Internet Mapping: from Art to Science

Kimberly Claffy Young Hyun Ken Keys Marina Fomenkov
Dmitri Krioukov
Cooperative Association for Internet Data Analysis,
San Diego Supercomputer Center, University of California San Diego
{kc, youngh, kkeys, marina, dima}@caida.org

Abstract

We are designing, implementing, deploying, and operating a secure measurement platform capable of performing various types of Internet infrastructure measurements and assessments. We integrate state-of-the-art measurement and analysis capabilities to try to build a coherent view of Internet topology. In September 2007 we began to use this novel architecture to support ongoing global Internet topology measurement and mapping, and are now gathering the largest set of IP topology data for use by academic researchers. We are using the best available techniques for IP topology mapping, and are developing some new techniques, as well as supporting software for data analysis, topology generation, and interactive visualization of resulting large annotated graphs. This paper presents our current results, next steps, and future goals.

1. Introduction

We now critically depend on the Internet for our professional, personal, and political lives. This dependence has rapidly grown much stronger than our comprehension of its underlying structure, performance limits, dynamics, and evolution. Fundamental characteristics of the Internet are perpetually challenging to research and analyze, and we must admit we know little about what keeps the system stable. As a result, researchers and policymakers currently analyze what is literally a trillion-dollar ecosystem essentially in the dark, and agencies charged with infrastructure protection have little situational awareness regarding global dynamics and operational threats. To make matters worse, the few data points suggest a dire picture, shedding doubt on the Internet's ability to maintain and strengthen its role as the world's communications substrate.

Situational awareness and architectural innovation are faced with the same obstacle as empirical Internet science: radically distributed ownership of its constituent parts, and an operational climate that generally disincents sharing data with researchers. To better understand essential information infrastructure in light of these constraints, we began a jointly NSF- and DHS-funded project to address a small piece of the Internet awareness problem, by building an infrastructure and operating system platform to support large-scale active measurement studies of the global Internet. This paper presents our current results and discusses next steps and future goals.

Our approach integrates state-of-the-art strategic measurement and analysis capabilities into the most comprehensive and coherent view of Internet topology. We are building a secure measurement platform capable of performing several types of Internet infrastructure measurements and assessments. We are using this architecture to support our own continuous global Internet topology mapping, including improving inference of ownership of network devices and other challenges the research community has only heuristically solved. We are developing supporting software for data processing, analysis, annotation, topology generation, and interactive visualization of resulting large annotated graphs. We have also demonstrated the ability for this infrastructure to serve other macroscopic studies of the Internet, including a comparison of probing methods and an assessment of networks allowing IP address spoofing.

Sec. 2 describes our new *Archipelago* (Ark) architecture, Sec. 3 describes the current deployment, and Sec. 4 reviews our accomplishments over the last year in data collection and topology data analysis. Sec. 5 reviews other projects using the Ark infrastructure, and Sec. 6 offers forward-looking comments.

2. Archipelago measurement infrastructure

Archipelago (Ark) [21] is CAIDA's newest active measurement infrastructure, the next generation of the skitter-based active measurement infrastructure [13] that CAIDA operated for nearly a decade.[1] This section describes the three qualities and features that Ark strives to enable.

2.1. Easy development and rapid prototyping

Easy development and rapid prototyping are important factors, not only in increasing productivity, but in how they promote discovery. By lowering the cost in time and effort needed to implement a measurement idea, a researcher can explore more experimental and risky ideas (which may have a high return) and increase the sophistication of implemented techniques. These benefits will hopefully lead to better and more useful measurements.

Ark supports rapid prototyping by promoting software development at a high-level of abstraction using dynamic scripting languages and pre-built API's and services. Specifically, we adopt Ruby [3] as the primary implementation language for measurements and provide libraries tailored for topology measurements. For example, we provide a library for controlling all aspects of the *scamper* topology measurement tool from a Ruby script (Sec. 4.1 has details on scamper). By interacting with scamper over a network connection, a client can control and steer its measurements. In this arrangement, scamper acts as the general-purpose measurement engine, handling the details of efficiently performing parallelized traceroute and ping measurements (and eventually other types of measurements, such as alias resolution and exhaustive enumeration of load-balanced multipaths [9]), and a user's Ruby script acts as the brain, selecting targets, frequency, and kinds of measurements. Although the scripting approach is the preferred mode of development, Ark does not preclude low-level development work using languages like C or C++, or the direct execution of stand alone measurement tools. We also hope to provide a high-level API for direct packet generation, capture, and analysis, taking inspiration from efforts such as Scriptroute [27], Metasploit Framework [5], and Scapy [4].

2.2. Dynamic and coordinated measurements

At its simplest, a measurement infrastructure executes a pre-configured set of measurements to a static set of targets. However, many desirable measurements require dynamism and coordination among measurement nodes. For example, we may want to estimate path diversity within a given announced prefix, and we could find it out by using a set of monitors to probe the prefix in a binary-search pattern, continually subdividing the prefix until we no longer observe path diversity. As another example, we may want to monitor a set of target prefixes, e.g., containing some critical infrastructure, with low frequency pings and traceroute, and then trigger more comprehensive measurements from many vantage points upon detection of unreachability or path change (to detect prefix hijacking, for instance) [22].

A distinguishing feature of Ark is its focus on *coordination*. Coordination, broadly speaking, is concerned with planning, executing, and controlling an ensemble of distributed computations [18, 26]. Coordination is what allows the heterogeneous pieces of a measurement infrastructure to work efficiently toward a common task. To enable coordination, Ark employs a new implementation, called Marinda, of the *tuple-space* coordination model first introduced by D. Gelernter in his Linda coordination language [17, 14]. A tuple space is a distributed shared memory combined with a small number of easy-to-use operations. The tuple space stores tuples, which are arrays of simple values (strings and numbers), and clients retrieve tuples by pattern matching.

When acting as a communication channel, the tuple space supports one-to-one and many-to-many communication. Decentralized measurement processes execute autonomously at each monitor, communicating as needed, for example, to trigger further measurements or analyses based on locally observed events. Because the tuple space abstraction is easy to use, and because the implementation shields client software from the complexities of network communication and faults, Ark lowers the barrier to deploying sophisticated distributed measurements.

The tuple space also provides shared state, which allows for decoupling of measurement processes in time and space. That is, processes reading and writing to the tuple space can have non-overlapping lifetimes (decoupling in time) and need not know the identity, location, or even existence of each other—tuples are not addressed to a recipient (decoupling in space). These qualities allow dynamically changing, open-ended sets of participants over the course of each experiment and the ability to decompose a complex measurement task into phases (by storing intermediate results in the tuple space) or into a cooperating set of processes having distinct duties.

2.3. Measurement services

Another distinguishing feature of Ark is its support for measurement services. The goal is to make it easy for researchers to use and to build upon the work of others at the granularity of services. This approach has already taken hold on the Internet in the form of web services, using technologies like XML-RPC and SOAP, and in enterprise systems in the form of the service-oriented architecture (SOA).

[1] There are a number of well-regarded measurement platforms in use in the networking research community, including PlanetLab, iPlane, DIMES, and Scriptroute [27]. Each platform has distinguishing features and benefits, but no single platform is a replacement for all others–they are complementary. The purpose of Ark is to further enrich this ecosystem with new capabilities.

This support for services is made possible by the tuple space, which acts as the unified mechanism for transport and messaging, in the terminology of the web services protocol stack. More concretely, a user can easily deploy a measurement service by simply writing a program that interprets tuples as commands, performs some measurement, and returns the result as a tuple.

For example, we have implemented a traceroute and ping service that runs on each deployed monitor. With this service, a user connected to any node in the infrastructure can easily initiate ad-hoc, on-demand measurements from any local or remote monitor. The following Ruby code illustrates the simplicity of performing a ping measurement to `www.caida.org` and printing out the result:

```
ts.write ["PING", "192.172.226.123"]
result = ts.take ["PING-RESULT",
                  "192.172.226.123", nil]
puts result[3]
```

The same approach can be used to implement support services, such as to (1) map IP addresses to prefixes and ASes, (2) randomly generate a destination meeting some criteria, (3) check destinations against a system-wide no-probe list, and (4) choose a vantage point based on monitor attributes like location and capabilities.

This services architecture based on the tuple space has these advantages:

- **low deployment effort and cost**: no need to deploy a separate web server or additional hardware,
- **anyone can provide a service**: no special privileges or access to special areas required,
- **decentralized management**: no central system on which all services run and no central authorization,
- **ease of implementation**: little code is needed to implement a basic service, and developers are shielded from complexities of network programming,
- **ease of aggregation**: easy to write services that call other services, and
- **diverse communication patterns**: supports client-server, peer-to-peer, delegation, asynchronous exchanges, and other patterns.

3. Ark monitor deployment

Fig. 1 depicts the 31 active Ark monitors deployed as of mid-December 2008: 12 in North America, 2 in South America, 9 in Europe, 1 in Africa, 5 in Asia, and 2 in Australasia.

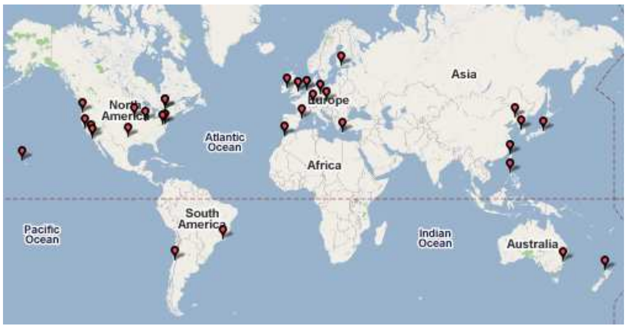

Figure 1. As of mid-Dec 2008, there are 31 Ark monitors in 20 countries.

We will continue to deploy Ark monitors at a rate of 1-2 monitors per month. Our goal is to deploy monitors in geographically diverse locations, so we can comprehensively sample the global Internet topology. In the next year we hope to deploy more monitors in underrepresented areas like South America and Africa. Another goal is to have diversity in the organizations hosting monitors. The majority of current monitors are currently deployed in academic/research organizations, but recently commercial ISPs are becoming more interested in participation.

Finally, we believe measurements of IPv6 adoption and performance will provide empirical data to inform policy as the exhaustion of the IPv4 address space approaches. We try to obtain IPv6 connectivity where available, and 6 deployed monitors have working IPv6 connectivity today. We expect 3 additional monitors to be IPv6-enabled in early 2009. We are currently implementing comprehensive ongoing IPv6 measurements, and hope to be in production in Jan 2009.

4. Internet measurements

4.1. Macroscopic IP topology

The Ark infrastructure supports CAIDA's Macroscopic Topology Project by systematically measuring IP-level paths to a dynamically generated list of IP addresses covering all /24 prefixes in routed IPv4 address space.

For scalability, resilience, and etiquette, we group monitors into teams and dynamically distribute the measurement tasks among team members. This parallelization allows us to obtain a traceroute measurement to all routed /24's in a short period of time—about 2 days for a team of 13 monitors probing 7.4 million /24's (that is, the full routed address space subdivided into /24's) at 100pps.

We currently have three teams active, and each team independently probes the same set of routed /24's, by sending probes to different random destinations within each /24, and typically to different /24's at any given moment in time. We

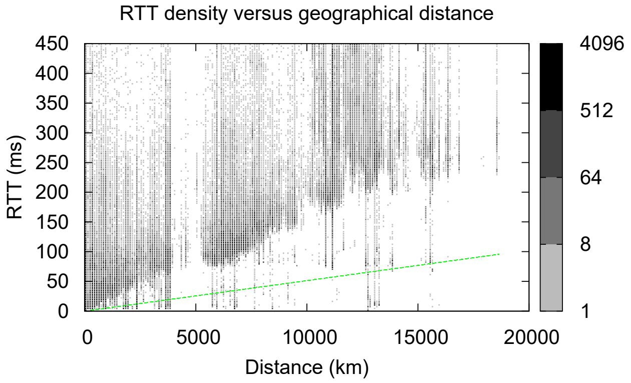

Figure 2. Round-trip time vs. geographic distance for traces to 637k destinations from a single Ark node on the US east coast.

probe the set of /24's themselves, as well as the destinations within each /24, in random order[2]. Random probing more broadly distributes measurement traffic topologically, reducing bias and measurement gaps caused by packet loss or transient routing problems on common links. The multi-dimensional randomness in probe ordering – of prefixes, assignments of prefixes to monitors per cycle, and of addresses within prefixes – also avoids regular probing patterns (such as sequential walking of an address block) that may elicit complaints.

We perform traceroute measurements using *scamper*, a flexible active measurement tool supporting IPv4, IPv6, traceroute, and ping. Scamper implements TCP-, UDP-, and ICMP-based traceroute measurements, including the Paris traceroute variants [7, 8]. Our experiments currently use ICMP Paris traceroute,[3] which a recent Ark-supported experiment determined to be the best overall topology probing method [23]. Scamper has been developed and maintained for several years by our collaborator Matthew Luckie at the University of Waikato.

An important product of these traceroute measurements is the CAIDA IPv4 Routed /24 Topology Dataset [1], which is available for download by researchers. We have collected this data from September 12, 2007 to present. As of Nov 30, 2008, we have collected 2.1 billion traceroutes in 833 GB of traces.

As an additional incentive for organizations to host Ark nodes, we are developing a set of web pages showing per-node connectivity and performance statistics, as exemplified in Fig. 2.

4.2. DNS resolution

We perform DNS lookups of all IP addresses seen in the IPv4 Routed /24 Topology Dataset. We use a customized bulk DNS lookup service that is capable millions of DNS lookups per day.[4] We attempt DNS lookups as soon as possible after we collect topology data (within 1-2 days) so that the DNS meta-data better matches the state of the Internet at trace collection time.

This collection system yields two datasets, the utility of which we have only begun to explore: 1) a simple IP-to-hostname map and 2) raw DNS query/response traffic generated by the lookup service. The first dataset is useful for annotating IP topology data with information commonly encoded in router names, such as geographic location, link capacity, router type (access vs. backbone), and customer network name. The second dataset is useful for studying characteristics of DNS name servers, such as the extent of support for DNSSEC and IPv6. Deeper analysis of the data might reveal other information, such as details on the relationships between organizations (backup name-servers can reveal trust and/or hierarchy). Because we probe every routed /24, this DNS traffic dataset includes a broad cross-section of DNS name servers currently in use, which could reveal macroscopic aspects of this critical layer of the Internet architecture, such as extent of redundancy or consolidation over time.

4.3. Alias resolution for router-level Internet maps

The traceroute data described in section 4.1 is a collection of traces, i.e., sequences of IP addresses. Reconstructing the router-level topology from this data requires grouping multiple IP addresses belonging to the same router. This grouping process is called *alias resolution*. Several IP alias resolution heuristic techniques have been developed. We have been working primarily with two techniques: the CAIDA `iffinder` tool [12] and the Analytical and Probe-based Alias Resolver (APAR) [20].

The `iffinder` tool implements one of the first IP alias resolution techniques introduced in the Mercator project [19]. The tool sends UDP probe packets to all or a subset of IP addresses seen in the traces, with destination UDP ports set to presumably unused values. If router R receives such a packet from prober P destined to one of R's IP interfaces, X, while R's route back to P goes via some other of R's IP interfaces, Y, then R is supposed to reply to P with an ICMP `Port Unreachable` message with its source address set to Y. Prober P can thus conclude that interfaces X and Y belong to the same router.

The idea behind the APAR techniques is to check the structure of the set of IP addresses observed in traces ver-

[2] The random ordering of /24's is the same across all cycles.

[3] We performed UDP-based non-Paris traceroute measurements up to Nov 2, 2007.

[4] The DNS lookup software code was written by David Moore.

sus common IP address assignment schemes. For example, IP addresses configured on point-to-point interfaces often belong to either /30 or /31 subnets. We can use this hint to check for the boundary IP addresses in such /30 and /31 subnets in the two paths going in opposite directions, thus inferring which IP addresses are likely configured on the same router. For example, if the direct trace is two IP addresses X, Y, while the reverse trace is Y', X', and both pairs (X, X') and (Y, Y') belong to the same /30 or /31 subnets, then we can conclude that X and Y' are configured on the same router. The APAR authors claim that this approach is more accurate, efficient, and simpler than all other existing techniques.

To support the needs of our global measurement project, we first cleaned and upgraded the original APAR code, improving its scalability by at least two orders of magnitude: the original code was used for tens of thousands IPv4 address pairs, while we intend to use it for millions of address pairs. We also augmented the tool with a new probing algorithm that increases the accuracy of subnet identification as follows. One of the central steps in APAR is the inference of interface subnets. In its initial version, APAR performs this inference iteratively, working from long candidate subnet prefixes (/30) to shorter subnet prefixes (e.g., /28's). We realized that this inference could be improved by checking for the presence of broadcast addresses in the candidate subnets. Specifically, if the address of an **observed** interface would be the broadcast address of a long subnet prefix (e.g., /30), then the subnet prefix of that interface must actually be shorter (e.g., /29 instead of /30). Therefore, we will explicitly probe some addresses not observed in Ark traces in order to better distinguish the candidate subnets. Note that more accurate subnet inference will avoid both false positives and false negatives in the later alias inference stage. An erroneously large subnet can cause false negatives in aliases because of the rule that two interfaces can not be aliases if they lie within the same subnet. An erroneously large subnet can also cause false positives because it creates more opportunities for lining up path segments. We combined all modifications and additions to produce `kapar`, an optimized and scalable version of the APAR probing code.

We are proceeding toward obtaining a router-level map of the Internet. First, we have conducted two runs of `iffinder` on the deployed Ark infrastructure. In each run, we probed all IPv4 addresses seen in Ark topology traces collected during the two-month interval immediately preceding the date of `iffinder` run. We have also augmented `iffinder` measurements with parallelized `ping` TTL measurements, and are currently cleaning the data for analysis and use in a `kapar` run. Analysis of the `iffinder` and the `kapar` output will allow us to find IPv4 addresses (interfaces) that physically belong to the same router as well as identify links between actual routers rather than IP interfaces. This knowledge will allow us to convert our IPv4-level map of the Internet into a more realistic router-level map.

Note that we used all Ark monitors to run `iffinder` and that this probing was parallel to, but did not hinder, the ongoing topology data collection. Success of these concurrent measurement experiments showcases the Ark infrastructure's versatility and multifunctional capabilities.

4.4. AS-level Internet topology maps

We will derive an AS-level topology map of the Internet from Ark and Route Views [2] data. This process consists of three steps. First, traceroute-like measurements capture the sequence of IP interface hops along the forward path from the source to a given destination. Second, using BGP tables provided by Route Views, we map the IP addresses in the gathered IP paths to the AS numbers that advertise the longest IP prefixes matching the corresponding IP addresses. If two consecutive IP hops in a trace resolve to different ASes, we interpret it as a link between these ASes. The set of these links constitutes an AS-level topology graph.

Mapping traceroute-observed IP addresses to AS numbers using BGP routing tables involves potential distortion, e.g., due to AS-sets, private ASes, multi-origin ASes (the same prefixes advertised by multiple ASes [25]), and unresolved links. Both multi-origin ASes and AS-sets create ambiguous mappings between IP addresses and ASes, so we filter them out. We also filter private ASes as they create false links. Unresolved IP hops in the traceroute data give rise to indirect links, i.e., links that connect two resolved IP hops with one or more unresolved hops in between. We discard indirect links as well. In our previous analysis of AS-level topology maps based on skitter data, total discarded and filtered links usually constituted approximately 5% of all links in the initial set of observations [24].

The resulting AS topology represents a simple undirected unweighted graph. To make this graph more realistic we will augment it with annotations, assigning various attributes characterizing links and nodes. Those assignments define specific link and node *types* abstracting intrinsic structural and functional differences of graph elements – ASes in our case. The annotations empirically ground our topology model by introducing reality constraints into the graph. Simply reproducing the structure of the Internet without any annotations is insufficient; we must also understand and reproduce annotations.

For the Internet topology at the AS level, link annotations represent different business relationship between ASs, e.g., customer-to-provider, peer-to-peer. To infer AS relationships, we will utilize techniques developed at CAIDA based on multiobjective optimization [15]. The main idea behind these inference heuristics is an optimally balanced trade-off between AS relationship information that can be extracted from AS degrees and maximization of the number of valid paths in the resulting annotated AS topology. We use these heuristics on topology data collected by Ark to provide weekly updates of AS business relationships observed in the global Internet [11].

Node annotations of the AS-level Internet graph may represent different types of ASes, e.g., large or small Internet Service Providers (ISPs), exchange points, universities, customer enterprises, etc. [16]. An example of such taxonomy is available in [6]. It can be regularly updated using publicly available data from CAIDA, Route Views, and Internet Routing Registries. Augmenting the AS-level graph with appropriate per-node or per-link annotations will allow us to capture and more accurately reproduce a variety of important global graph properties. For example, instead of considering only shortest paths in this graph, we may be able to study the structure of paths that respect other constraints imposed by routing policies and AS business relationships, as well as path diversity and network resilience to random or intentional attacks.

4.5. Dual AS-router level Internet topologies

After completing research steps described in sections 4.3 and 4.4, we will obtain two independent maps of the Internet topology::
- map 1: IP address to router ID;
- map 2: IP address to AS number.

Although derived from the same raw data, these maps are intrinsically distinct because they are derived using completely different techniques: heuristics to resolve IP addresses that are assigned to the same router for map 1, and mapping IP addresses to AS numbers using Route Views for map 2. Our task is to merge these two maps creating a dual AS-router level Internet topology. In this dual graph, links between ASes are annotated with router IDs that actually connect those ASes and nodes in the router-level graph are annotated with AS numbers to which these routers belong.

Unfortunately, traceroute data contains no information that would indicate which router physically belongs to which AS. Thus, assigning routers to ASes is non-trivial. Given the two maps, we can only compose a map from each router ID to the set of ASes that advertise the IP addresses assigned to this router. Such a map does not unambiguously identify router ownership. We are testing new heuristics to construct topologies that simultaneously and accurately represent the Internet at both the router and AS granularities.

Our current approach to the problem of assigning routers to ASes uses empirical data collected by Ark monitors as well as our previously developed heuristics inferring AS business relationships [11]. The basic idea is that ASes in the Providers category provide address space for connections to their customers. Therefore, it is the Customer side of a Provider-Customer AS link that physically owns the router representing this link. The Provider side only lends an address from its address space to an interface on the Customer's router. When IP addresses on both sides of the link belong to the same AS, our job is easy: we assign this router to this AS. In a few cases when we cannot determine a Provider-Customer relationship for a set of ASes accessing the same router, we assign this router to the AS with the smallest outdegree. We will refine this ownership analysis using hostnames, exchange point information, and available government data sources. We will also conduct a validation through surveys of infrastructure owners.

The resulting dual map will merge router- and AS-level graphs into an integrated view where links and nodes in both graphs are consistently annotated with semantically relevant meta-data. This map will still be incomplete, but it will represent a huge step forward in Internet mapping. The ability to construct and regularly update such maps of the Internet will contribute to answering both practical and theoretical questions about the present and future Internet. It will increase our situational awareness of this critical infrastructure as well as open new grounds for understanding, describing, and modeling Internet evolution.

5. Enabling macroscopic Internet research

We have already demonstrated that researchers can use Ark to quickly design, implement, and easily coordinate the execution of experiments across a widely distributed set of dedicated monitors. Ark coordination facilities also assist researchers with data transfer, indexing, and archiving. Two researchers outside of CAIDA have already made successful use of Ark for their measurement projects.

In early 2008 Matthew Luckie, a collaborator in New Zealand, used Ark infrastructure to study which topology probing method is the most efficient in discovering the Internet topology. For example, do per-flow load balancers implement different forwarding policies for TCP and UDP? Archipelago provided a perfect platform for launching this comparison study, and we co-authored a paper for this year's IMC conference [23]. We found that ICMP-based traceroute methods tend to successfully reach more destinations, as well as collect evidence of a greater number of AS links. We also discovered UDP-based methods infer the most IP links, despite reaching the fewest destinations.

More recently, we are supporting researcher Rob Beverly with extending the scope of his MIT spoofer analysis project [10], for which we hope to have results by early 2009. We will be deploying traffic listeners at each Ark monitor, which will receive UDP probes from spoofer test clients, and forward the traffic over the tuple space to Rob's server for analysis of the extent of 'spoofable' networks.

6. Looking forward

We are in the early stages of an exciting project, and look forward to the Internet measurement infrastructure we have built getting substantial use, by us as well as other research communities. We are now gathering the largest set of IP topology data available to researchers, and continue to expand the set of analysis tools we use and the questions we ask of the data. In 2009 we will perform ongoing IPv6 topology measurements, explore more dynamic IPv4 topology measurements using our new ad-hoc topology measurement facility, and implement new visualizations of IP- and AS-level topology. We will continue to support software needs for third parties conducting specific vetted measurements. We hope this work will eventually lead to the capability to regularly provide rich topology maps of observable Internet infrastructure, as well as support other Internet research and homeland security objectives.

7. Acknowledgments

This project would not be possible without the support of all the sites hosting monitors.

The project is sponsored by U.S. Department of Homeland Security (DHS) Science and Technology (S&T) Directorate via Contract N66001-08-C-2029, "Science and Technology of Internet Mapping", and by the National Science Foundation via grant CRI 0551542, "Community-oriented Network Measurement Infrastructure."

References

[1] CAIDA IPv4 routed /24 topology dataset. `http://imdc.datcat.org/collection/1-0360-J`.

[2] Routeviews sh ip bgp snapshots. `http://archive.routeviews.org/oix-route-views/`.

[3] Ruby language. `http://www.ruby-lang.org/`.

[4] Scapy. `http://www.secdev.org/projects/scapy/`.

[5] The Metasploit Project. `http://www.metasploit.com/`.

[6] Autonomous System taxonomy repository. `http://www.ece.gatech.edu/research/labs/MANIACS/as_taxonomy/`.

[7] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with Paris traceroute. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 153–158, Rio de Janeiro, Brazil, Oct. 2006.

[8] B. Augustin, T. Friedman, and R. Teixeira. Measuring load-balanced paths in the Internet. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 149–160, San Diego, California, USA, Oct. 2007.

[9] B. Augustin, T. Friedman, and R. Teixeira. Multipath tracing with Paris traceroute. In *Workshop on End-to-End Monitoring Techniques and Services (E2EMON)*, pages 1–8, Munich, Germany, May 2007.

[10] R. Beverly. The mit/bbn spoofer project. `http://spoofer.lcs.mit.edu/`.

[11] CAIDA. AS links annotated with AS relationships dataset. `http://www.caida.org/data/active/as-relationships/index.xml`.

[12] CAIDA. iffinder. `http://www.caida.org/tools/measurement/iffinder/`.

[13] CAIDA. Macroscopic Topology Measurements. Research Project. `http://www.caida.org/analysis/topology/macroscopic/`.

[14] N. Carriero and D. Gelernter. *How to write parallel programs: a first course*. MIT Press, Cambridge, MA, USA, 1990.

[15] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claf fy, and G. Riley. AS relationships: Inference and validation. *Comput Commun Rev*, 37(1), 2007.

[16] X. Dimitropoulos, D. Krioukov, G. Riley, and kc claffy. Revealing the Autonomous System taxonomy: The machine learning approach. In *PAM*, 2006.

[17] D. Gelernter. Generative communication in linda. *ACM Trans. Program. Lang. Syst.*, 7(1):80–112, 1985.

[18] D. Gelernter and N. Carriero. Coordination languages and their significance. *Commun. ACM*, 35(2):97–107, 1992.

[19] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *INFOCOM 2000*, pages 1371–1380, Tel-Aviv, Israel, Mar 2000.

[20] M. Gunes and K. Sarac. Resolving IP aliases in building traceroute-based Internet maps. Technical Report UTDCS-62-06, University of Texas at Dallas, 2006.

[21] Y. Hyun. Archipelago measurement infrastructure. `http://www.caida.org/projects/ark/`.

[22] E. Katz-Bassett, H. V. Madhyastha, J. P. John, and A. Krishnamurthy. Studying black holes in the Internet with Hubble. In *Networked Systems Design and Implementation (NSDI)*, Apr 2008.

[23] M. Luckie, Y. Hyun, and B. Huffaker. Traceroute Probe Method and Forward IP Path Inference. In *IMC'08*, Oct 2008.

[24] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, kc claffy, and A. Vahdat. The Internet AS-level topology: Three data sources and one definitive metric. *Comput Commun Rev*, 36(1):17–26, 2006.

[25] Z. M. Mao, J. Rexford, J. Wang, and R. Katz. Towards an accurate AS-level traceroute tool. In *Proc. ACM SIGCOMM*, pages 365–378, Karlsruhe, Germany, Sept. 2003.

[26] S. Ossowski and R. Menezes. On coordination and its significance to distributed and multi-agent systems. *Concurrency and Computation: Practice and Experience*, 18(4):359–370, 2006.

[27] N. Spring, D. Wetherall, and T. Anderson. Scriptroute: A public Internet measurement facility. In *4th USITS*, Mar 2003.