effect of PM_{min_th}, the minimal number of measurements threshold, 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. It is caused by a loss of connectivity inside a PoP which cause it to split to several PoPs located at the same place. 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 of 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 itself 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 DIMES 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 II-C. 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.

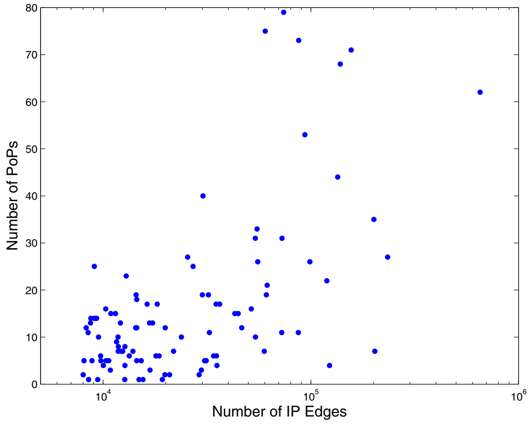

Fig. 1. Number of Discovered PoPs vs. Number of measured IP Edges

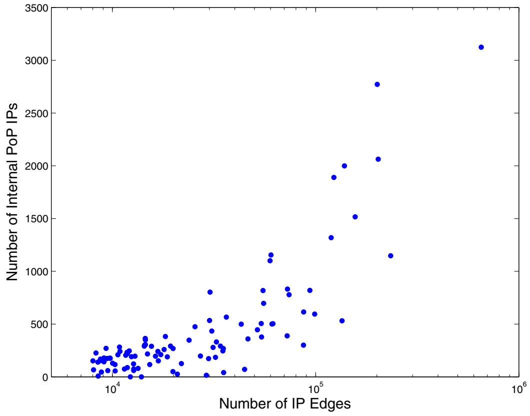

Fig. 2. Number of IPs in PoPs vs. Number of measured IP Edges

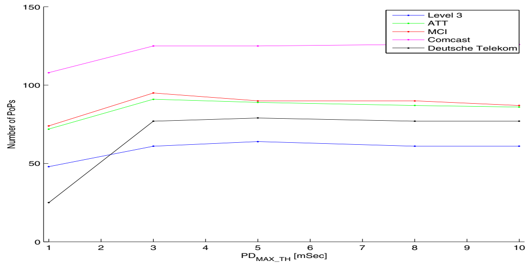

Fig. 3. Number of PoPs vs. Maximal Delay

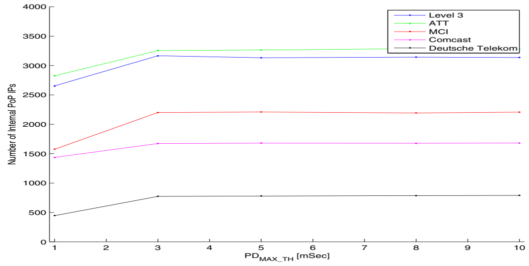

Fig. 4. Number of IPs in PoPs vs. Maximal Delay

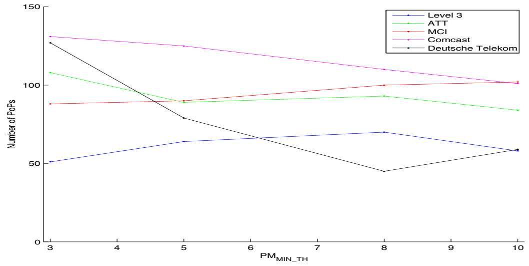

Fig. 5. Number of PoPs vs. Minimal Number of Measurements

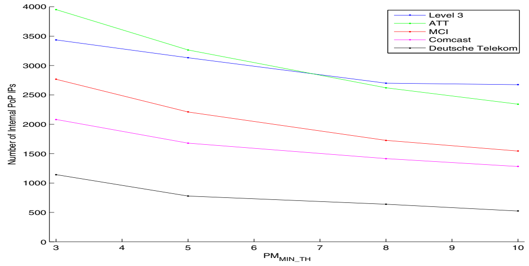

Fig. 6. Number of IPs in PoPs vs. Minimal Number of Measurements

C. 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 *et al.* [6] 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 the recent years geolocation services are being 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 to a great extent of accuracy. While most location service do not reveal their level of accuracy, country-level assignment is typically over 99% accurate, as the IP assignments to ASN are in most cases bounded within a single country. MaxMind GeoIP service[11] provides accuracy information on city level, within a radius of 25 miles of true location, which ranges from 59% (United Kingdom) to 93% (Malta, Singapore) in industrial countries. Unites states, for example, has 83% accuracy on the city level. A further assessment of the geolocation information is therefore required.

We use 3 geolocation services to maximize the accuracy of our PoP location: MaxMind GeoIP [11], Ipligence [12] and Hostip.info [13].

The PoP location algorithm attempts to locate a point of presence based on the best known geographic location of each of the IPs included in it. First, each of the three geolocation databases is queried for the location (longitude, latitude) of each IP included in the PoP. Next, the center weight of the PoP location is found by calculating the median of all PoP's IP locations. Unlike average, where a single wrong IP can significantly deflect a location, median provides a better suited starting point. On its own, median is certainly not enough. If there is complete disagreement between geolocation databases as for the location of a PoP, e.g., if one of them places all the PoP IPs in London, and the other in New-York, a median may be far away from the real distance. However, since geolocation databases are very reliable in country-level assignment, such an example is highly unlikely.

Next, we look for the PoP location a range of convergence. We begin at the median location, and check if there is a majority vote for the PoP location within a radius 0.01 degrees (one latitude/longitude degree is roughly equivalent to 111km). If the circle includes less than 50% of the located IPs, we continue and increase the radius of the circle, by 0.01 degrees each step, until the PoP location has a majority vote, or alternatively, the circle radius reaches 1 degree, which we define as the maximal range of error. With a majority vote we ensure that at least two, if not more, of the geolocation databases agree on the PoP location. The PoP location accuracy is then improved by finding the PoP location center of weight based only on the geolocation of IPs within the range of convergence.

Our algorithm also finds a second range of convergence, which seeks only minimal majority - meaning that more than 33% of the location votes are within the range, in which case there is for sure some agreement between the databases, but not necessarily more than 50%. This comes in handy when two of the databases lack information on some of the IPs. Although the above algorithm seems to be simplistic, we show in the results that it achieves exceptionally good accuracy.

D. 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

Fig. 7. PoPs World Map

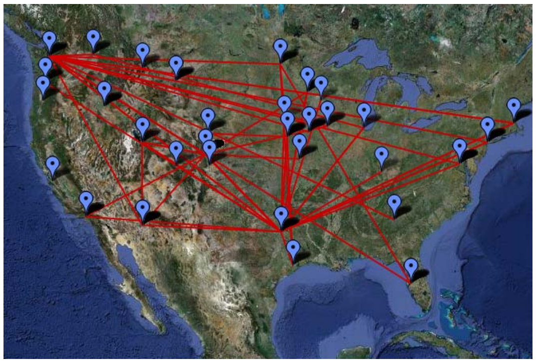

Fig. 8. QWEST US PoP Map

of presence around the globe. Second, it examines the quality of the geolocation services and finds their pitfalls.

Based on the algorithm results it converges successfully. 84% percent of the PoPs have a range of convergence of 0.01 degrees (equivalent to one kilometer) and 78% of the nodes have a level of agreement above 50%, yet for only 28% of the nodes there is over 90% agreement between the different location services, indicating inaccuracies in some of them. We will address this issue later in this section.

Figure 7 shows the discovered PoPs located on the world map. Clearly, US and Europe have a 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 PoP location map per Internet service provider. The maps display the PoPs of all the AS residing under the same provider, 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. Figure 8 show 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 [14], Qwest [15], Global Crossing [16], British Telecom [17], ATT [18] 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 to 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 hide most of the routing path. Luckily, as our results show, these protocols are not deployed vastly 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 kilometers radius. The last PoP, belonging to The University of Pisa, was located by the algorithm in Rome instead, due to an inaccuracy in 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.

For less than 5% of the PoPs we fail to find the location with high confidence. About a third of these cases have only a single source database with location information, but less than 1% have no location information in any of the location databases. The rest of the failures are due to disagreement between the location services or lack of data in the location service databases. While in some cases the disagreement is a result of incorrectly estimated links, as suggested in II-A, the majority is caused by geolocation databases inaccuracy.

This brings us to our second objective: assessing the quality of the location services. We have discovered several problems in the location information provided by Hostip.info, MaxMind and IPligence. To fairly compare between the three, a separate PoP location table was generated for each. The first problem experienced was a lack of location information for some of the PoPs. This problem was especially evident with the Hostip.info database, where almost a third of the PoPs and over 28% of the nodes had no location information. For MaxMind, the equivalent results were 10% of the PoPs and almost 12% of the IPs. IPligence has no location information on 6.5% of the IPs and 1% of the PoPs. Using our algorithm information can therefore help retrieve location of unlocated IPs.

While the geolocation services information is very accurate on country level, e.g., simply based on the regional internet registry database, the information is not always known on city level. For this reason, some of the location services, such as MaxMind, use the country information when city information is not available. Unlike MaxMind, which gives a table with the default location provided when city is unknown (being the middle of the country), for IPligence it is not public knowledge what information is provided in such cases. Supposedly, it should return as Null, though evidence show that they do too have some default returned location. For example, for Qwest, 73 different PoPs were discovered by our algorithm. Maxmind located them in 55 different locations, Hostip.info in 46 different locations, and IPligence returned only 4 distinct locations for the IPs included in this group. 70 of these PoPs where located by IPligence in Denver, where Qwest headquarters are located.

In some cases, there is a contradiction between different location services. For example, in one case IPligence pointed out that all the IPs within a certain PoP are located in Los-Angeles, while according to Maxmind all the PoP's IPs were located in Washington, DC.

It should be noted that MaxMind have informed us that their quality information is in regards to end user nodes, and they do not claim to have accurate information on router IP location. In addition, inaccurate location information found in their May-2009 database was fixed in their July-2009 update. A similar correspondence with IPligence did not bear fruit.

III. AS GEO-POP CONNECTIVITY MAPS

The PoP location maps that we generate can be used to examine the connections between different ASes. Taking the top 20 ASes (by DIMES measured AS degree) can show where each ISP connects to another ISP, and more important, to how many different ISP each discovered PoP connects.

On the average, each PoP connects to over 50 different ASes, with a median of 22 different AS connections. We count as a connection only a direct edge between two discovered PoPs belonging to different ASes. Table II shows, for selected large ASes, the number of distinct edges measured by DIMES over a fortnight, the number of discovered PoPs, and the number of inter-AS connections. Note that if a single PoP connects to a different AS with more than a single IP edge, it is counted only once, but if different PoPs connect to the same AS, each connection is counted. Every discovered PoP was connected to at least one PoP in a different AS, namely, we did not discover any PoP connected only within their AS. These results reflect the conclusions we discussed above, namely that our algorithm tends to discover mainly large PoPs, which are likely to be located in an interchange point between different ASes. We identify that over half of the PoP connections are to small, local AS, while most of the other connections are to large scale ISP. Only a small number of the connections are between different ASes owned by the same ISP.

IV. CONCLUSION

In this paper we presented a novel structural approach to automatically generate world-wide PoP maps using the DIMES project infrastructure. The PoP maps have location information for each PoP, deduced from geolocation databases. We explored pitfalls originating from using geolocation databases and pointed out several ways by which using PoP maps can improve their accuracy. In addition, the connectivity between different ASes was considered based on the generated PoP maps and DIMES measurements. 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 spread, as well as our algorithm. However, it might also require joining our results with other techniques used in the past such as iPlane's [4].

ISP	**IP Edges**	**No. of PoPs**	**Inter-AS PoP connections**
Level 3	1307406	62	13011
ATT	536264	83	4904
TaliaNet	492072	27	6204
Cogent	402106	35	5398
MCI	304456	79	4411
Comcast	287848	75	3358
China Backbone	276061	124	2015
Global Crossing	269554	44	6317
Korea Telecom	246152	4	327
NTT America	245430	26	2779
Sprint	242682	22	2819
Qwest	178490	73	4574
Tata Communications	162700	44	3178
KDDI	160356	31	615
Deutsche Telekom	147878	79	3371
Net Access Corp.	123138	75	3358

TABLE II
POP INTER-AS CONNECTIVITY

For the geo-location, we intend to use high confidence PoP location with high confidence delay information to better select the location of low confidence PoPs. This way one can propagate the geo-location along the map and get a better placement of the PoPs.

REFERENCES

[1] Y. Shavitt and E. Shir, "DIMES: Let the internet measure itself." in *ACM SIGCOMM Computer Communication Review*, vol. 35, Oct. 2005.
[2] D. G. Andersen, N. Feamster, S. Bauer, and H. Balakrishnan, "Topology inference from BGP routing dynamics," in *Internet Measurement Workshop*, 2002, pp. 243–248.
[3] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," in *In Proc. ACM SIGCOMM*, 2002, pp. 133–145.
[4] H. V. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, and 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] B. Wong, I. Stoyanov, and E. G. Sirer, "Octant: A comprehensive framework for the geolocalization of internet hosts," in *NSDI*, 2007.
[6] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and 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.
[7] K. Yoshida, Y. Kikuchi, M. Yamamoto, Y. Fujii, K. Nagami, I. Nakagawa, and H. Esaki, "Inferring PoP-level ISP topology through end-to-end delay measurement." in *PAM*, ser. Lecture Notes in Computer Science, vol. 5448. Springer, 2009, pp. 35–44.
[8] A. Sardella, "Building next-gen points of presence, cost-effective pop consolidation with juniper routers," Juniper Networks, White Paper, June 2006.
[9] B. R. Greene and P. Smith, *Cisco ISP Essentials*. Cisco Press, 2002.
[10] D. Feldman and Y. Shavitt, "Automatic large scale generation of internet PoP level maps," in *GLOBECOM*, 2008, pp. 2426–2431.
[11] MaxMind LLC, "GeoIP," http://www.maxmind.com, 2009.
[12] IPligence, "IPligence Max," http://www.ipligence.com, 2009.
[13] hostip.Info, "hostip.info," http://www.hostip.info, 2009.
[14] Sprint, "Global IP Network," https://www.sprint.net/network_maps.php.
[15] Qwest, "IP Network Statistics," http://66.77.32.148/index_flash.html.
[16] Global-Crossing, "Global Crossing Network," http://www.globalcrossing.com/html/ map062408.html.
[17] BT-Global-Services, "Network maps," http://www.bt.net/info/europe.shtml.
[18] ATT-Global-Services, "Att global services global network map," http://www.corp.att.com/globalnetworking/media/network_map.swf.