4TH QUARTER 2007, VOLUME 9, NO. 4

IEEE COMMUNICATIONS SURVEYS
The Electronic Magazine of Original Peer-Reviewed Survey Articles

www.comsoc.org/pubs/surveys

INTERNET TOPOLOGY DISCOVERY: A SURVEY

BENOIT DONNET, UNIVERSITÉ CATHOLIQUE DE LOUVAIN
TIMUR FRIEDMAN, UNIVERSITÉ PIERRE & MARIE CURIE AND CNRS

ABSTRACT

Since the beginning of the nineties, the internet has undergone impressive growth. This growth can be appreciated in terms of the equipment, such as routers and links, that has been added, as well as in the numbers of users and the value of commerce that it supports. In parallel to this expansion, over the past decade the networking research community has shown a growing interest in discovering and analyzing the internet topology. Some researchers have developed tools for gathering network topology data while others have tried to understand and model the internet's properties. These efforts have brought us to a crucial juncture for toplogy measurement infrastructures: while, previously, these were both small (in terms of number of measurement points) and monolithic, we are starting to see the deployment of large-scale distributed systems composed of hundreds or thousands of monitors. As we look forward to this next generation of systems, we take stock of what has been achieved so far. In this survey, we discuss past and current mechanisms for discovering the internet topology at various levels: the IP interface, the router, the AS, and the PoP level. In addition to discovery techniques, we provide insights into some of the well-known properties of the internet topology.

This survey focuses on measurements of the *network topology*, i.e., the representation of the interconnection between directly connected peers in the network. While some of this information can be gleaned from passive measurements, researchers largely obtain the topology and its characteristics from active measurements.

There are three different levels at which to describe the network topology: the *link layer topology*, the *network layer topology*, sometimes referred to generically as the *internet topology*, and the *overlay topology*. The link layer topology, as defined by Breitbart *et al.* [1], refers to the characterization of the physical connectivity relationships that exist among entities in a communications network. In other words, it is the description of how data link layer devices, switches and bridges, are interconnected and how the different hosts are connected to them.

Maintaining an accurate and complete knowledge of the link layer topology is a prerequisite to many critical network management tasks such as network diagnostics and resource management.

There is considerable scientific literature devoted to techniques for the discovery of link-layer topology. Interested readers might refer to Robertson [2], Tang and Sugla [3], Wood *et al.* [4], Breitbart *et al.* [1], Lowekamp *et al.* [5] and, more recently, Bejerano [6]. However, a deep description of link layer topology discovery mechanisms is beyond the scope of this article as link-layer topology discovery is an intra-domain task and this survey focuses on topology discovery at the scale of the internet.

A typical overlay topology would be the topology of a peer-to-peer system. An overlay topology can be *unstructured* or *structured*. Structured overlays are exemplified by distributed hash tables, such as Chord [7] or CAN [8]. As explained by Stutzbach *et al.* [9]," peers select neighbors through a predominantly random process. An overlay topology is influenced by peer participation (i.e., join and leave mechanisms) as well as the protocol behavior (i.e., neighbor selection). Characterizing an overlay topology can be done by examining properties of snapshots of the overlay." These snapshots can be gathered using a topology crawler, an engine that queries peers for a list of their neighbors ([9, 10]).

As stated by Stutzback *et al.* [11],"a deep understanding of the topological characteristics in overlay systems is required to meaningfully simulate and evaluate the actual performance of the proposed search and replication techniques."

The overlay topology has drawn the attention of the net-

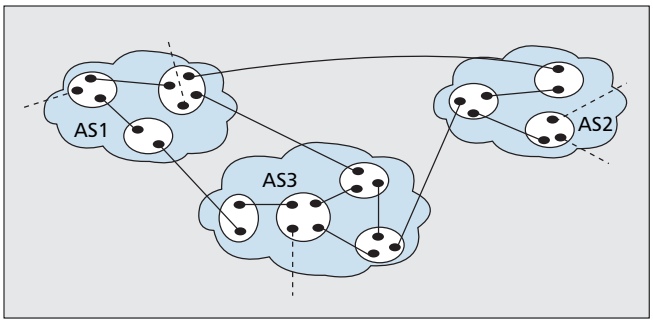

Figure 1. *The different levels of Internet topology.*

working research community in the past few years. However, in this article, we are not directly concerned with peer-to-peer systems. Consequently, describing the overlay topology in more detail would be beyond the scope of this survey. Interested readers might refer to the work of Ripeanu *et al.* [9], Stutzbach *et al.* [11] and Liang *et al.* [12].

The internet topology, the subject of this article, can itself be seen at four different levels. The first one, the *IP interface level*, considers IP interfaces of routers and end-systems. Usually, this topology is obtained by using data collected with a probing tool such as traceroute. The second level, *the router level*, treats each router as a single node in the topology graph. It can be obtained by aggregating IP interfaces through a technique called *alias resolution* ([13–16]). The point of presence (PoP) level, is a third level, that can be obtained by further aggregating the routers, or directly aggregating the interfaces, that are identified as being geographically co-located. Finally, the *AS level* provides information about the connectivity of autonomous systems (ASes). This information is not primarily drawn from active measurements, but rather from inter-domain routing information and address databases.

Figure 1 illustrates three levels of the levels of the internet topology. Black dots represents router interfaces, blank shapes stand for routers and shaded areas for ASes. The plain and dotted lines correspond to links. The IP interface level is illustrated by the black dots. The router level is obtained when all interfaces of a router are grouped in a single identifier. Finally, the AS level is obtained when we look only at ASes and the links between them.

Gathering information about the Internet topology is a mandatory task for modeling the network but also for monitoring the network. We discuss the motivations for internet topology discovery. The remainder of this survey is organized as follows: we discuss the topology at the IP interface level; we present the router level topology; we talk about the AS level topology; we also address the PoP level topology; we then take a look at known properties of the Internet; Finally, we conclude this article and discusses further directions for internet topology discovery.

Motivations

This section discusses the most principal reasons for wanting to discover the internet's topology.

First, the topology data collected can form the basis for a formal graph of the internet. Depending on the level considered, a vertex in the graph can be an IP interface, a router, a PoP, or an AS. Once the graph is built, one can study its characteristics, such as the *average degree*, the *degree distribution*, the *clustering coefficient* or the *betweenness centrality*. See, for instance, Pastor-Satorras and Vespignani's book [17] for details about graph theory applied to the internet topology. Some of these characteristics will be discussed later.

Values for a particular graph metric may capture a graph's theoretic resilience to failure or say something about its efficiency for routing. The knowledge of appropriate metric values may influence the engineering of future topologies, strategies for repair in the face of failures, and the understanding of fundamental properties of existing networks.

Further, the properties derived from the internet graph can be used as input to simulations. Because of the complexity of the network, simulations play a vital role in the attempt to characterize how different facets of the internet behave, and how proposed changes might affect the different network properties. Well-known simulators, such as ns [18] and SSFNet [19] use topology generation algorithms to allow researchers to perform their simulations.

Internet mapping Topology data gathered can also be used to draw a map of the network. Interested readers might find topology maps on the Cooperative Association for Internet Data Analysis (Caida) website [20]. According to Cheswick *et al.* [21], such a map can be useful to monitor the connectivity of the internet. For instance, it might be helpful to visualize how connectivity changes before and during an attack on the internet infrastructure.

Knowledge of the internet topology might have some applications in security. For instance, Burch and Cheswick propose to use internet topology information to track anonymous packets back to their source [22].

Siamwalla *et al.* find that network topology information can be applied to network management [23]. They claim that network topology information is useful in deciding where to add new routers and to figure out whether current hardware is correctly configured. It also allows network managers to find bottlenecks and failures in the network.

Protocol design can use network topology knowledge. For instance, Radoslavov *et al.* discuss the impact of topology on the design and evaluation of four multicast protocols [24].

Multimedia content is increasingly shared between internet users. In order to improve the quality of service (QoS) offered to users and provide a high availability of the shared data, it is common to store the data in replicated servers distributed across the internet. The replication of data over different machines makes the choice of its location a challenging problem that can be addressed with knowledge of the internet topology. Interested readers might refer, for instance, to the work performed by Qiu *et al.* [25] and Radoslavov et al [26].

IP Interface Level

As explained earlier, the IP interface level is the first level of the internet topology. It is composed of the IP interfaces of routers and end-hosts. All routers and some hosts have multiple interfaces, and each interface appears as a separate node in this topology. The graph's links consist of the link-layer connections between nodes. These may not be point-to-point beneath IP: there may be tunnelling across lower-layer protocols, such as MPLS, and there might be traversal of multiple layer-2 devices.

ºSiamwalla *et al.* [23] propose four algorithms to discover the IP layer topology. The first one is an algorithm based on SNMP. The algorithm's principle is the following: for each router, one finds neighboring routers from that router's `ipRouteTable` entry. Hosts are obtained from the router's Address Resolution Protocol (ARP) [32] table entries. ARP is a protocol used to map IP network addresses to the hardware addresses used by a data link protocol. All entries are obtained through SNMP. Each router is pinged to be sure it is alive. This algorithm can only be used on networks where SNMP is enabled on all routers.

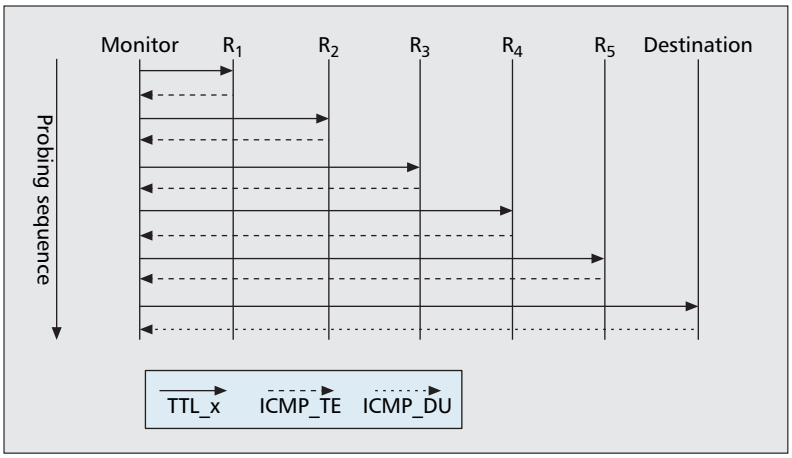

Figure 2. *Traceroute example.*

The second algorithm is based on the *broadcast* ping and the *DNS transfer zone*. A broadcast ping refers to a ping packet addressed to an entire subnet. This can be done by addressing either the "255" or the "0" node in the subnet. A broadcast ping is received by all hosts in the subnet, each of which is supposed to reply to the originator of the ping. A domain's DNS name server keeps a binding from every name in the domain to its IP address [33, 34]. Most DNS servers respond to a "transfer zone" command by returning a list with every name in the domain. It is thus useful in finding all hosts and routers within a domain. The idea behind this algorithm is to, first, get the list of all hosts in a domain, using the DNS transfer zone and, second, check the validity of this list with a broadcast ping. However, this algorithm heavily depends on DNS transfer zone and broadcast ping, which both may be unavailable for security reasons.

The third algorithm is based on DNS transfer zone and traceroute. The basic idea of the algorithm is to get a list of all routers and hosts in the domain with DNS transfer zone and then initiate a traceroute to each member of this list.

The last algorithm is based only on traceroute. The difference between this algorithm and the previous one is the way which the IP addresses are obtained. Here, a heuristic is used to discover the address space to probe.

In the following, we first describe how traceroute works. We next provide an overview of several internet topology tools-based on traceroute. We finally discuss the limitations of traceroute-based mapping.

TRACEROUTE

Traceroute is a networking tool that allows one to discover the path a data packet takes to go from a machine S (the *source* or the *monitor*) to a machine D (the *destination*). Traceroute was created by Van Jacobson in 1989. A variant of Van Jacobson's traceroute, the NANOG traceroute, is maintained by Gavron [35]. NANOG traceroute has additional features such as AS lookup, TOS support, microsecond timestamps, path MTU discovery and parallel probing.

Figure 2 illustrates how traceroute works. `Monitor` is the source of the traceroute, `Destination` is the destination and the Ris are the routers along the path. The monitor sends multiple User Datagram Protocol (UDP) probes into the network with increasing time-to-live (TTL) values. Each time a packet enters a router, the router decrements the TTL. When the TTL value is one, the router determines that the packet has consumed sufficient resources in the network, drops it, and informs the source of the packet by sending back an Internet Control Message Protocol (ICMP) time exceeded message (`ICMP_TE` in Fig. 2). By looking at the IP source address of the ICMP message, the monitor can learn one of the IP addresses of the router at which the probe packet stopped.

When, eventually, a probe reaches the destination, the destination is supposed to reply with an ICMP *destination unreachable* message (`ICMP_DU` in Fig. 2) with the code *port unreachable*. This works if the UDP packet specifies a high, and presumably unused, port number, i.e., above 1024.

Unfortunately, the traceroute behavior explained above is the ideal case. A router along the path might not reply to probes because the ICMP protocol is not enabled, or the router employs *ICMP rate limiting*. In order to avoid waiting an infinite time for the ICMP reply, the traceroute monitor activates a timer when it launches the UDP probe. If the timer expires and no reply was received, then, for that TTL, the machine is considered as *non-responding*. Such a router is also called an *anonymous router*. Yao *et al.* have proposed heuristics for inferring more accurate topologies in the presence of anonymous routers [36].

Further, a particular problem occurs when it is the destination that does not reply to probes because, for instance, of a restrictive firewall. In this case, the destination will be recorded as non-responding but it is impossible to know that it was reached. In order to avoid inferring abound less path, an upper bound on the number of successive non-responding machines is used. For instance, in skitter, this upper bound is set to five. In comparison, Van Jacobson's traceroute limits traceroutes to 30 hops total, by default.

Standard traceroute, as just described, is based on UDP probes. However, two variants exist. The first variant is based on ICMP. Instead of launching UDP probes, the source sends ICMP *Echo Request* messages. With ICMP traceroute, the destination is supposed to reply with an ICMP *Echo Reply*. The second variant sends Transport Control Protocol (TCP) packets. The TCP traceroute [37] aims to bypass most common firewall filters by sending TCP SYN packets. It assumes that firewalls will permit inbound TCP packets to specific ports, such as port 80 (HTTP), listening for incoming connections. The behavior of the traceroute for the intermediate routers is the same as in standard traceroute.

TRACEROUTE–BASED MAPPING

Nowadays, Skitter [38], developed by CAIDA, is probably the best-known mapping system. Skitter records paths from a source to many destinations using parallel ICMP traceroutes. Skitter stores the replies from each router on the path to the destination host, along with the round-trip times (RTTs). Skitter is run on 24 monitors scattered around the world (in the United States, Canada, the United Kingdom, France, Sweden, the Netherlands, Japan, and New Zealand). The different monitors share a common destination set of 971, 080 IPv4 addresses. Each monitor cycles through the destination set at its own rate, taking typically three days to complete a cycle. Similarly to Skitter, *Scamper* [39] makes use of several monitors to traceroute IPv6 networks. Furthermore, it implements a TCP-based traceroute.

RIPE NCC's Test Traffic Measurement (TTM) [40] measures key parameters of the connectivity between a given site and other test boxes. The TTM system performs measurements in a full mesh between roughly a hundred monitors. In addition to traceroute data, the TTM system also records,

among others, one-way delay,[1] packet loss, and bandwidth. Measurements have been performed approximately once every ten minutes, starting October 2002. An anonymized version of the measurement data is freely available.[2]

NLANR's Active Measurement Project (AMP)[41] performs active measurements connected by high performance IPv4 networks. 150 AMP monitors are currently deployed and take site-to-site measurements. AMP monitors are mainly deployed throughout the United States. Some monitors are, however, scattered around the world: among others, Taiwan, Switzerland, Chili and Korea host AMP monitors. Like RIPE NCC TTM, NLANR AMP avoids probing outside its own network. In addition to traceroute, AMP measures RTT, packet loss, and throughput. An IPv6 version of AMP performs measurements between eleven sites. Finally, note that regular AMP data collection ceased in early September 2006. Starting July 2006, CAIDA took over operational stewardship of all NLANR machines and data.

The Distributed Internet Measurements and Simulations [42] (DIMES) system is a measurement infrastructure that achieves a large scale by following the model of SETI@home [43]. SETI@home provides a screensaver that users can freely install, and that downloads and analyzes radio telescope data for signs of intelligent life. The project obtains a portion of the computing power of the users' computers, and in turn the users are rewarded by the knowledge that they are participating in a collective research effort, by attractive visualisations of the data, and by having their contributions publicly acknowledged. DIMES provides a publicly downloadable route tracing tool, with similar incentives for users. It was released as a daemon in September 2004. The DIMES agent performs internet measurements such as traceroute and ping at a low rate, consuming at peak 1KB/sec. At the time of writing this survey, DIMES counts more than 8, 700 agents scattered over five continents.

Branigan *et al.* [44] use a simple traceroute to map the internet. They randomly select a target host from every network announced via BGP to the internet core or listed in various internet databases. They stop discovering a route when it contains at least two unresponsive hops or when a firewall is encountered. They scan about 10% of the list of networks each day and scan the entire list on the first of each month.

Atlas [45] is a system that facilitates the automatic capture of IPv6 network topology information from a single probing source. Atlas is based on "source-routed IPv6 traceroute", i.e., itperformstracerouteonIPv6 networks and the traceroute can use source routing facilities. Although source routing is largely disabled in IPv4 networks, it is enabled in IPv6 networks. Source routing allows greater coverage than can ordinarily be achieved by a single traceroute monitor. Atlas consists of four components: the *probe engine*, which collects raw path information by exhaustively using source-routed traceroutes between all known addresses; the *topology constructor*, which builds the connectivity topology graph based upon the path information gathered by the probe engine; the *topology verifier*, which looks at the constructed topology and reasserts the existence of routers and links modeled by the graph; and the *interactive visualization program*, which extends the 3D hyperbolic space layout tool H3 [46], a tool allowing interactive navigation of topology graphs. To initiate the discovering process, Atlas relies on probing paths among a set of known addresses called *seeds*. The seeds are derived from theinformationinthe6Bone registry[47], a public database of sites and their respective address allocations. To increase probing performance without overloading the network, Atlas uses caching. For each trace, the probe engine caches the hop distance to the *via-router*, which is the intermediate router used for source routing. If the same via-router is used in a subsequent trace, then the cache distance provides the initial hop distance and alleviates the need to re-probe from the probing source to that via-router.

Mercator [14] is a system that uses a mechanism similar to traceroute to infer the internet map. The challenge is twofold: performing measurements from a single point in the network and using only hop-limited probes. Mercator does not need a destination list as input. Rather, it makes use of the *informed random address probing* heuristic. This heuristic aims at determining the targets of probes both by results from earlier probes, as well as by exploiting common IP address allocation policies. It does not rely upon the availability of any external information, such asBGP routing tables. Mercator uses source routing, where available in IPv4, to direct probes in other directions than radially from the sender and to discover cross links. To limit the overhead caused by probes sent in the network, Mercator uses two heuristics:

- The probing is self-clocking, i.e., the subsequent probe is not sent until the response to the previous one has been received. In addition, a mechanism is added to avoid "starvation", i.e., waiting for a response that does not arrive.
- Not all probes start with a TTL of one. For each path to probe, Mercator identifies the furthest router R in that path that was already in the map and starts probing at the TTL corresponding to R. If the first response is from R, Mercator continues to probe the path. Otherwise, it backtracks and starts probing with a TTL of one.

Rocketfuel [15] tries to get the best possible picture of individual internet service providers(ISPs), typically ones that are in the center, not on the edges of the internet. Rocketfuel is based upon three principles:

- Measurement selection so that the number of required probes is decreased. This is done with two heuristics: *directed probing* (i.e., the identification of traceroutes passing through the ISP) and *path reduction* (i.e., the elimination of traceroutes passing through already known paths). Note that path reduction is composed of two techniques: *ingress reduction* and *egress reduction*. Ingress reduction is based on the observation that probes to a destination from multiple monitors may converge and enter a target ISP at the same node. Egress reduction acts on the observation that probes to multiple destinations may leave the target ISP at the same node.
- Alias resolution, i.e., identifying different IP interfaces belonging to the same router
- Router identification and annotation, i.e., determining which router belongs to the ISP being mapped, its geographical location and its part in the topology. To perform this, Rocketfuel relies upon DNS and ISP naming conventions. Readers interested by the geolocation of internet hosts might refer to [48–51].

Scriptroute [52] is a system that allows an ordinary internet user to perform network measurements from several distributed vantage points. It proposes remote measurement execution on PlanetLab nodes [53] through a daemon that implements ping, traceroute, hop-by-hop bandwidth measurement, and a number of other utilities. Note that Script route uses the Reverse Path Tree (RPT) discovery tool to avoid overloading the network when multiple monitors probe towards a given

[1] *This is possible as each box in the system has a GPS.*

[2] *See http://watt.nlanr.net/active/maps/ampmapactive.php for details on the available data set.*

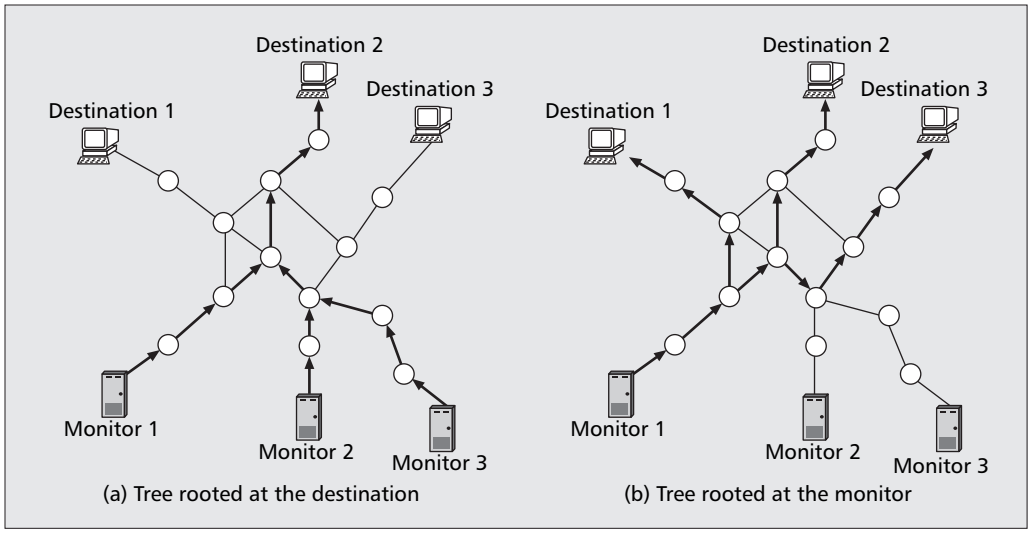

Figure 3. *Tree-like structure of routes: a) tree rooted at the destination; b) tree rooted at the monitor.*

destination. A reverse path tree is a destination-rooted tree, i.e., a tree formed by routes converging from a set of monitors on a given destination (Fig. 3a). The RPT tool avoids retracing paths by embedding a list of previously observed IP addresses in the script that directs the measurements. A given monitor stops probing when it reaches a part of the tree that has already been mapped.

Doubletree [54], proposed by the authors of this article, is a cooperative network topology discovery algorithm. Doubletree assumes that routes in the internet have a tree-like structures, as illustrated in Fig. 3. Routes converging towards a destination from multiple monitors form a tree that is rooted at the destination (Fig. 3a). Similarly, routes leading out from a monitor towards multiple destinations form a tree rooted at the monitor (Fig. 3b). A monitor probe shop by hop so long as it encounters previously unknown interfaces. However, once it encounters a known interface, it stops, assuming that it has touched a tree and the rest of the path to the root is also known. Using these trees suggests two different probing schemes: backwards (based on a monitor-rooted tree) and forwards (based on a destination-rooted tree). It is not necessary for Doubletree monitors to maintain information about the whole tree structures. Instead, both backwards and forwards probing use data structures, called *stop sets*. The one for backwards probing, called the *local stop set*, consists of all interfaces already seen by that monitor. Forwards probing uses the *global stop set* of (interface, destination) pairs accumulated from all monitors. A pair enters the stop set if a monitor visited the interface while sending probes with the corresponding destination address.

Table 1 summarizes and compares the different probing techniques discussed in this section.

Looking first at the technique used, we note that most of the tools are based on a standard traceroute (ICMP, UDP or TCP), as explained earlier. Atlas and Mercator differ from the others as they make use of the source routing IP option. In addition, Doubletree and Mercator also makes use of backwards probing (i.e., decreasing TTL instead of increasing).

The majority of techniques are dedicated to IPv4, except Scamper, AMP, and Atlas, which are designed for IPv6 networks. We believe that, in the near future, tools will have to evolve towards IPv6, as it is becoming ever more widespread.

Regarding the number of monitors spread around the world, one can see that DIMES is currently the largest-scale deployed technique. Scriptroute is currently deployed over 200 PlanetLab hosts but the number of used sources depends on the experiment that is run. We do not indicate any source information about Doubletree as it is not yet deployed as a full-time service, despite the fact a prototype has already been implemented [55].

Finally, the last column in Table 1 indicates whether the method makes use of an overhead reduction technique for details about network overhead induced by traceroute probing). One can see that five methods do not use any method for overhead reduction. These methods only implement standard traceroute behavior, as explained earlier.

The probing far mechanism, the caching, and the RPT techniques are similar to each other, while the ingress and egress reduction heuristics are similar to Double tree's forwards and backwards stopping rules. However, Rocketfuel applies its heuristics exclusively at the boundaries of ISPs, and so it does not take advantage of the redundancy reductions that might be found by paths that converge within an ISP. Doubletree reduces redundancy starting at the point of convergence, wherever that might be found. Nor does Rocketfuel employ backwards probing. In contrast, Doubletree employs both. Besides, Rocketfuel assumes a centralized controller, thus it does not consider how the information regarding where to stop probing could be efficiently encoded for exchange between monitors. In Doubletree, this is done through the global stop set for forwards probing, encoded as a series of (interface, destination) pairs.

Limitations and Issues

Although active probing methods have key advantages, they have their own share of limitations due to inherent inaccuracies of hop-limited probes. Implementation and network load issues are also prone to arise, particularly when a distributed infrastructure is used.

In addition to difficulties encountered by traceroute, as described earlier, one immediate limitation is the fact that these probes only discover forward paths towards a given destination, since reverse paths may differ because prefix-based routing policies and hot-potato routing can cause asymmetry. Asymmetric routing also prevents inferences from being made on the link delay between two consecutive hops from the reported round-trip time: the difference between the RTTs could be due to the link, or to the existence of different return paths from the two routers. A way to partially circumvent this issue is to perform a mesh measurement, as carried out by NCC and AMP.

A second limitation is that active probing follows primary paths and thus miss out on backup paths (i.e., a path that is

Technique	Name	Network	# Sources	Overhead reduction technique
Traceroute	skitter	IPv4	24	None
	scamper	IPv6	6	
	TTM	IPv4	142	
	AMP	IPv4/v6	150	
	DIMES	IPv4	10,200	
	Rocketfuel	IPv4	800	ingress and egress reduction
	Scriptroute	IPv4	200	Reverse Path Tree (RPT)
Source routing	Atlas	IPv6	1	caching
	Mercator	IPv4	1	self-clocking and probing far
Forward and backward	Doubletree	IPv4	N.A.	stop set

Table 1. *Comparison of traceroute-based methods.*

used only when the primary path is broken) if these are not needed. A larger time-window for probing as well as source routing increases, however, the possibility of discovering them[14].

A third limitation is that classic traceroute has been found to systematically misfunction when traffic is split across multiple paths by a load balancer. This is due to traceroute's use of certain header fields to identify its probes, these fields being the same ones that load-balancing routers use to identify flows. The resulting topology contains missing nodes and links, as well as false links. The recently introduced *Paris traceroute* tool goes some way towards correcting these problems [56], but its techniques are not yet widely adopted.

A fourth limitation can occur inside an AS that internally makes use of multi-protocol label switching (MPLS) [57], which gives it the ability to hide the underlying topology, suspending the TTL mechanism used by traceroute. Routers using MPLS may be configured either to decrement the TTL, as traceroute requires, or to ignore the TTL field because, the switched path of MPLS being configured to have no loops, the IP TTL is not needed. The MPLS specification, however, recommends that the TTL be decremented where possible [57]. Note that current version of traceroute, making use of ICMP extensions [58, 59], can be aware of MPLS nodes along the path.

Fifth, active measurements are at the mercy of inconsistent behavior from networked elements: misconfigurations sometimes lead to the appearance of private non-routable addresses, non-RFC compliant implementations cause different interfaces to respond depending on their router's vendor, and firewalls prevent some probed routers from responding.

Lastly, Hyun *et al.* quantify the magnitude of traceroute inaccuracies in real world traceroute paths [60]. They focus on *third-party address*, i.e., the address of a router interface that does not lie in the actual path taken by packets. Using traceroute data collected by skitter, Hyun *et al.* find that most of the third-party addresses are located at the destination edge of the network, multihoming[3] being the most likely cause.

In addition to these inherent inaccuracies, one has to point out the network load generated by probe traffic. Donnet *et al.* analyze this issue and distinguish two types of redundancy measurements carried out by traceroute-like probing mechanisms [54]. The first is the *intra-monitor redundancy*. It corresponds to the redundancy measurements made by an individual monitor in isolation from the rest of the system. Donnet *et al.* show that intra-monitor redundancy occurs mainly close to a monitor. Indeed, due to the tree-like structure of routes rooted at the monitor, a monitor tends to repeatedly trace over these interfaces. In a large-scale system, the degree of such redundancy could be very high as a nearby interface would be visited for each probed destination. The second is the *inter-monitor redundancy*. This occurs when one monitor duplicates another's work. Donnet *et al.* show that such redundancy mainly arises close to destinations due to the tree-like structure of routes emanating from a set of monitors and converging on the destination. Such redundancy grows linearly with the number of monitors.

In addition to imposing additional load on internal nodes and links of the network, redundant measurements can also appear to destination nodes as distributed denial-of-service (DDoS) attacks, since probes would be received from a large set of sources.

As explained earlier, Rocketfuel, Scriptroute, Mercator, Atlas and Doubletree have developed techniques for reducing the probing redundancy. Note also that standard traceroute has been extended in order to permit backwards probing, i.e., starting from the destination and decreasing the TTL ([61, 62]).

Router Level

The second level of the internet topology, as explained earlier (Fig. 1), is the router level topology. It can be seen as an aggregation of the IP interface level, i.e., the summary of all the IP addresses of a router into a single identifier. The summary technique is called *alias resolution* and is illustrated in Fig. 4. As explained earlier, traceroute lists interfaces addresses from path and identifies, in our example, interfaces A, B, C_1, C_2 and D (Fig. 4a). Alias resolution clusters all interfaces of a router to reveal the true topology. As shown in Fig. 4b, interfaces C_1 and C_2 are aliases. Alias resolution has received recent attention. Based on synthetic topologies, Gunes and Saracs how that the accuracy of alias resolution has an impor-

[3] *Multihoming refers to the ability of having different connections to the internet, potentially through different providers.*

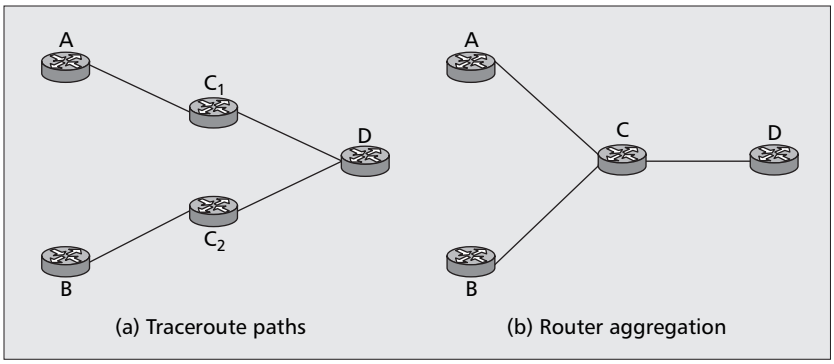

Figure 4. *Alias resolution principle: a) traceroute paths; b) router aggregation.*

tant effect on the observed topological characteristics such as, for instance, the number of nodes and edges or the average node degree [63]. In this section, we describe the currently existing approaches for alias resolution.

The first is the *address based method* and is described in RFC 1122 [64]. The principle is simple: the source sends a UDP probe with a high port number to the router's interface X. If the source address of the resulting "Port Unreachable" ICMP message is Y, then X and Y are aliases for the same router. The drawback of this solution is that some routers do not generate ICMP messages, making alias resolution impossible. This technique has been implemented in many tools, such as *iffinder* [65] and Mercator.

The second is the *IP identification based method* and has been implemented in *Ally*, Rocketfuel's alias resolution component. Ally is based on the ID field of the IP header. The basic idea is the following: send a UDP probe packet with a high port number to the two potential aliases. The "Port Unreachable" ICMP responses are encapsulated within IP packets and, so, each one includes an IP identifier (x and y). Then, one sends a third packet to the address that responded first. Assume that z is the IP identifier of the third response and x was the IP identifier of the first response. If $x < y < z$ and $z - x$ is small, the addresses are likely aliases. This method, in like the address based method, works only if a router responds to probes.

The third is the *DNS based method*. This method considers similarities in router host names and works when an AS uses a systematic naming scheme for assigning IP addresses to router interfaces. This method is especially interesting as it can work even if a router does not respond to probes directed to itself. Ally uses this technique against unresponsive routers with the help of the Rocketfuel's name DNS decoder.

The fourth is the *graph based method* proposed by Spring *et al.* [66].This method extracts from traceroute outputs agraph of linked IP addresses in order to infer likely and unlikely aliases. It is based on two assumptions:

- If two IP addresses precede a common successor IP address, then they are likely to be alias
- Two addresses found in a same traceroute are unlikely to be aliases

This method is mainly used as a preprocessing step to reduce the number of probe pairs for an active probing approach, such as the address and IP identification based methods.

The fifth method is the Analytical Alias Resolver (AAR) introduced by Gunes and Sarac [67]. They propose a graph theoretic formulation of the alias resolution problem and developed the AAR algorithm to solve it. Given a set of path traces, AAR utilizes the common IP address assignment scheme to infer IP aliases within the collected path traces.

The sixth is the *TTL-limited with record route option method* proposed by Sherwood and Spring [68]. The idea is to perform a standard traceroute with the Record Route (RR) IP option enabled. This option is supposed to force an intermediate router to record its IP address in the IP packet that traverses it. Due to size constraints, an IP packet cannot contain more than nine IP addresses of intermediate routers. Note that the addresses discovered by traceroute and RR do not overlap as RR records the outgoing interface while the *time exceeded* message solicited by traceroute comes from the in-going interface. Figure 5 illustrates how this method works. The probing source discovers both ingoing and outgoing interfaces of each router along the path by performing traceroute with the RR option enabled. The ith address in the RR array is an alias for the router that sends the ICMP time exceeded message if both addresses are different. This technique works only in networks where routers support the RR option, which is not necessarily the case in modern networks.

Finally, the last method is called the *IPv6 based method* and has been implemented in Atlas. Atlas tries to find addresses belonging to the same router relying on the assumption that routing header processing in IPv6 routers is separate from delivering packets to the TCP/UDP layers. To elicit the equivalence of two addresses X and Y, Atlas performs a traceroute to Y via X. When the first probe reaches router X, at a distance h, the first swaps the address X in the destination field with final address Y contained in the routing header.[4] Next, the hop limit is checked. If we assume that the value is 1, an ICMPv6 *hop limit exceeded in transit* message response is triggered. Because the destination address field of the probe packet is now Y, the source address of the ICMPv6 response also becomes Y.[5] The next probe packet, with hop limit h1, is delivered to the UDP layer, causing a *port unreachable* response. Thus, if X and Y belong to the same router, the trace X-Y will report Y-Y and the trace Y-X will report X-X.

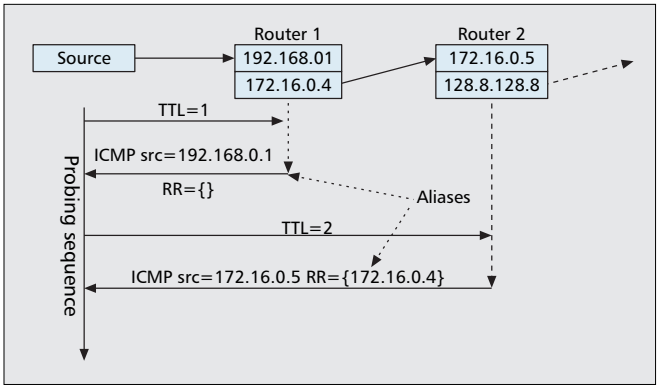

Figure 5. *Alias resolution using record route. Figure from [68].*

[4] *Source routing in IPv6 is based on a routing header that specifies a list of intermediate nodes that a packet has to traverse on the path to its destination. See RFC1883 [69] for details about routing header and Gain [70, Appendix A.2] for an example of source routing in IPv6.*

[5] *Determining the source address of an ICMP message in IPv6 is somewhat different to IPv4: if the ICMP message is a response to a message sent to one of the router unicast addresses, the source address of the ICMP message must be that same address. See RFC2463 [71] for further details on ICMPv6.*

Method	Name	Technique	Available implementation
Active	Address based	UDP packet	iffinder, Mercator
	IP based	UDP packet	Ally
	DNS based	DNS request	Ally
	IPv6 based	source routing	Atlas
	TTL-limited record route	record route	Passenger
Analytical	Graph based	common successor	None
	AAR	common IP address assignment	

Table 2. *Comparison of alias resolution techniques.*

Table 2 proposes a comparison of the various alias resolution techniques presented in this section.

We classify the techniques in two groups. The first group covers the *active methods*, i.e., those that need to inject additional traffic in the network and require router participation in order to resolve aliases.[6] To be efficient, these alias resolution methods must be applied, preferably, at the same time the data is collected. All of these techniques have been implemented in tools that are deployed in the internet.

The second group concerns the *analytical methods*. On the contrary to active methods, the analytical methods do not require additional traffic. Further, they can be applied "offline", i.e., after the data has been collected. Note that the common successor and the common IP address assignment are quite similar in their basic principles. It makes the assumption of point-to-point links. This is illustrated in Fig. 6, where boxes are routers and circles are interfaces. At left is the IP address graph. Nodes *A*, *B* and *C* represent interface address. At right, *A* and *B* are shown to represent interfaces on the same router, connected by point-to-point link to *C*. To the best of our knowledge, none of these techniques has been implemented in any tool.

AS Level

An Autonomous System (AS) is either a single network or a group of networks that is under the control of a single administrative entity, typically an ISP or a very large organization (for instance, a university, a business enterprise or division) with independent connections to multiple networks. An AS is also sometimes referred to as a *routing domain*. Each AS is identified by a unique 16-bit number assigned by the internet assigned numbers authority (IANA). Note that the special problem of determining the topology within a single AS is a separate area of inquiry (see, for instance, Bejerano and Rastogi's work [72]). It is not, strictly speaking, internet topology discovery, and it is considerably helped by the privileged access available to the administrator of an AS.

In this section, we first describe the relationships between ASes. Secondly, we discuss topology information sources at the AS level. Finally, we detail the evolution of AS topology discovery and, in particular, the methods used to infer it.

AS Relationships

In the internet AS topology graph, an edge between two ASes(nodes) represents a business relationship which results in the exchange of internet traffic between them. An AS can have one or more relationships with different kinds of neighboring ASes. Each relationship may correspond to several distinct physical links.

On one side, an AS' *access links* connect to customer networks. Customer networks buy internet connectivity from the AS. On the other side, *peering links* connect to transit providers from which it buys its own connectivity. Peering links also connect to private peers with which exchange of traffic is negotiated without exchanging money as a way to avoid sending traffic through a provider. No transit traffic is allowed through peering links; only traffic with the peer or its customers is permitted. These are the most observed relationships in the network and are usually referred to as the provider-to-customer (p2c), customer-to-provider (c2p) and peer-to-peer (p2p) relationships.

A less common relationship found in the internet is called the sibling-to-sibling (s2s) relationship. This relationship generally resides between two ASes of a same company. The key difference with peering is that siblings exchange all kinds of traffic, not only between their respective customers. An s2s relationship covers everything except the p2c, c2p and p2p relationships. It appears in various cases such as when two

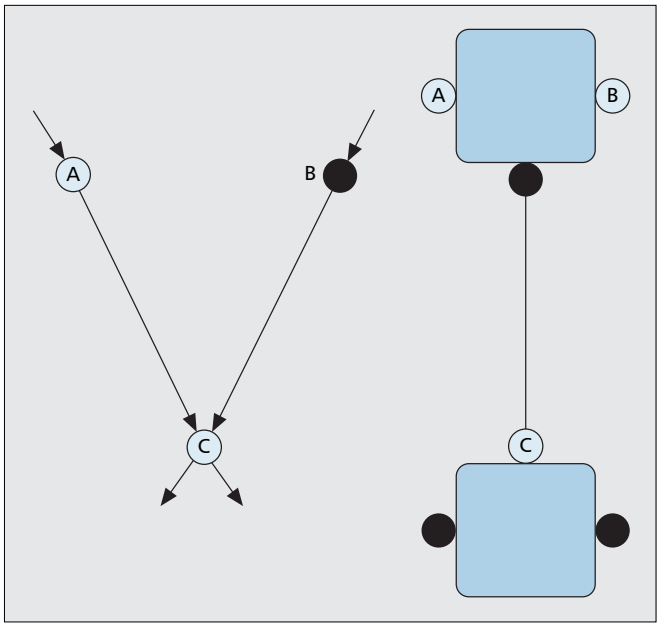

Figure 6. *Analytical method for alias resolution. Figure from [66].*

[6] *This is not the case for the TTL-limited record route method.*

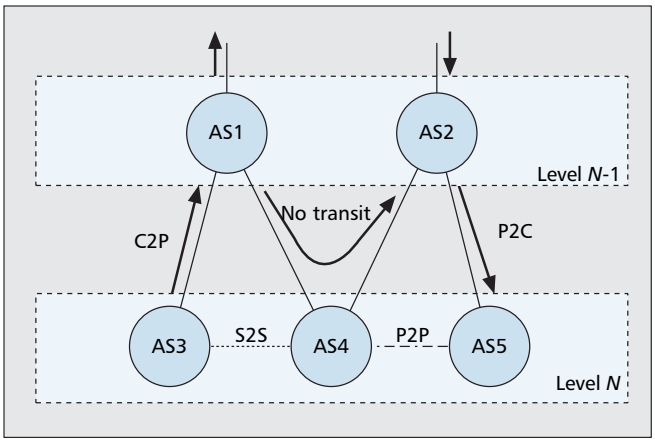

Figure 7. *AS relationships.*

ASes act as backups for each other, or when two ISPs merge and decide to become siblings instead of merging into a single AS which can be very complex. Two peering ISPs have a special agreement for specific prefixes for which they transit all kinds of traffic for each other. Figure 7 illustrates AS relationships.

These relationships have a major impact on routing in the internet, as shown by Tangmunarunkit *et al.* [73]. Inside an AS, routing uses *hop-count* as a metric, but because intra-domain protocols support hierarchies, the resulting paths are not always the shortest in terms of *hop-distance*. Between ASes, routing is determined by policy. Many internet path lengths thus may also benefit from a detour [74, 75] which would incur more router-level hops than shortest-router-hop path routing. For simulation purpose, it is therefore most appropriate to model the network with policy-based routing rather than AS shortest path-based routing.

Topology Information Sources

Two sources of AS level topology data are available: *internet registries* and *BGP routing information*. This section describe these two sources along with their advantages and limitations.

Routing Registry Information — Many publicly-available registries share information about the internet and its topology. *Regional Internet Registries* [76] are organizations responsible for allocating AS numbers and IP address blocks, all of which are accessible using the WHOIS protocol[77]. Internet Routing Registry (IRR) [78] is another group of databases maintained by several organizations and containing documented routing policies. These policies are available through the WHOIS protocol and are expressed in the Routing Policy Specification Language (RPSL) [79].

Topology discovery using internet registry information has several advantages. Firstly, the access is simpler and more efficient to implement than active method probing, such as those described earlier. Indeed, they do not have to explore the network to obtain the topology and the information is grouped at specific locations. Secondly, they provide high-level information such as routing policies which are otherwise more difficult to obtain.

This information source has, however, limitations mainly due to the fact that they are based on data provided by ISPs and not based on the real state of the network. Firstly, the provided information is often incomplete for various reasons such as confidentiality and administrative overhead. Secondly, as shown in RIPE consistency check reports[80], registry data quality is questionable and often inconsistent as information about a same object in one registry overlaps and sometimes even contradicts information in other registries. Thirdly, due to their inherent nature, these registries are not able to precisely reflect the actual state of routing in the network. For instance, it cannot determine whether portions of the internet are reachable or not, or whether backup links exist and are used.

These limitations are the reason why current work has tended to focus on other information sources for topology discovery at the AS level. Nevertheless, routing registries still provide a useful source of information when combined with other sources.

BGP Routing Information — As opposed to link-state protocols such as OSPF [81] or IS-IS [82], BGP does not maintain any unified view of the network. Each BGP router chooses its best path for a specific destination which is propagated to its neighbors, leading to an individual view of the network for each router. This view depends on factors such as the choices made by its neighbors, the order in which it receives their announcements, etc. This distributed nature calls for the use of information gathering methods in order to obtain the most complete common view of the topology.

Common BGP information sources are *looking glasses* and *route servers*. A looking glass is a web interface to a BGP router which usually allows BGP data querying and limited use of debugging tools such as ping and traceroute. A route server is a BGP router offering interactive login access permitting to run most non-privileged router commands. Both are usually made public to help network operators in their debugging tasks, but they can also provide BGP information to properly crafted network discovery tools. A list of accessible looking glasses and route servers is available at [83].

A second source of BGP information is *BGP dumps*. Projects such as RouteViews [84] or RIPE NCC provide collected information from BGP routers around the world. Route collectors are deployed in various locations and peer with BGP routers from multiple ASes. They then periodically save snapshots of their state, known as *table dumps*, along with all routing updates received between the preceding and current snapshot, known as *updatetraces*. Another way to get BGP information is to use a Zebra router configured to log all BGP update messages. Zebra is an open-source routing daemon [85].

There are several advantages to AS level topology discovery using BGP routing information. First, in the fashion of routing registries, data has been gathered and is available at specific places. There is therefore no need to deploy an infrastructure for exploring the network. Secondly, unlike routing registry data, provided information by BGP corresponds to the actual state of the network, even though it only provides local views of it. Finally, BGP update traces allows dynamic behavior analysis such as backup link detection.

Using BGP routing information has, however, drawbacks. As noticed by Chang *et al.* [86], BGP does not provide complete information due to missing AS relationships that include both p2c and p2p type relationships. Further, BGP routing information seems to provide a less complete picture of inter-domain routing as for example using node-probing, confirmed by Broido and claffy studies [87].

Inferring an AS Level Topology

Early research assumed that two ASes were linked if their AS numbers were adjacent in an AS path. Gao and Rexford [88] then made a substantial advance by noticing c2p links creating so a hierarchy. Gao went on to identify the p2p and s2s relationships [89].

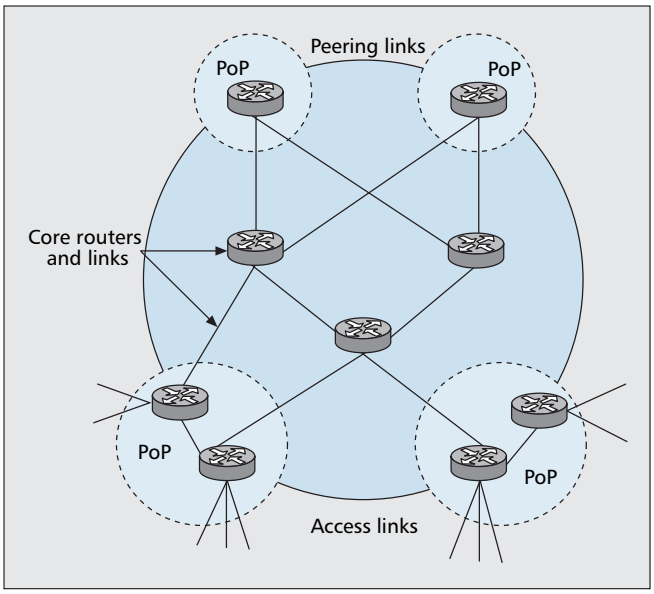

Figure 8. *PoP level topology within an AS.*

Inferring these relationships is a problem of its own. In her study, Gao [89] first tackles the problem by developing an inference mechanism which extracts information from BGP tables and summarized *valley-free*[7] property of AS paths. Subramanian *et al.* [90] formulates AS relationship assignment as an optimization problem, a type of relationship (ToR) problem. Battista *et al.* [91] prove its NP-completeness and present an approximately optimal solution. Gao and Xia [92] evaluate then the accuracy of these algorithms and improve them by introducing techniques on inferring relationships from partial information. This improvement was made possible due to Quoitin and Bonaventure's work that show how the BGP community attribute indicate relationships amongst ASes [93].

Andersen *et al.* propose a method of inferring logical relationships between network prefixes within an AS using only passive monitoring of BGP update messages [94].ABGP update message is either an announcement, or a withdrawal. Each update message contains a timestamp indicating the second at which it was received and the prefix that was affected. Andersen, *et al.* group IP address prefixes based upon how frequently they observe BGP updates for both prefixes within the same time window. Then, a clustering algorithm is applied to join these prefixes into successively larger groups.

Chang *et al.* show that many existing links do not actually appear in BGP [95]. Chang *et al.* propose to infer the AS topology from internet's router topology. Given a path obtained thanks to traceroute, Chang *et al.* determine the AS of each router along this path and extract ASes adjacency information from the resulting AS sequence. To achieve that, Chang *et al.* build an AS mapping table. The idea is to map each existing address prefix to the corresponding AS. This mapping is based on two resources: the BGP routing tables and the IRR. Broido and claffy [87] reports that the obtained topology differs from the BGP inferred ones in having much denser inter-AS connectivity. It is also richer because it is capable of exposing multiple points of contact between ASes. This is in contrast to BGP table dumps that only provide information on whether two ASes peer or not.

Chang *et al.* also propose a means to identify *border routers* of an AS. This is not trivial as the IP addresses of a border router might belong to its own AS, to the AS of a peer, or to a third party such as an Internet eXchange Point (IXP).

Mapping IP addresses to AS number is not as simple as it may seem. A common problem is the origination of a same prefix by multiple ASes, known as the multiple origin AS (MOAS) problem. Zhao *et al.* show that the number of conflicts is not trivial (the median value in 2001 was 1294 conflicts) [96]. An IP-to-AS mapping study by Mao *et al.* [97] identifies that 10% of traceroute paths contained one or more hops that did not map to a single AS number. Furthermore, mapping IP addresses to AS numbers paths result in loops in the inferred AS path for about 15% of the node-level paths examined. As loops are not permitted by BGP, this indicates an error in mapping. Mao *et al.* improve accuracy by proposing heuristics comparing BGP-derived AS paths against traceroute-derived AS paths and by performing reverse DNS lookups. The heuristics, though effective, are labor-intensive and mostly ad hoc; Mao *et al.* improve this result and propose a systematic way to perform the same tasks using dynamic programming anditerative improvement [98].

Although these topologies inferred from various sources present substantial differences, their comparison by Mahadevan *et al.* [99] seems to have underlined fundamental characteristics of the network, such as its *joint degree distribution* that provides information about 1-hop neighborhoods around a node. However, the question of which most closely matches the actual internet AS topology remains open.

POP LEVEL

A point of presence (PoP) is a collection of routers owned by an AS in a specific location (city or suburb). A PoP level topology can be produced by adding information about geographic location to inter-AS topology. Different ASes sometimes have routers in the same building, such places are known as colocation facilities or *exchange points*. In a PoP level graph, the links between two PoPs belonging to a same AS are its *backbone or corelink*. The links between two nodes belonging to different ASes can either be the access or peering links. Fig. 8 illustrates a PoP level map.

A PoP level analysis is useful for understanding the geographic properties of internet paths as it provides straightforward constraints about latency between two PoPs.

The pioneering work of Govindan and Tangmunarunkit with Mercator provides techniques for inferring detailed PoP level topologies using traceroutes: IP addresses appearing in traceroute paths are mapped to their corresponding PoP by performing reverse DNS lookups. In later work, Teixeira *et al.* find that inferred topology had significantly higher path diversity[16], i.e., distinct number of AS paths exist between an AS and the rest of the internet. Teixeira *et al.* suspect that the large number of false links were due to imperfect alias resolution. However, this could not explain the false PoP level edges. Recent developments by Pai *et al.* [100] show DNS misnaming to be a major source of false edges and offer ways to fix them.

KNOWN PROPERTIES OF THE INTERNET

Probably, the most well-known study about the internet topology was performed by Faloutsos *et al.* [101]. They find the existence of relationships between several properties of the AS graph. These relationships are expressed as *power laws*. Recall that a power law has the shape $y \propto x^a$, where the two values that are of interest are x and y and $\propto$ stands for "pro-

[7] *After traversing a p2c or a p2p edge, the AS path cannot traverse a c2p or p2p edge. In other words, an AS does not provide transit between any two of its providers or peers.*

portional to." The first relationship found by Faloutsos *et al.* is between the out-degree of a node (the number of outgoing edges) and its rank (its index in order of decreasing out-degree). It reflects the way domains connect. There is a trade-off between the benefits and the cost of adding an edge, from a financial and functional point of view. The second power law links an out-degree value and its frequency. It indicates that the internet node degree distribution is not arbitrary. The higher degrees are rare and the lower degrees are the most frequent. Note that this aspect is a natural behavior of power laws. The last power law establishes a relation between the eigenvalue λ_i of the graph adjacency matrix and the order i. This power law means that the eigen exponent can distinguish the differences between graph families. For example, the eigen exponent of an inter-domain graph will be different from the eigen exponent of the router graph. Finally, Faloutsos *et al.* derive formulas that link the exponent of the power laws with useful graph metrics, such as the number of nodes, the number of edges and the average neighborhood size. Magoni and Pansiot [102] provide several extensions to Faloutsos *et al.*'s work. Broido and claffy further show that Weibull distribution can be used to approximate the outdegree distribution of routers [87].

Barabási and Albert [103] explain the origin of power laws in the internet by two factors: *preferential connectivity* and *incremental growth*. The preferential connectivity of a new node is the tendency of a new node to connect to existing nodes with a high out-degree. Incremental growth refers to open networks that are formed by the continual addition of new nodes, and thus the gradual increase in the size of the network. Medina *et al.* [104] also find two other reasons for the existence of power laws in the internet. First, they consider how the nodes are distributed in space. Medina *et al.* assume that the internet topologies have a high degree of clustering. Another possible cause for power laws is the tendency of anew node to connect to existing nodes that are close in distance.

Tangmunarunkit *et al.* [105] ask whether an alternative explanation to the AS power law node degree distribution given by Barabzi and Albert exists. Their observations, based on data collected by Mercator over a period of one year and a half, suggest that AS size[8] might determine degree distribution and that AS sizes are highly variable. The observation that degree follows size captures the idea that large ASes, by setting up a large initial capital investment and building out a large network, are able to attract more customers and peers than smaller ASes.

Barford *et al.* [106] ask how far the underlying topology can be precisely characterized when the number of end-toend vantage points increases. Barford *et al.* assume that the network graph being studies is composed of two parts, the central routing core and a set of links feeding the backbone, and they classify nodes accordingly. Viewing the topology from a small number of sources (< 5), many backbone nodes are misclassified. If one increases the number of sources, the classification increases in accuracy. However, the *marginal utility* of adding sources decreases rapidly. By marginal utility, Barford *et al.* mean the incremental benefit obtained by conducting one or more additional measurements. This incremental benefit could be seen with a fixed destination set and even with a bigger data set and not fixed destinations. By fixed destinations, the authors mean that every source has the same destination set. Not fixed destinations means that each source has its own destination set, different from other sources. The fact that additional measurements may provide low marginal coverage does not necessarily imply that the overall coverage is high. Finally, the results shed light on how typical IP routes pass through a relatively well defined switching core.

Faloutsos *et al.* proposed that the hierarchical internet structure could be represented in a more compact way through power laws. The accuracy of this model has been discussed by Chen*et al.* [107].Chen *et al.* ask whether the measurements used by Faloutsos *et al.* are sufficiently complete to establish a strict power law relationship for AS vertex degree distribution. They also ask how the available measurements can be used to establish the validity of topological models at AS level as defined by Barabási and Albert. Chen *et al.* estimate that the Barabási and Albert model (BA model) does not explain why this kind of distribution appears in an AS context. Furthermore, the vertex degree distribution and peering characteristics of new ASes are more complex than the simple models used in the BA model. In fact, the distribution of small vertex degrees in the BA model can be considered as equivalent to that in the internet. On the other hand, the large vertex degree distribution in BA model is different from that in the internet. Chen *et al.* also show that BGP routing tables do not constitute a sufficient data source. The internet maintains a richer connectivity than that which can be observed by aggregating a small number of BGP routing tables.

Lakhina *et al.* [108] show that if a graph has a node degree distribution that is very different from a power law, sampling from a small set of vantage points will yield a picture of a graph with a node degree distribution that follows a power law. One of the reasons evoked by Lakhina *et al.* is that the sampled graph has fewer edges than the genuine graph. To detect this bias, two criteria are proposed: *Do the highest degree nodes tend to be near the source(s)?* and *Does the distributional shape near the source differ from that further from the source?*. To perform tests, Lakhina *et al.* made the assumption that sources and destinations are randomly placed in the graph and that routing follows the shortest path. Tests suggested that the genuine router graph might have a higher proportion of high degree nodes that it would seem in the simple measurement extrapolation.

Clauset and Moore [109] explain analytically the bias in the degree distribution described by Lakhina *et al.* in [108]. Clauset and Moore demonstrate that, for sparse random graphs of large average degree, the apparent degree distribution displays a power law of the form $P(k) \sim k^{-1}$ for k below the average degree. According to Clauset and Moore, this illustrates the danger of concluding the existence of a power law from data over too small a range of degrees, and, more specifically, the danger of sampling traceroutes from just a few sources. In addition, Clauset and Moore propose another explanation for the emergence of power law degree distributions. Decisions made by routers interact in a complex way with the link-level topology. It may well be that routers only use a small fraction of the edges in the network, and that the edges they actually use give rise to an effective degree distribution with a power law form.This solution implies that, if the real degree distribution of the internet is something very different from a power law, it may not matter as these extra links are not utilized in normal routing decisions.

These biases have been further studied by Petermann and De Los Rios [110], and Dall'Asta *et al.* [111]. Guillaume and Latapy [112] have extended these studies to include the trade-off between the number of monitors and the number of destinations.

Li *et al.* have shown that the node degree distribution is not enough to describe network topology [113].Li *et al.* point

[8] *Tangmunarunkit et al. evaluated the size of an AS by the number of routers it contains.*

out that there exist many different graphs having the same distribution of node degree. Some of them might be considered opposites from the point of view of network engineering. Instead, Li *et al.* propose models that in corporate technological constraints on router and link bandwidth and connectivity, together with abstract models of user demand and network performance.

Amini *et al.* [114] collect four different data sets from the Oregon Route View Project [84] and the Looking Glass sites[9] [83]. For the analysis, Amini *et al.* consider three metrics: the AS path asymmetry, the BGP `AS_PATH` prediction of traceroute AS path and the AS degree. Based on these data sets, Amini *et al.* demonstrate that the data collection technique can have a major impact on some attributes, such as the AS degree distribution and the average path length.

Conclusion

The past ten years have seen the rise of a new networking measurement area: the internet topology discovery. Due to its particular structure, the network topology can be understood at various levels. In this article, we focused on the work performed by the research community on the network layer topology, sometimes also called the internet topology.

In this article, we first explained that the internet topology discovery is driven by important questions. For instance, one might want to model the internet in order to reproduce its behavior in a laboratory.

Internet is a complex decentralized system that can be decomposed in different sublayers: the IP interface level, the router level, the PoP level and the AS level. Regarding the IP interface level, we explained that the most common technique together data is to use traceroute, a common tool that allows a probing source to elicit router interfaces along the path toward a destination. We also described techniques that extend the standard traceroute in order to trace more efficiently. We finally discussed the limitations and issues related to traceroute-like probing.

The router level aims at summarizing all the IP addresses of a router into a single identifier. This aggregation process is called alias resolution. In this article, we described the two families of alias resolution method. The first one makes use of active probing, i.e., injects additional traffic in the network and requires router participation. The second one is an analytical method. The key advantage is that it can be applied offline, i.e., after the data has been collected.

Concerning the third sublayer, i.e., the AS level, we first provided a reminder about AS relationships and then, described the topology information sources, i.e., the routing registry and the BGP routing information. Next, we discussed techniques for inferring the internet topology at the AS level using these data sets.

We also gave an insight of the PoP level, a PoP being a collection of routers owned by an AS in a specific location.

We finally summarized some of the known properties of the internet. We notably discussed the controversy about the power law relationships in the internet.

However, although the amount of work performed by the research community is huge, this is not the end of the story. We are starting to see the deployment of large-scale distributed measurement infrastructures made of hundreds or thousands of monitors. Thousands of users already download distributed measurement tools. The grenouille [115] tool is widely used for measuring available bandwidth in French ADSL systems. Researchers at Georgia Tech have recently made the passive measurement tool NETI@home [116] available for public download. In addition, on-demand large-scale measurement systems, such as N-Tap [117] and DipZoom [118], are also on the way. Future challenges will concern, for instance, the distribution of gathered data among the measurement points and how to efficiently query this distributed database to provide to the research community or to an application information about the internet topology.

[9] *Each site provides an HTTP interface to run traceroutes to specified destinations and to query the site's local BGP router for the AS_PATH associated with an IP address.*

Acknowledgements

Mr. Donnet's work is supported by the European-founded 034819 OneLab project.

References

[1] Y. Breitbart *et al.*, "Topology Discovery in Heterogeneous IP Networks," *Proc. IEEE INFOCOM*, Mar. 2000.

[2] S. Robertson, "Netdig software," Aug. 1991, available by anonymous ftp from ftp.ctr.columbia.edu:/pub/net/netdig.3.5.share.Z

[3] N. Tang and B. Sugla, "Netmap: A Network Discovery Tool," Network & Services Management Research Labs, Lucent Technologies, Tech. Rep., Sept. 1998.

[4] D. C. M. Wood, S. S. Coleman, and M. F. Schwartz, "Fremont: A System for Discovering Network Characteristics and Problems," *Proc. USENIX Winter Conf.*, Jan. 1993.

[5] B. Lowekamp *et al.*, "A Resource Query Interface for Network-Aware Applications," *Cluster Computing*, vol. 2, no. 2, 1999, pp. 139–51.

[6] Y. Bejerano, "Taking the Skeletons Out of the Closets: A Simple and Efficient Topology Discovery Scheme for Large Ethernet," *Proc. IEEE INFOCOM*, Apr. 2006.

[7] I. Stoica *et al.*, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Trans. Net.*, vol. 11, no. 1, Feb. 2003, pp. 17–32.

[8] S. Ratnasamy *et al.*, "A Scalable Content Addressable Network," *Proc. ACM SIGCOMM*, Aug. 2001.

[9] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design," *IEEE Internet Computing J.*, vol. 6, no. 1, Aug. 2002.

[10] D. Stutzbach and R. Rejaie, "Capturing Accurate Snapshots of the Gnutella Network," *Proc. IEEE Global Internet Symp.*, Mar. 2005.

[11] D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing Unstructured Overlay Topologies in Modern p2p File-Sharing Systems," *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC)*, Oct. 2005.

[12] J. Liang, R. Kumar, and K. W. Ross, "The Kazaa Overlay: A Measurement Study," *Computer Networks*, vol. 49, no. 6, Oct. 2005.

[13] J. J. Pansiot and D. Grad, "On Routes and Multicast Trees in the Internet," *ACM SIGCOMM Computer Commun. Review*, vol. 28, no. 1, Jan. 1998, pp. 41–50.

[14] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet Map Discovery," *Proc. IEEE INFOCOM*, Mar. 2000.

[15] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP Topologies with Rocketfuel," *Proc. ACM SIGCOMM*, Aug. 2002.

[16] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker, "In Search of Path Diversity in ISP Networks," *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC)*, Oct. 2003.

[17] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach*, Cambridge University Press, 2004.

[18] S. Bajaj *et al.*, "Improving Simulation for Network Research," University of Southern California, Tech. Rep. 99-702b, Mar. 1999.

[19] J. Cowie, A. Oglieski, and D. Nicol, "The SSFNet Network Simulator," see http://www.ssfnet.org/homePage.html, Renesys Corporation.
[20] Cooperative Association for Internet Data Analysis, "AS internet graph," see http://www.caida.org/analysis/topology/as_core_network/AS_Network.xml
[21] B. Cheswick, H. Burch, and S. Branigan, "Mapping and Visualizing the Internet," *Proc. USENIX Annual Tech. Conf.*, Jun. 2000.
[22] H. Burch and B. Cheswick, "Tracing Anonymous Packets to Their Approximate Source," *Proc. USENIX Large Installation System Administration (LISA) Conf.*, Dec. 2000.
[23] R. Siamwalla, R. Sharma, and S. Keshav, "Discovering Internet Topology," Cornell University, Ithaca, NY 14853, Tech. Rep., July 1998.
[24] P. Radoslavov *et al.*, "On Characterizing Network Topologies and Analyzing Their Impact on Protocol Design," Computer Science Department, University of Southern California, Tech. Rep. 00-731, Feb. 2000.
[25] L. Qiun, V. N. Padmanabhan, and G. M. Voelker, "On the Placement of Web Server Replicas," *Proc. IEEE INFOCOM*, Apr. 2001.
[26] P. Radoslavov, R. Govindan, and D. Estrin, "Topology-Informed Internet Replica Placement," *Proc. 6th Int'l. Wksp. Web Caching and Content Distribution*, June 2001.
[27] Hewlett-Packard, "HP's Open View Network Node Management," see http://www.openview.hp.com
[28] Tivoli, "Tivoli for AIX," see http://www.tivoli.com
[29] J. D. Case *et al.*, "Simple Network Management Protocol (SNMP)," IETF, RFC 1157, May 1990.
[30] Dartware, "InterMapper: Network Monitoring and Alerting Software," see http://www.intermapper.com
[31] Cisco, "SNMP description," 1996, see http://www.cisco.com/warp/public/535/3.html
[32] D. C. Plummer, "An Ethernet Address Resolution Protocol or Coverting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware," IETF, RFC 826, Nov. 1982.
[33] P. Mockapetris, "Domain Names: Concepts and Facilities," IETF, RFC 1034, Nov. 1987.
[34] P. Mockapetris, "Domain Names: Implementation and Specification," IETF, RFC 1035, Nov. 1987.
[35] E. Gavron, "NANOG Traceroute," see ftp://ftp.login.com/pub/software/traceroute/
[36] B. Yao, V. R., F. Chang, and D. Waddington, "Topology Inference in the Presence of Anonymous Routers," *Proc. IEEE INFOCOM*, Apr. 2003.
[37] M. Torren, "tcptraceroute — A Traceroute Implementation using TCP packets," UNIX," man page, 2001, see source code: http://michael.toren.net/code/tcptraceroute/
[38] B. Huffaker *et al.*, "Topology Discovery by Active Probing," *Proc. Symp. Applications and the Internet (SAINT)*, Jan. 2002.
[39] M. Luckie, "IPv6 scamper," 2005, WAND Network Research Group, See http://www.wand.net.nz/_mjl12/ipv6-scamper/
[40] F. Georgatos *et al.*, "Providing Active Measurements as a Regular Service for ISPs," *Proc. Passive and Active Measurement Workshop (PAM)*, Apr. 2001.
[41] A. McGregor, H.-W. Braun, and J. Brown, "The NLANR Network Analysis Infrastructure," *IEEE Commun. Mag.*, vol. 38, no. 5, 2000.
[42] Y. Shavitt and E. Shir, "DIMES: Let the Internet Measure Itself," *ACM SIGCOMM Comp. Commun. Review*, vol. 35, no. 5, 2005, see http://www.netdimes.org
[43] D. P. Anderson *et al.*, "SETI@home: An Experiment in Public-Resource Computing," *Commun. ACM*, vol. 45, no. 11, 2002, see http://setiathome. berkeley.edu/
[44] S. Branigan *et al.*, "What Can You Do with Traceroute?" *IEEE Internet Computing*, vol. 5, no. 5, Sept./Oct. 2001.
[45] D. G. Waddington *et al.*, "Topology Discovery for Public IPv6 Networks," *ACM SIGCOMM Computer Commun. Review*, vol. 33, no. 3, July 2003, pp. 59–68.
[46] T. Munzer, "H3: Laying Out Large directed Graph in 3D Hyperbolic Space," *Proc. IEEE Symp. Information Visualisation*, Oct. 1997.
[47] 6Bone Registry, http://www.viagenie.qc.ca/en/ipv6/registry/
[48] V. N. Padmanabhan and L. Subramanian, "An Investigation of Geographic Mapping Techniques for Internet Hosts," *Proc. ACM SIGCOMM*, Aug. 2001.
[49] A. Ziviani and S. Fdida, "Toward a Measurement-Based Geographic Location Service," *Proc. Passive and Active Measurement Wksp. (PAM)*, Apr. 2004.
[50] A. Lakhina *et al.*, "On the Geographic Location of Internet Resources," *Proc. ACM SIGCOMM Internet Measurement Wksp. (IMW)*, Nov. 2002.
[51] B. Gueye *et al.*, "Constraint-based Geolocation of Internet Hosts," *IEEE/ACM Trans. Net.*, vol. 14, no. 6, Dec. 2006, pp. 1219–32.
[52] N. Spring, D. Wetherall, and T. Anderson, "Scriptroute: A Public Internet Measurement Facility," *Proc. USENIX Symp. Internet Technologies and Systems (USITS)*, Mar. 2002, see http://www.cs.washington.edu/research/networking/scriptroute/
[53] PlanetLab Consortium, "PlanetLab project," 2002, see http://www.planet-lab.org
[54] B. Donnet *et al.*, "Deployment of an Algorithm for Large-Scale Topology Discovery," *IEEE JSAC*, Sampling the Internet: Techniques and Applications, vol. 24, no. 12, Dec. 2006, pp. 2210–20.
[55] B. Donnet *et al.*, "Evaluation of a Large-Scale Topology Discovery Algorithm," *Proc. IEEE Int'l. IP Operation and Management (IPOM) Wksp.*, Oct. 2006.
[56] B. Augustin *et al.*, "Avoiding Traceroute Anomalies with Paris Traceroute," *Proc. ACM/USENIX Internet Measurement Conf. (IMC)*, Oct. 2006.
[57] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switsching (MPLS) Architecture," IETF, RFC 3031, Jan. 2001.
[58] R. Bonica *et al.*, "Extending the Internet Control Message Protocol (ICMP)," IETF, Internet Draft draft-bonica-internet-icmp-01, Jan. 2006, work in progress.
[59] R. Bonica, D. Gan, and D. Tappan, "ICMP Extensions for Multiprotocol Label Switching," IETF, Internet Draft draftbonica-internet-icmp-00, Mar. 2006, work in progress.
[60] Y. Hyun, A. Broido, and K. Claffy, "On Third-Party Addresses in Traceroute Paths," *Proc. Passive and Active Measurement (PAM) Wksp.*, Apr. 2003.
[61] T. Moors, "Streamlining Traceroute by Estimating Path Lengths," *Proc. IEEE Int'l. Wksp. IP Operations and Management (IPOM)*, Oct. 2004.
[62] B. Donnet, P. Raoult, and T. Friedman, "Efficient Route Tracing from a Single Source," *arXiv*, cs.NI 0605133, May 2006.
[63] M. H. Gunes and K. Sarac, "Importance of IP Alias Resolution in Sampling Internet Topologies," *Proc. IEEE Global Internet Symp.*, May 2007.
[64] R. Braden, "Requirements for Internet Hosts. Communication layers," IETF, RFC 1122, Oct. 1989.
[65] K. Keys, "iffinder," A Tool for Mapping Interfaces to Routers, see http://www.caida.org/tools/measurement/iffinder/
[66] N. Spring *et al.*, "How to Resolve IP Aliases," UW CSE, Tech. Rep. 04-05-04, May 2004.
[67] M. Gunes and K. Sarac, "Analytical IP Alias Resolution," *Proc. IEEE Int'l. Conf. Commun. (ICC)*, June 2006.
[68] R. Sherwood and N. Spring, "Touring the Internet in a TCP Sidecar," *Proc. ACM/USENIX Internet Measurement Conf. (IMC)*, Oct. 2006.
[69] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," IETF, RFC 2460, Dec. 1998.
[70] S. Gain, *Internetworking IPv6 with Cisco Routers*, Mcgraw-Hill (Tx), 1998.
[71] A. Conta and S. Deering, "Internet Control Message Protocol (IMCPv6) for the Internet Protocol Version 6 (IPv6) Specification," IETF, RFC 2463, Dec. 1998.
[72] Y. Bejerano and R. Rastogi, "Robust Monitoring of Link Delays and Faults in IP Networks," *Proc. IEEE INFOCOM*, Apr. 2003.
[73] H. Tangmunarunkit *et al.*, "The Impact of Routing Policy on Internet Paths," *Proc. IEEE INFOCOM*, Apr. 2001.
[74] L. Gao and F. Wang, "The Extent of as Path Inflation by Routing Policies," *Proc. IEEE Global Internet Symp.*, Nov. 2002.
[75] H. Tangmunarunkit, R. Govindan, and S. Shenker, "Internet Path Inflation Due to Policy Routing," *Proc. SPIE Int'l. Symp.*

Convergence of IT and Communication (ITCom), Aug. 2001.
[76] R. I. Registries, see http://www.isoc.org/briefings/021/
[77] L. Daigle, "WHOIS Protocol Specification," IETF, RFC 3912, Sept. 2004.
[78] T. M. Network, "Internet Routing Database," see ftp://ftp.radb.net/routing.arbiter/radb/dbase/
[79] C. Alaettinoglu *et al.*, "Routing Policy Specification Language (RPSL)," IETF, RFC 2622, June 1999.
[80] RIPE NCC, "Routing Registry Consistency Check Reports," see http://www.ripe.net/projects/rrcc/
[81] J. Moy, "OSPF version 2," IETF, RFC 2328, Apr. 1998.
[82] ISO, "Intermediate System to Intermediate System Intra-Domain Routeing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless-Mode Network Service (ISO 8473)," International Standard 10589:2002.
[83] T. Kernen, "Traceroute Organization," see http://www.traceroute.org
[84] University of Oregon, "Route views, University of Oregon Route Views Project," see http://www.antc.uoregon.edu/route-views/
[85] Gnu Zebra, See http://www.zebra.org
[86] H. Chang *et al.*, "Towards Capturing Representative AS-level internet topologies," *Proc. ACM SIGMETRICS*, June 2002.
[87] A. Broido and K. Claffy, "Internet topology: Connectivity of IP graphs," *Proc. SPIE Int'l. Symp. Convergence of IT and Commun. (ITCom)*, Aug. 2001.
[88] L. Gao and J. Rexford, "Stable Internet Routing Without Global Coordination," *Proc. ACM SIGMETRICS*, June 2000.
[89] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *Proc. IEEE Global Internet Symp.*, Nov. 2000.
[90] L. Subramanian *et al.*, "Characterizing the Internet Hierarchy from Multiple Vantage Points," *Proc. IEEE INFOCOM*, June 2002.
[91] G. Di Battista, M. Patrignani, and M. Pizzonia, "Computing the Types of the Relationships Between Autonomous Systems," *Proc. IEEE INFOCOM*, Apr. 2003.
[92] J. Xia and L. Gao, "On the Evaluation of as Relationship Inferences," Proc. IEEE Global Commun. Conf. (GLOBECOM), Nov. 2004.
[93] B. Quoitin and O. Bonaventure, "A Survey of the Utilization of the BGP Community Attribute," IETF, Internet Draft draft-quoitin-bgp-comm-survey-00, Feb. 2002, work in progress.
[94] D. G. Andersen *et al.*, "Topology Inference from BGP Routing Dynamics," *Proc. ACM SIGCOMM* Internet Measurement Wksp. (IMW), Nov. 2002.
[95] H. Chang, S. Jamin, and W. Willinger, "Inferring AS-level Internet Topology from Router-Level Path Traces," *Proc. SPIE Int'l. Symp. Convergence of IT and Commun. (ITCom)*, Aug. 2001.
[96] X. Zhao *et al.*, "An Analysis of BGP Mulitple Origin AS (MOAS) Conflicts," *Proc. ACM SIGCOMM Internet Measurement Wksp. (IMW)*, Nov. 2001.
[97] Z. Mao *et al.*, "Towards an Accurate AS-Level Traceroute Tool," *Proc. ACM SIGCOMM*, Aug. 2003.
[98] Z. M. Mao *et al.*, "Scalable and Accurate Identification of AS-level Forwarding Paths," *Proc. IEEE INFOCOM*, Mar. 2004.
[99] P. Mahadevan *et al.*, "The Internet AS-Level Topology: Three Data Sources and One Definitive Metric," *ACM SIGCOMM Computer Commun. Review*, vol. 36, no. 1, Jan. 2006, pp. 17–26.
[100] M. Zhang, Y. Ruan, V. Pai, and J. Rexford, "How DNS Misnaming Distorts Internet Topology Mapping," *Proc. USENIX Annual Technical Conf.*, May/June 2006.
[101] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," *Proc. ACM SIGCOMM*, Sept. 1999.
[102] D. Magoni and J. J. Pansiot, "Analysis of the Autonomous System Network Topology," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 31, no. 3, July 2001, pp. 26–37.
[103] A. L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, Oct. 1999, pp. 509–12.
[104] A. Medina, I. Matta, and J. Byers, "On the Origin of Power Laws in Internet Topologies," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 30, no. 2, Apr. 2000, pp. 18–28.
[105] H. Tangmunarunkit *et al.*, "Does AS Size Determine Degree in AS Topology?" *ACM SIGCOMM Computer Commun. Review*, vol. 31, no. 5, Oct. 2001, pp. 7–8.
[106] P. Barford *et al.*, "On the Marginal Utility of Network Topology Measurements," *Proc. ACM SIGCOMM Internet Measurement Wksp. (IMW)*, Nov. 2001.
[107] Q. Chen *et al.*, "The Origin of Power Laws in Internet Topologies Revisited," *Proc. IEEE INFOCOM*, June 2002.
[108] A. Lakhina *et al.*, "Sampling Biases in IP Topology Measurements," *Proc. IEEE INFOCOM*, Apr. 2003.
[109] A. Clauset and C. Moore, "Traceroute Sampling Makes Random Graphs Appear to Have Power Law Degree Distributions," *arXiv*, cond-mat 0312674, Feb. 2004.
[110] T. Petermann and P. De Los Rios, "Exploration of Scale-Free Networks," *The European Physical J. B*, vol. 38, 2004, p. 201.
[111] L. Dall'Asta *et al.*, "A Statistical Approach to the Traceroute-Like Exploration of Networks: Theory and Simulations," *Proc. Combinatorial and Algorithmic Aspects of Networking (CAAN) Wksp.*, Aug. 2004.
[112] J. L. Guillaume and M. Latapy, "Relevance of Massively Distributed Explorations of the Internet Topology: Simulation Results," *Proc. IEEE INFOCOM*, Mar. 2005.
[113] L. Li, D. Alderson, W. Willinger, and J. Doyle, "A First-Principles Approach to Understanding the Internet's Router-Level Topology," *Proc. ACM SIGCOMM*, Aug. 2004.
[114] L. D. Amini, A. Shaikh, and H. G. Schulzrinne, "Issues with Inferring Internet Topological Attributes," *Proc. SPIE Internet Performance and Control of Network Systems*, Aug. 2002.
[115] Collectif des ĉablés Wanadoo, "grenouille, la météo du net," Nov. 1999, see http://www.grenouille.com/
[116] C. Simpson and G. F. Riley, "NETI@home: A Distributed Approach to Collecting End-to-End Network Performance Measurements," *Proc. Passive and Active Measurement Wksp. (PAM)*, Apr. 2004.
[117] K. Masui and Y. Kadobayashi, "N-TAP: A Platform of large-Scale Distributed Measurement for Overlay Network Applications," *Proc. Int'l. Wksp. Dependable and Sustainable Peer-to-Peer Systems (DAS-P2P)*, Jan. 2007.
[118] Z. Wen, S. Triukose, and M. Rabinovich, "Facilitating Focused Internet Measurements," *Proc. ACM SIGMETRICS*, June 2007.

BIOGRAPHIES

BENOIT DONNET () received his MS degree in computer science from the Institut d'Informatique of the Facultés Universitaires Notre Dame De La Paix (Namur — Belgium) in 2003. He received his Ph.D. degree in computer science form the Université Pierre et Marie Curie in 2006. He is currently Research Assistant at the Université Catholique de Louvain. His research interests are in internet measurements, focusing on large-scale topology discovery algorithms, Bloom filters, WiMAX and coordinate systems.

TIMUR FRIEDMAN [S'96, A'02, M'04] received the A.B. degree in philosophy from Harvard University and the M.S. degree in management from Stevens Institute of Technology. He received the M.S. and Ph.D. degrees in computer science from the University of Massachusetts Amherst, in 1995 and 2001, respectively. He is currently a Maˆıtre de Conférences (assistant professor) of computer science at the Universit é Pierre et Marie Curie in Paris, and a researcher at the Laboratoire d'Informatique de Paris 6 (LIP6). His research interests include large scale network measurement systems and disruption tolerant networking.