Internet Topology Discovery

Benoit Donnet

Université de Liège – Liège – Belgium

Abstract. Since the nineties, the Internet has seen an impressive growth, in terms of users, intermediate systems (such as routers), autonomous systems, or applications. In parallel to this growth, the research community has been looking for obtaining and modeling the Internet topology, i.e., how the various elements of the network interconnect between themselves. An impressive amount of work has been done regarding how to collect data and how to analyse and model it.

This chapter reviews main approaches for gathering Internet topology data. We first focus on hop limited probing, i.e., traceroute-like probing. We review large-scale tracerouting projects and discuss `traceroute` limitations and how they are mitigated by new techniques or extensions. Hop limited probing can reveal an IP interface vision of the Internet. We next focus on techniques for aggregating several IP interfaces of a given router into a single identifier. This leads to a router level vision of the topology. The aggregation can be done through a process called alias resolution. We also review a technique based on IGMP probing that silently collect all multicast interfaces of a router into a single probe. We next refine the router level topology by adding subnet information. We finish this chapter by discussing the AS level topology, in particular the relationships between ASes and the induced hierarchy.

Keywords: Internet topology, traceroute, alias resolution, IGMP, Merlin, subnet, AS.

1 Introduction

Internet is made of a vast set of heterogeneous and interconnected entities enabling the communication between millions of machines. Typically, this network is described as a graph [1] where nodes refer to IP interfaces, routers, or autonomous systems (ASes)[1] and links represent the existence of a direct connection between those nodes. This is illustrated in Fig. 1 where black dots represent router interfaces, blank shapes stand for routers, and shaded areas for ASes. The plain and dotted lines correspond to links. The router graph can be obtained when all interfaces of a router are grouped in a single identifier. This process is known as *alias resolution*. Finally, the AS level is obtained when we look only

[1] Note there are other possible levels, not shown on Fig. 1, such as the Point-of-Presence (PoP) level [2–5] or the subnet level [6–8]. This latter level will be the subject of Sec. 4, while the PoP level will not be addressed in this chapter.

E. Biersack et al. (Eds.): Data Traffic Monitoring and Analysis, LNCS 7754, pp. 44–81, 2013.

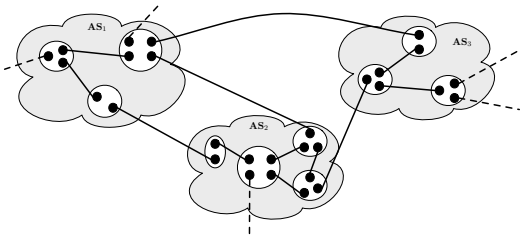


Fig. 1. The different levels of Internet topology

at ASes and the links between them (in some sense, we aggregate all routers belonging to a given AS into a single identifier, the AS number).

This infrastructure, known as *Internet topology*, makes possible the course of the information from a given node towards any other node with the help of intermediate infrastructures acting as relay for the information. This is, de facto, the backbone of a large number of applications, ubiquitous in our society, like Internet browsing, email, peer-to-peer systems, cloud computing, and many others.

Consequently, our deep understanding of the Internet topology properties (see, for instance, [1, 9–14]) and its dynamics is crucial as it impacts our capacity to maintain good performance of the network [15], to improve its efficiency and to design relevant protocols for various applications [16]. Those tasks naturally lean upon theoretical studies and simulations realized with artificial graphs obtained from models of the Internet topology [10].

However, it must be understood that the network organisation cannot be, at a given time, directly available. Its evolution is not ruled by any central authority that could have a global vision of its structure. As a consequence, the data collection can only rely on network measurements: researchers and engineers use complex procedures and tools for building Internet maps, as complete as possible, gathering so light on some Internet properties. Secondly, assuming that the data collected is correct and representative of the actual network, efforts are made for creating Internet models that are essential for simulations [10]. Lots of works have thus been done for collecting larger and larger amount of data and modeling as accurately as possible the network.

In this chapter, we investigate how Internet topology data can be collected. In particular, we focus on techniques for gathering IP interface information through `traceroute`-like probing (Sec. 2). We describe how `traceroute` [17] works, review large-scale projects using `traceroute`, and discuss `traceroute` main limitations and how they can be circumvented. We also focus on the router level of the Internet topology (Sec. 3). We describe techniques for aggregating IP interfaces of a router into a single identifier (i.e., alias resolution), those techniques being active or passive. We also discuss a recent active probing technique, *IGMP probing*, that naturally provides a router level view of the topology. We next refine the router level with subnet information (Sec. 4). Finally, we have a look at the AS level discovery (Sec. 5). In particular, we describe the ASes hierarchy and the relationships between ASes and their inference. We describe the common dataset for studying the Internet AS level topology.

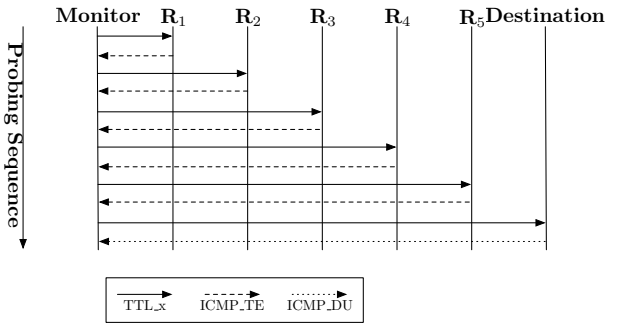


Fig. 2. `traceroute` example

2 IP Interface Level Discovery

The IP interface level is the lowest level of the Internet topology. It considers 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 [18, 19], and there might be traversal of multiple layer-2 devices [20].

As the IP interface level of the Internet topology can be discovered using `traceroute`, we first review how the `traceroute` tool works (Sec. 2.1). Next, we discuss several topology mapping projects that are based on `traceroute` (Sec. 2.2). Finally, we discuss some `traceroute` limitations and how they are fixed (Sec.2.3).

2.1 `traceroute`

`traceroute` [17] is a networking tool for revealing the path a data packet traverses, 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 [21]. NANOG `traceroute` has additional features such as AS lookup, TOS support, microsecond timestamps, path MTU discovery, and parallel probing.

Fig. 2 illustrates how `traceroute` works. `Monitor` is the `traceroute` source, `Destination` is the destination and the $\texttt{R}_i$s 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. It is worth to notice that this source address is supposed to be the outgoing interface of the reply and not the interface on which the packet triggering the reply was received [22].

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.

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` [23] aims at bypassing 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`.

The probing method used has an impact on the collected dataset. Indeed, Luckie et al. show that there are significant differences in the topology observed following the probing method [24]. ICMP-based `traceroute` is able to reach more destinations and to collect more evidence of a greater number of AS links (after an IP-to-AS mapping [25–27]). If UDP-based probing reaches less destinations, it is however able to reveal much more IP links.

2.2 `traceroute`-Based Projects

`traceroute` is a tool easy to implement, manipulate, and deploy (standard `traceroute` is part of any operating system distribution). As such, it became the de-facto standard for probing the network, not only for collecting topology data but also as a network diagnostic tool. In this section, we review several topology mapping projects using `traceroute` for gathering data.

Archipelago [28] is CAIDA's current measurement infrastructure (i.e., it is the successor of *skitter* [29]). Archipelago is based on team probing, i.e., probing monitors are grouped into teams and the measurement work is dynamically divided among team members. Currently, Archipelago is made of more than 50 monitors (globally distributed among commercial and research networks) grouped in three teams. The parallelization allows one to obtain `traceroute` data from all routed /24's in about two or three days. `traceroute` performed by Archipelago are made using *scamper* [30], a modern `traceroute` implementation.

RIPE NCC's *Test Traffic Measurement* (TTM) [31] 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[2], packet loss, and bandwidth. Measurements have been performed approximately once every ten minutes, starting in October 2002.

NLANR's Active Measurement Project (AMP) [32] performed active measurements connected by high performance IPv4 networks. 150 AMP monitors were deployed and take site-to-site measurements, mainly throughout the United States. Like RIPE NCC TTM, NLANR AMP avoided probing outside its own network. In addition to `traceroute`, AMP measured RTT, packet loss, and throughput. An IPv6 version of AMP performed measurements between eleven sites. AMP data collection ceased in early September 2006.

The *Distributed Internet MEasurements and Simulations* (DIMES) [33] system is a measurement infrastructure that achieves a large scale by following the model of SETI@home [34]. 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.

Atlas [35] 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., it performs `traceroute` on IPv6 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. To initiate the discovering process, Atlas relies on probing paths among a set of known addresses called *seeds*. The seeds are derived from the information in the 6Bone registry, 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*, i.e., 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. Note that others tools for discovering the IPv6 network are available, such as *Dolphin* [36], scamper [30], Archipelago [28], and SRPS [37].

iPlane [38] is a service providing Internet path performance predictions by building an annotated map of the Internet. Measurement points are deployed on the PlanetLab testbed and the network is daily probed (`traceroute` probing). The obtained data is then postprocessed in order to provide finer grained information, such as router level topology (see Sec. 3 for details on how to build router level maps from `traceroute` data), IP-to-AS mapping, IP-to-PoP mapping, bandwidth estimation, etc.

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

Discarte [39] is an example of a new scalable probing technique that relies on existing techniques. Indeed, Discarte extends `traceroute`-like probing by using the *Record Route* option defined in the IP header [40]. This option is a way to record the route of an IP packet. If a router detects the Record Route option in the IP packet received and this router enables this option, the router must record in the IP header the address it uses for forwarding the packets. General usability of IP options has been investigated by Fonseca et al. [41]. They found that half of the paths drops packets with IP options but those drops occur mainly at the edge and are done by a minority of ASes.

Using the Record Route option within `traceroute` comes with several advantages, one of them being the ability to gather multiple IP interfaces of a router (and, thus, perform alias resolution) without additional probing. However, it has the drawbacks that this option is length limited (only nine IP addresses can be inserted in this IP option) and is not broadly supported by routers. In addition, Discarte uses a logical inference and constraint solving technique to merge Record Route and `traceroute` data. Hynetd [42] also uses Record Route to improve the discovering process.

Gulliver [43] is a measurement platform (currently more than 50 monitors) aiming at observing the behavior of the Internet from all over the world. Monitors are performing DNS measurements, as well as `traceroute` exploration.

2.3 Limitations

If `traceroute` is the most used tool for discovering the IP interface level of the Internet topology, it suffers from several limitations.

First, `traceroute` is *routing dependent*. This means that, when tracerouting, we are only able to observe what the Internet accepts to reveal. Thus, for instance, backup links are unlikely to be traversed by `traceroute`. The number of probing monitors and `traceroute` destinations have been subject to intensive works [44–46], in particular how the number of monitors and destinations can can increase the quantity of topology information collected. If obtaining a list of `traceroute` destinations can be straightforward (it is enough to pick an IP address in each routed /24, for instance), obtaining probing monitors might not be that easy. DIMES solves this issue by proposing a "community-oriented" solution, i.e., the tool is deployed based on the community goodwill. Chen et al. suggest a measurement platform that scales with the growing Internet [47]. They embed a `traceroute` utility into a popular peer-to-peer (P2P) system and traceroutes are performed each time a P2P client is up and running. Doing this way, Chen et al. were able to probe from more than 900,000 IP addresses during the data collection period (one year and a half). Note that it can be difficult to perform reliable measurements since the P2P nodes can go down at any time without warning. Speeding up the probing process is also supposed to improve the topology vision because network dynamics could be better captured [48, 49].

Second, `traceroute` probes are subject to *load balancing*, leading to the inference of false links between IP interfaces. This drawback is explored in Sec. 2.3.1.

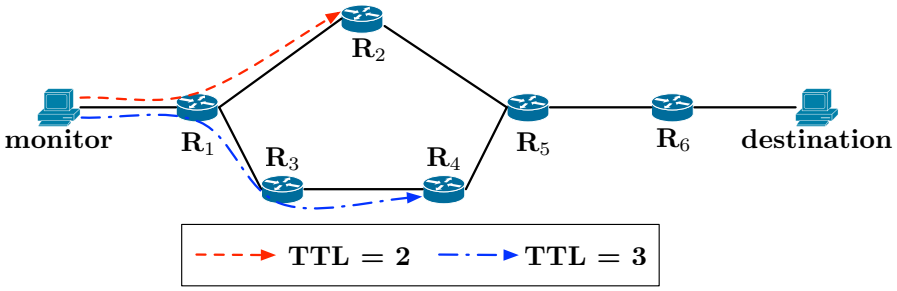


Fig. 3. Effect of load-balancing on `traceroute` exploration

Third, `traceroute` consumes a lot of network ressources by repeatedly discovering the same interfaces and links. This problem is known as *redundancy* and is the focus of Sec. 2.3.2.

Fourth, it is believed that MPLS tunnels obscure topology information and `traceroute` traversing a tunnel cannot reveal the tunnel content. This problem is known as *hidden routers* as MPLS hides topology information to `traceroute`. We focus on this limitation in Sec. 2.3.3.

Fifth, `traceroute` is *unidirectional*. If we `traceroute` from A to B, it is likely the path inferred will differ from the *reverse path*, i.e., the one from B to A. We discuss this issue in Sec. 2.3.4.

Finally, not all routers respond to `traceroute` probes. Such routers are called *anonymous routers* and imply holes in the inferred paths between sources and destinations. We discuss anonymous routers in Sec. 2.3.5.

Additional `traceroute` anomalies are also described by Augustin et al. [50] and Viger et al. [51]. Finally, note that the problem of *third-party* addresses [52, 53] will be addressed in Sec. 5.5 as it only concerns AS inference from traceroute data.

2.3.1 Load Balancing

Load balancing for Internet paths is extensively used by ISPs for ensuring reliability of their networks and improve ressource utilisation. This can be done through intra-domain routing protocols, such as OSPF [54] and IS-IS [55], supporting *equal cost multipath* (ECMP) [56], a routing strategy in which the next-hop to a given destination can occur over multiple paths. Load balancing can also be considered in the context of multihomed ISPs, for selecting which provider will receive which packet [57]. A router performing load balancing is called *load balancer*. Fig. 3 illustrates a load balancing path between a source and a destination. Router R_1 acts as load balancer as the path to "destination" can go through R_2 or through $R_3 \rightarrow R_4$.

There are three types of load balancing [58, 59]: *per packet*, *per flow*, and *per destination*. With per flow load balancing, a flow is assigned to each packet through information in the packet header, and the load balancer forwards all packets belonging to the same flow to the same interface. A flow identifier can be built, for instance, based on the classic five-tuple: source IP address,

destination IP address, source port, destination port, and protocol. Others IP field, such as ToS, can also be considered for the flow identifier. The per packet load balancing only ensures an even load on the links, making no attempt to maintain a flow. Finally, the per destination load balancing forwards all packets with the same destination to the same interface of the load balancer and can be seen as equivalent to standard routing.

The presence of load balancing implies that there are multiple routes. The historical `traceroute`, as developed by Van Jacobson (see Sec. 2.1), is sensitive to load balancing. This can lead to the inference of false links between routers, as illustrated in Fig. 3. If router R_1 is, for instance, a per packet load balancer, a `traceroute` packet with TTL = 2 (dashed link on Fig. 3) could reach router R_2 but, with the packet with TTL = 3 (dash-dotted link on Fig. 3), it could reach router R_4, inferring so a link between R_2 and R_3. However, such a link does not exist.

Luckie et al. evaluated the inaccuracies induced from false links inferences [60]. This evaluation has been done at two levels: macroscopic probing (i.e., tracerouting towards a very large set of destinations, in the fashion of Archipelago – see Sec. 2.2) and ISP probing (i.e., in the fashion of Rocketfuel [61]). Regarding macroscopic probing, the impact of false links seems to be minor while, for the latter, two third of the links were suspicious.

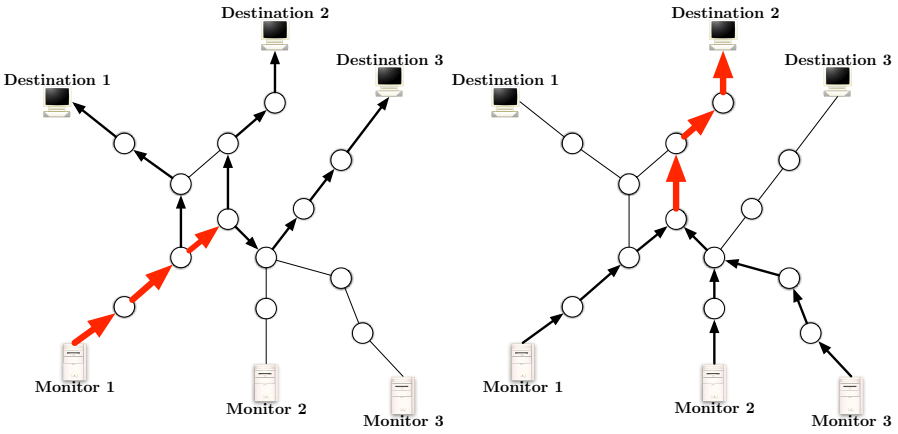
A new `traceroute`, called *Paris traceroute*, has been developed by Augustin et al. in order to take into account load balancing and to avoid false link inference [50]. The idea behind Paris traceroute is to control the `traceroute` packet header fields so that all probes towards a destination follow the same path. This allows one to avoid the negative effects of per flow load balancing on `traceroute` exploration. Per packet load balancing is much more difficult to mitigate due to its random nature.

Based on Paris traceroute, an algorithm, *Multipath Detection Algorithm* (MDA), for detecting and listing all routes induced by a load balancer has been proposed [62–64]. The deployment of this algorithm shows that 39% of the source-destination pairs traverse a per flow load balancer and 70% a per destination load balancer.

2.3.2 Redundancy

Donnet et al. evaluate how `traceroute` probing involves duplicate efforts [65, 66]. This is of high importance as `traceroutes` emanating from a large number of monitors and converging on selected targets can easily appear as a distributed denial-of-service (DDoS) attack. Whether or not it triggers alarms, it is not desirable for a measurement system to consume undue network resources. Duplicated effort in such systems takes two forms: measurements made by an individual monitor that replicate its own work, and measurements made by multiple monitors that replicate each other's work. Donnet et al. term the first *intra-monitor redundancy* and the second *inter-monitor redundancy*.

On one hand, intra-monitor (shown in Fig. 4(a) with very thick arrows illustrating redundant portion of the explored graph) redundancy occurs in the context of the tree-like graph that is generated when all `traceroutes` originate



(a) Intra-monitor redundancy (b) Inter-monitor redundancy

Fig. 4. `traceroute` redundancy [65]

at a single point. Since there are fewer interfaces closer to the monitor, those interfaces will tend to be visited more frequently. In the extreme case, if there is a single gateway router between the monitor and the rest of the Internet, a single IP address belonging to that router should show up in every one of the `traceroutes`. On the other hand, inter-monitor redundancy (shown in Fig. 4(b) with very thick arrows illustrating redundant portion of the explored graph) occurs when multiple monitors visit the same interface.

Solutions to probing redundancy have been explored. First, *Scriptroute* [67]'s *Reverse Path Tree* (RPT) discovery tool is used to avoid the network overloading when multiple monitors probe towards a given 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 (see Fig. 4(b)). 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. Scriptroute thus can avoid inter-monitor redundancy.

Second, *Rocketfuel* [61] is a tool for mapping router-level ISP topologies. For reducing the number of measurements required, Rocketfuel makes use of *ingress reduction* and *egress reduction* heuristics. 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 is based on the observation that probes to multiple destinations may leave the target ISP at the same node.

Finally, *Doubletree* [65, 66] takes advantage of the tree-like structure of routes in the context of probing, as illustrated in Fig. 4. Routes leading out from a monitor towards multiple destinations form a tree-like structure rooted at the monitor (Fig. 4(a)). Similarly, routes converging towards a destination from multiple monitors form a tree-like structure, but rooted at the destination (Fig. 4(b)). A

monitor probes hop 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* (monitor-rooted tree – decreasing TTLs) and *forwards* (destination-rooted tree – increasing TTLs).

For both backwards and forwards probing, Doubletree uses 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 global stop set if a monitor receives a packet from the interface in reply to a probe sent towards the destination address.

A Doubletree monitor starts probing for a destination at some number of hops h from itself. It will probe forwards at $h + 1$, $h + 2$, etc., adding to the global stop set at each hop, until it encounters either the destination or a member of the global stop set. It will then probe backwards at $h - 1$, $h - 2$, etc., adding to both the local and global stop sets at each hop, until it either has reached the distance of one hop or it encounters a member of the local stop set. It then proceeds to probe for the next destination. When it has completed probing for all destinations, the global stop set is communicated to the next monitor. Note that in the special case where there is no response at distance h, the distance is halved, and halved again until there is a reply, and probing continues forwards and backwards from that point. Each monitor sets its own value for h in terms of the probability p that a probe sent h hops towards a randomly selected destination will actually hit that destination. Doubletree's efficiency has been largely explored [68–71].

Note that Rocketfuel's ingress and egress reduction heuristics are similar to Doubletree'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.

2.3.3 Hidden Routers

Multiprotocol Label Switching (MPLS) [72] was designed to reduce the time required to make forwarding decisions. It is now deployed to provide additional virtual private network (VPN) services [73] and traffic engineering capability [74, 75]. To accomplish this, an IP router inserts one or more 32-bit *label stack entries* (LSE) into a packet, before the IP header, that determines the forwarding actions made by subsequent MPLS *Label Switching Routers* (LSRs) in the network. A series of LSRs connected together form a *Label Switched Path* (LSP). MPLS networks are deployed on IP routers that use a label distribution protocol [76, 77].

In an MPLS network, packets are forwarded using an exact match lookup of a 20-bit label found in the LSE. An MPLS LSE also has a time-to-live (LSE-TTL) field and a type-of-service field. At each MPLS hop, the label of the incoming packet is replaced by a corresponding outgoing label found in an MPLS switching table. The MPLS forwarding engine is lighter than the IP forwarding engine

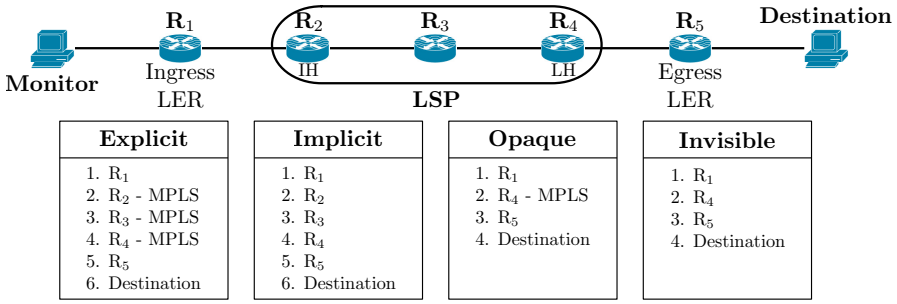


Fig. 5. Taxonomy of MPLS tunnel configurations and corresponding `traceroute` behaviors [19]

because finding an exact match for a label is simpler than finding the longest matching prefix for an IP address.

MPLS routers may send ICMP `time-exceeded` messages when the LSE-TTL expires. In order to debug networks where MPLS is deployed, routers may also implement RFC 4950 [78], an extension to ICMP that allows a router to embed an MPLS label stack in an ICMP `time-exceeded` message. The router simply quotes the MPLS label stack of the probe in the ICMP `time-exceeded` message. RFC4950 is particularly useful to operators as it allows them to verify the correctness of their MPLS tunnels and traffic engineering policy. This extension mechanism has been implemented by router manufacturers since 1999 [79], and is displayed by modified versions of `traceroute` [21] that report the label stack returned by each hop in addition to RTT values currently displayed. If the first MPLS router of an LSP (the *Ingress* Label Edge Router - LER) copies the IP-TTL value to the LSE-TTL field rather than setting the LSE-TTL to an arbitrary value such as 255, LSRs along the LSP will reveal themselves via ICMP messages even if they do not implement RFC4950. Operators configure this action using the `ttl-propagate` option provided by the router manufacturer.

These two "MPLS transparency" features – RFC 4950 functionality and the `ttl-propagate` option – increase the observability of otherwise opaque MPLS tunnels during IP-level topology discovery based on `traceroute`. Unfortunately, lack of universal deployment of these two features (ingress LERs that do not enable the `ttl-propagate` option, and LSRs that do not support the RFC4950 ICMP extensions) means that current `traceroute`-based inference methods can cause false router-level links to be inferred and underestimates MPLS deployment in the Internet.

Based on those two MPLS transparency features, Donnet et al. [19] have proposed an MPLS taxonomy made of four classes. Fig. 5 illustrates the four classes that are

– *explicit* tunnels: both `ttl-propagate` and RFC4950 are enabled. The tunnel and its internal structure are visible. Each hop within the LSP is flagged as such (as illustrated with "MPLS" in Fig. 5). Explicit tunnels have been

extensively studied by Sommers et al. [18] and are the most represented MPLS tunnels: roughly, 30% of the path traverse, at least, one MPLS explicit tunnel. This proportion of explicit tunnels has also been observed by other studies [19, 80].
- *implicit* tunnels: the router that pushes the MPLS label enables the `ttl-propagate` option but LSRs do not implement RFC4950. In this case, while the internal IP structure of the tunnel is visible, its existence as an MPLS tunnel is not revealed. Donnet et al. have proposed signature-based techniques for revealing such tunnels and demonstrated that implicit tunnels are three times less prevalent than explicit ones.
- *opaque* tunnels: LSRs implement RFC4950 but the ingress LER does not enable the `ttl-propagate` option. Only the router that pops the MPLS label reveals a LSE and the internal structure of the LSP is hidden. In Fig. 5 the opaque tunnel hides two LSRs (R_2 and R_3), allowing an erroneous link to be inferred between R_1 and R_4. Donnet et al. have also proposed a technique for inferring the length of opaque tunnels (i.e., the number of hidden routers in the LSP). They also suggested that opaque tunnels are very infrequent.
- *invisible* tunnels: the ingress LER does not enable the `ttl-propagate` option and RFC4950 is not implemented by the router popping the MPLS label. In Fig. 5, two IP hops are hidden and the last router of the MPLS path does not flag itself as part of an LSP. Again, a link between R_1 and R_4 is erroneously inferred. Up to now, there is no technique for revealing and quantifying invisible tunnels.

Based on observation made by Sommers et al. [18] and Donnet et al. [19], MPLS is a frequent feature in the Internet but is not a brake to Internet topology discovery (thanks to RFC 4950 and `ttl-propagate` option). Only opaque and invisible tunnels are a problem but it seems they are infrequent. If for the moment invisible tunnels cannot be revealed and quantified, the actual impact of opaque tunnels on topology discovery must still be evaluated.

2.3.4 Unidirectionality

`traceroute`, as defined in Sec. 2.1, is unidirectional. It means that it is only able to capture the path from the `traceroute` source towards a given destination but without providing any information on the path from the destination to the `traceroute` source itself (i.e., the *reverse path*). Said differently, `traceroute` gives the path from the source to anywhere, but not the path from anywhere to the `traceroute` source. As routing is asymmetric [82], both paths (i.e., one-way and reverse) are likely to be different. This situation is illustrated in Fig. 6 in which the `traceroute` from the source to a given web server gives a path traversing AS_9, AS_1, AS_4, and AS_7. On the contrary, the path from the web server to the source traverses AS_7, AS_6, AS_8, AS_{12}, and AS_9.

Reverse `traceroute` [81] has been proposed to overcome this fundamental drawback. Reverse traceroute builds the reverse path incrementally, using a set of controlled vantage points and several measurement techniques. First, vantage points are used to measure the path from themselves towards the `traceroute`

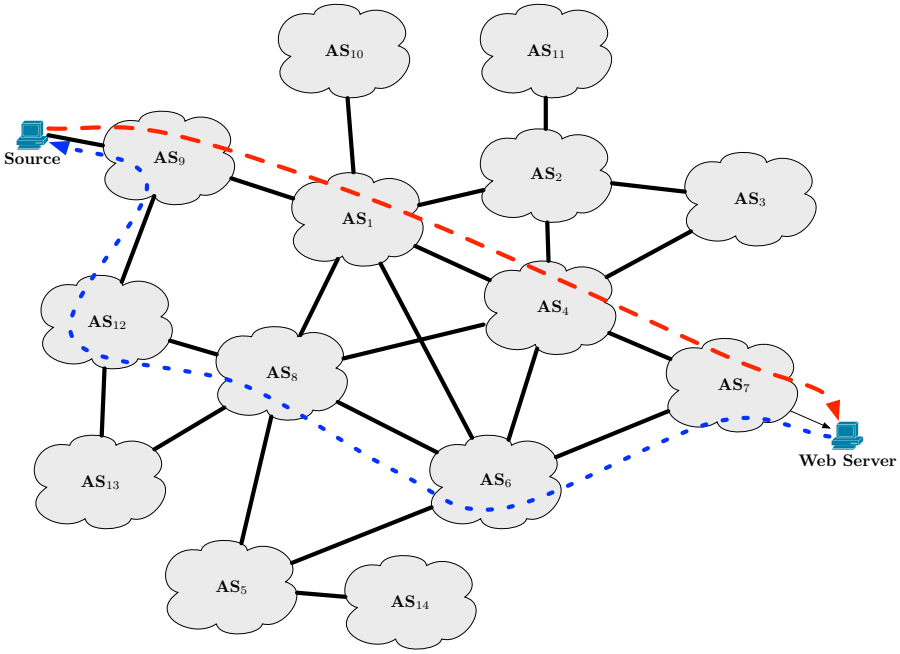


Fig. 6. `traceroute` unidirectionality and reverse path

source. Tree-like structure of routes is considered for avoiding probing redundancy. The set of paths collected is used as bootstrap for incrementally building the reverse path, starting from the `traceroute` destination and hop-by-hop back to the `traceroute` source. This incremental process is stopped when encountering a router belonging to the set of routes collected at first. This process can be seen as somewhat equivalent to Doubletree's local stop set (see Sec. 2.3.2).

Reverse traceroute considers three measurement mechanisms for building the reverse path. First, globally, Internet routing is destination-based. This means that Reverse traceroute should be able to capture the reverse path one hop at a time and, next, glue all hops together. Second, several IP options, such as Record Route [40] and IP timestamp [83], are used to identify the various hops along the path. Finally, *IP spoofing* is used to overcome limitations in IP timestamp support. IP spoofing, first described by Morris [84] and deeply discussed by Bellovin [85], is one of the common tools used by hackers to perform attacks. It allows the attacker to hide his identity by forging the source IP address of packets. Instead of carrying the source IP address of the machine the packet comes from, it contains an arbitrary IP address that is selected either randomly or intentionally. Despite various techniques for avoiding IP spoofing (see, for instance, ingress filtering [86, 87], hop count [88], probabilistic marking [89], or hash-based traceback mechanisms [90]), a large-scale study has shown that IP spoofing is still widely possible [91]. Based on measurements distributed throughout the

world, Beverly and Bauer find that approximately one-quarter of the observed addresses, netblocks, and autonomous systems (AS) still permit full or partial spoofing [91].

2.3.5 Anonymous Routers

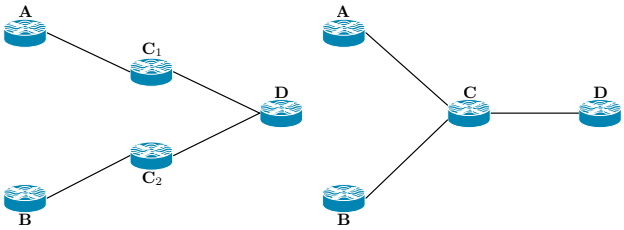
Unfortunately, the `traceroute` behavior explained in Sec. 2.1 is the ideal case. A router along the path might not reply to probes. In order to avoid waiting an infinite time for the ICMP reply, the `traceroute` monitor activates a timer when it launches the probe. If the timer expires and no reply was received, then, for that TTL, the machine is considered as *non-responding* and the router is flagged as "*" in the `traceroute` output. Such a router is also called an *anonymous router*. The task of identifying all "*" belonging to the same router is known as *anonymous router resolution*. It has been shown that this process is NP-complete [92].

Gunes and Sarac identify five reasons for a router for being anonymous [93]:

1. The router is configured to ignore all `traceroute` requests, i.e., it nevers sends back an ICMP *ttl exceeded* packet.
2. The router applies ICMP rate limiting and is anonymous if the incoming `traceroute` queries rate is above the preset threshold.
3. The router is configured to ignore `traceroute` queries when it is congested. Otherwise, it responds to queries.
4. A border router might be configured to filter all outgoing ICMP packets coming from its domain. As a consequence, all routers belonging to this domain are anonymous.
5. The router that replies with a non-publicly routable IP address [94] should be considered as anonymous as a given non-publicly routable IP address can be used by several routers (i.e., there is no uniqueness guarantee).

It is worth to notice that anonymous router resolution cannot be done, generally, during the probing time. This process is, typically, done after the data has been collected. We can thus see anonymous router resolution as a passive process. Historically, simple solutions to anonymous routers have been proposed. For instance, Cheswick et al. [95] stop tracerouting towards a destination once an anonymous router is encountered. This approach has the drawback of potentially missing useful information. Broido and claffy [96] replace anonymous routers with arcs (e.g., the route portion $R_i \rightarrow * \rightarrow R_j$ is replaced by $R_i \rightarrow R_j$) or with a unique identifier in order to consider each anonymous router as a unique node in the topology. Such a solution can lead to inaccuracies in the resulting topology. Finally, Bilir et al. [97] compress successive anonymous routers between two nodes into a single identifier. This solution has a limited scope.

More recent solutions to anonymous router resolution are based on optimization problem [92], on heuristics on link delays or neighbor matching [93], and on graph data mining [98].



(a) `traceroute` paths (b) Router aggregation

Fig. 7. Alias resolution principle

3 Router Level Discovery

The router level is the second level of the Internet topology. Nodes in the graph are routers themselves, meaning that all IP interfaces of a router collected with `traceroute` have been aggregated into a single identifier. This aggregation, historically, is made using *alias resolution* (Sec. 3.1). This alias resolution is either made at the same time of probing IP interfaces, requiring additional probing to `traceroute` (active techniques), or after the data collection by analyzing the resulting graph (passive techniques).

Recently, a new technique has been developed and is based on *IGMP probing* (Sec. 3.2). This technique has the advantage of collecting, in a single probe, all multicast interfaces of a router. This means that alias resolution is not anymore required.

3.1 Alias Resolution

The router level topology 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. 7. As explained in Sec. 2.1, `traceroute` lists interfaces addresses of a path and identifies, in our example, interfaces `A`, `B`, $\texttt{C}_1$, $\texttt{C}_2$ and `D` (Fig. 7(a)). Alias resolution clusters all interfaces of a router into a single value to reveal the true topology. As shown in Fig. 7(b), interfaces $\texttt{C}_1$ and $\texttt{C}_2$ are aliases. Based on synthetic topologies, Gunes and Sarac show that the accuracy of alias resolution has an important effect on the observed topological characteristics such as, for instance, the number of nodes and edges or the average node degree [99]. In this section, we describe the currently existing approaches for alias resolution.

3.1.1 Active Methods

Active methods for alias resolution are based on several techniques: address based method, IP identifier based method, DNS, Record Route IP option, and timestamps. We describe these techniques in this section.

The *address based method* is described in RFC 1122 [100]. The principle is simple: the source sends a UDP probe with a high port number to the router

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* [101] and Mercator [2].

The *IP identifier based method* has been implemented in *Ally*, *Rocketfuel*'s alias resolution component [61]. Ally is based on the ID field of the IP header, a 16-bit field used to identify the fragments of one datagram from those of another [40]. The ID value is supposed to be unique for a given (source, destination) pair and protocol during the time the packet could be alive in the network. The basic idea of the IP identifier based method 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, like the address based method, works only if a router responds to probes. Further, it is network resource greedy as it requires $O(n^2)$ probes to infer aliases among n IP addresses.

To overcome this network resource limitation, Bender et al. introduce *RadarGun* [102]. Basically, RadarGun models the IP ID as a counter over time and infers the rate at which the counter increases. This rate is named the *velocity* of a router and is typically close to a straight line. The distance between pairs of lines is computed and the pairs with a small distance are labeled as aliases.

MIDAR (Monotonic ID-Based Alias Resolution) [103] is designed for performing alias resolution, using the IP identifier technique, for a very large-scale of targets (on the order of several millions). The comparison of IP identifiers is, here, based on monotonicity instead of proximity. The scalability is achieved by considering multiple vantage points, multiple probing methods, and a sliding-window probe scheduling.

The *DNS based 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. *AROMA* [104] also combines DNS based method and Ally's technique. However, it has been shown that DNS can introduce errors due to misnaming, leading so to bad alias resolution [105].

The *TTL-limited with record route option method* has been proposed by Sherwood and Spring [39]. The idea is to perform a standard `traceroute` with the *Record Route* (RR) IP option enabled [40]. 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 ingoing interface. This technique works only in networks where routers support the

RR option, which is not necessarily the case in modern networks. This solution is computationally expensive [106].

Palm Tree [107] takes a list of IP addresses from a target network and aims at identifying IP aliases among IP address of this target network. The alias resolution technique applied by Palm Tree is based on common practice for assigning IP addresses [108]. Palm Tree must be seen as a complementary tool to existing techniques.

The last technique, proposed by Sherry et al. [109] and implemented in *motu* [110], is based on the *IP timestamp option* [83]. Originally, the timestamp option has been introduced for measuring the one-way delay of Internet links. A router receiving an IP packet with the IP timestamp option is supposed to provide, in the option, a timestamp consisting of milliseconds since midnight UTC. There are several sub-types of IP timestamp option: (*i*), "timestamp only" for which the router writes the timestamp in the option; (*ii*), "timestamp with ID" for which the router writes the timestamp in the option preceded with its IP address; (*iii*), "prespecified timestamp", in which IP addresses of routes are prespecified in the IP packet. A router inserts its timestamp in the option only if its IP address has been prespecified by the packet sender.

For testing if A and B are aliases, Sherry et al. send multiple ICMP *Echo Request* probes with the prespecified timestamp option enabled. If A and B are aliases, the router should record timestamps for both A and B with consistent values. If timestamps in the replies are consistent, further investigations are made to ensure both IP addresses are aliases.

The large-scale applicability of the IP timestamp option has been evaluated and it has been shown that using this option is only effective for 12.9% of the destinations [111]. Note that it seems that using the IP timestamp option is a little bit more effective in the context of reverse traceroute (see Sec. 2.3.4).

Routers behaviors regarding active alias resolution have been recently analyzed [112]. Direct probing based on ICMP packets with IP identification provides the best identification ratio.

3.1.2 Passive Methods

On the contrary to active methods, passive methods do not require additional probing. The alias resolution is done offline, after the `traceroute` data has been collected. Those passive techniques are based either on graph methods, either on IP addresses and subnets assignments.

The *graph based method* extracts from `traceroute` outputs a graph of linked IP addresses in order to infer likely and unlikely aliases [113]. It is based on two assumptions: (1) if two IP addresses precede a common successor IP address, then they are likely to be alias, and (2) 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 second method is the *Analytical Alias Resolver* (AAR) introduced by Gunes and Sarac [114]. Given a set of path traces, AAR utilizes the common IP address assignment scheme to infer IP aliases within the collected path traces.

However, AAR assumes point-to-point links to resolve IP aliases on a given pair of path traces between two vantage points and completely ignores multi-access links.

The *Analytic and Probe-Based Alias Resolver* (APAR) [115] is an extension of AAR to overcome its limitations. It uses a set of inference rules, based on identifying the subnets linking routers and then aligning `traceroute` paths using those subnets. *kapar* [106] is an optimized implementation of APAR that overcomes APAR's scalability issues.

3.2 IGMP Probing

Historical alias resolution techniques, discussed in Sec. 3.1, come with inherent limitations. Indeed, as they are based on `traceroute` data, they naturally inherits from `traceroute` limitations (see Sec. 2.3) and biases. Further, additional probing or analysis is also biased. This can lead to a high proportion of false aliases, giving so an inaccurate vision of the router level topology.

Recently, IGMP probing has been considered for collecting router level data [20]. Although it is limited to multicast-enabled routers and unfiltered networks (Sec. 3.2.4), IGMP probing comes with the advantage of silently collecting all multicast interfaces of a router in a single probe. IGMP probing is made possible with `mrinfo` (Sec. 3.2.1), `mrinfo-rec` (Sec. 3.2.2), and MERLIN (Sec. 3.2.3).

3.2.1 `mrinfo`

`mrinfo` [116] messages use the Internet Group Management Protocol (IGMP [117]). IGMP was initially designed to allow hosts to report their active multicast groups to a multicast router on their local area network (LAN). Most IGMP messages are sent with a `time_to_live` of 1. However, the Distance Vector Multicast Routing Protocol, DVMRP, has defined two special types of IGMP messages that can be used to monitor routers [118]. Although current IPv4 multicast routers do not use DVMRP anymore, they still support these special DVMRP messages. Upon reception of an IGMP `ASK_NEIGHBORS` message, an IPv4 multicast router replies by sending an IGMP `NEIGHBORS_REPLY` message that lists all its multicast enabled local interfaces[3] with some information about their state. Cisco and Juniper routers also report in the IGMP `NEIGHBORS_REPLY` message the version of their operating system. Fig. 8 shows an example of the usage of `mrinfo` to query the router R_2, `1.1.0.2` being the responding interface of R_2. `mrinfo` reports that this router is directly connected to R_0 (through interface `1.1.0.1`). We can also notice that R_2 is connected to routers R_5 and R_6 through an L2 network (labeled "switch" in Fig. 8) because interface `1.1.2.3` appears twice in the `mrinfo` reply (see bold text in Fig. 8). Finally, `mrinfo` reports that interface `1.1.3.1` has no multicast neighbor because the right IP address is equal to 0.0.0.0 (or is directly connected to a LAN,

[3] It has been reported that a router may reply to a `ASK_NEIGHBORS` message through one of its purely unicast interface [119]. In such a case, the set of collected interfaces is not only multicast.

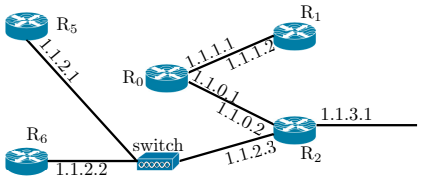


Fig. 8. `mrinfo` example [20]

as indicated by the "leaf" keyword). All this information is obtained by sending a single IGMP message. In practice, `mrinfo` provides information similar to the output of a `show` command on the router's command line interface.

3.2.2 `mrinfo-rec`

Mérindol et al.'s approach in probing the network with `mrinfo` is recursive and is called `mrinfo-rec` [20]. Initially, `mrinfo-rec` is fed with a single IP address corresponding to the first router attached to the `mrinfo-rec` vantage point. `mrinfo-rec` probes this router and recursively applies its probing mechanism on all the collected IP addresses. These recursive queries stop at unresponsive routers or when all discovered routers have been queried. The same process is run every day. It is worth to notice that an address not replying to an `mrinfo` probe during a given day will not be queried the days after except if it appears again in a list of captured addresses.

To illustrate this behavior, let us apply it on the topology depicted in Fig. 8. `mrinfo-rec` receives, as input, an IP address belonging to router R_0. From R_0, `mrinfo-rec` collects a set of neighbor IP addresses, i.e., $\{1.1.1.2, 1.1.0.2\}$. For all IP addresses in this set that were not previously probed, `mrinfo-rec` sends an IGMP `ASK_NEIGHBORS` message and, if the probed router replied, it again runs through the set of neighbor IP addresses collected.

3.2.3 `MERLIN`

Initial implementations of `mrinfo` and `mrinfo-rec` suffer from several issues and limitations [119]. First, there is a lack of support for IGMP-fragmented `NEIGHBORS_REPLY` messages as `mrinfo` is unable to deal with multiple received packets. It only processes the first packets (there is no continuation flag forcing the wait for the remaining fragments). Mérindol et al. have observed this behavior on large degree CISCO routers. Second, `mrinfo` is unable to multiplex IGMP-based measurements. The initial version of `mrinfo` sends its IGMP query, then waits for a possible reply during a given time. Possibly it performs several retries if no response has been collected within the previous time frame. Further, IGMP does not consider port and query numbers to multiplex incoming/outgoing connections. This leads to scalability issues when performing large-scale IGMP probing.

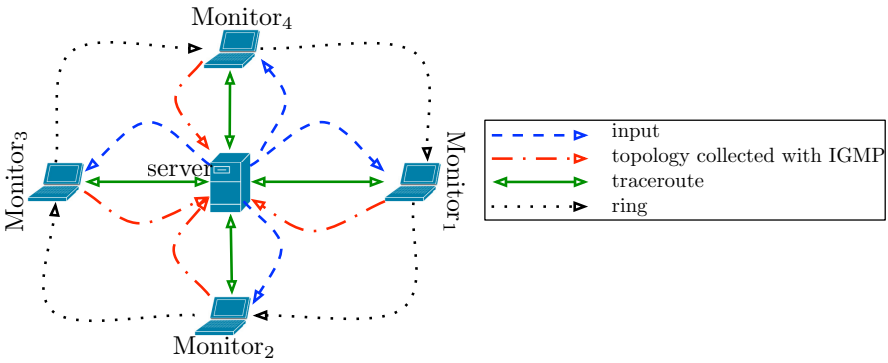


Fig. 9. MERLIN global overview [120]

MERLIN [119] has been introduced to overcome these limitations. MERLIN decouples the sending and receiving processes in order to avoid the use of timers between queries and replies and improve the probing efficiency. Furthermore, all replies having the same source IP address are considered as part of a largest message in order to re-assemble IGMP fragmented packets of a given router. If `mrinfo` and `mrinfo-rec` were probing the entire Internet, MERLIN has been designed to focus on ASes.

If MERLIN is still based on recursive IGMP probing, it also improves `mrinfo` and `mrinfo-rec` by probing from several vantage points, each one being managed by a central server [120], in order to increase the exploration coverage while limiting the probing redundancy. Fig. 9 illustrates how MERLIN vantage points are organized around the central server. The "input" is the initial set of destinations provided to each vantage points. The MERLIN vantage points are organized in a ring, a vantage point probing after its predecessor in the ring. In addition, MERLIN makes use of `traceroute` to discover active addresses (typically in targeted ASes) in order to circumvent the recursion limitations. It has been shown that this multiple vantage points probing increases the amount of information collected by the IGMP probing [119].

3.2.4 IGMP Probing Limitations

The first limitation of IGMP probing is inherent to the technique itself: only multicast enabled network can be probed. This limits thus the scale of the topology that can be collected.

When probing a multicast enabled AS with IGMP probing, one expects obtaining its complete *backbone* as it should be entirely multicast to ensure the correct multicast tree establishment by the PIM multicast routing protocol [121]. By multicast backbone, one means the AS areas where links and routers providing connectivity to non-multicast customers or peers are pruned. Unfortunately, some routers do not reply to IGMP probes sent by MERLIN, leading

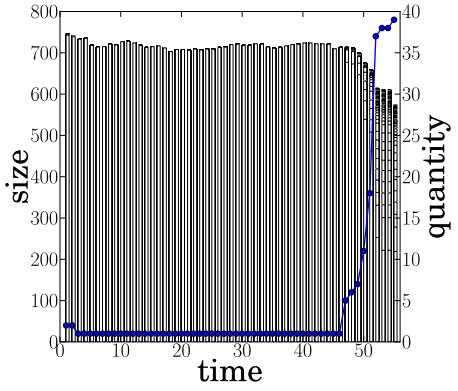


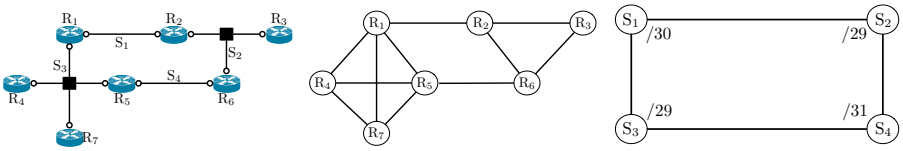
Fig. 10. Connected components evolution over time (AS1239) [120]

to an anonymous behavior that is similar to the one observed with `traceroute` (see Sec. 2.3.5). This phenomenon is called *IGMP filtering*. As a consequence, the topology obtained using solely IGMP probing is incomplete and disconnected: the collected IGMP graph exhibits a set of disjoint components. Fig. 10 illustrates this problem on the Sprint Tier-1 AS.[4] The horizontal axis shows IGMP probing snapshots over 56 weeks (one snapshot per week) between May 2004 and December 2008. The left vertical axis provides the size of each connected component and must be read in conjunction with stacked bars. The right vertical axis gives the quantity of connected component plotted in a stacked bar and must be read with the line. While at the beginning, one was able to capture a single large connected component (more than 700 connected nodes), the number of connected components starts exploding in 2008: up to 38 connected components, a lot of them being made of only 2 nodes. The explanation of this degradation is the progressive introduction of IGMP filtering in the network. Indeed, the number of connected components increases with the number of non-responding routers.

IGMP filtering is of two kinds [122]: some multicast routers do not reply to IGMP probes (*local filtering*) while some others do not forward IGMP messages (*transit filtering*). While the second problem can be reduced with the use of multiple vantage points in a cooperative distributed platform [120], the first one is more challenging as it impacts the collected topologies. Indeed, multicast routers that do not respond to IGMP probes may divide the resulting collected multicast graph into disjoint components.

Merlin tries to limits the effect of local filtering by applying `traceroute` and alias resolution for glueing together disconnected components [122]. This reconnection procedure is achieved by the central server, once IGMP components have been collected.

[4] The same phenomenon can be observed in others ASes (Tier-1, Transit, and Stub).



(a) Example topology (b) Router graph (c) Subnet graph

Fig. 11. How a network can be represented as a router graph and a subnet graph [124]

4 Subnet Level Discovery

A *subnetwork* (or, simply, *subnet*) refers to a set of devices that are on the same connection medium and that can communicate directly with each other at the link layer [123].

The subnet level is a way to enrich router level maps with subnet level connection information [6]. Subnet discovery presents some similarities with alias resolution (see Sec. 3.1). Indeed, alias resolution follows the goal of aggregating IP addresses (appearing in various traces) of a router into a single identifier. Similarly, subnet detection aims at identifying multiple links (appearing to be separate) and at combining them to represent their single hop connection medium (point-to-point or multi-access) [8].

Fig. 11 illustrates the concept of subnet. Fig. 11(a) provides the groundtruth topology for our example. This topology is made of seven routers, some of them being connected through layer-2 devices (the black square – see for instance the connection between R_4 and R_5). Fig. 11(b) gives the router level graph view of the topology. Finally, Fig. 11(c) aggregates all routers (and layer-2 devices) belonging to the same subnetwork into a single identifier and connects a subnet to others if they communicate with each other.

4.1 Inference Techniques

In the fashion of alias resolution (see Sec. 3.1), subnet inference can be divided into two kinds of methods: passive and active techniques.

4.1.1 Active Methods

`traceNET` [6] attempts to collect a subnet at each router on the same path. `traceNET` works iteratively and starts by creating a temporary /31 subnet from a given interface. It then grows the subnet by decreasing prefix lengths (i.e., /30, /29, etc.). For each subnet, `traceNET` probes the potential IP addresses within the subnet range to ensure that those IP addresses are assigned to IP interfaces. Note that this decision is taken based on heuristics. If one of the heuristics is not met, the IP address is declared as not belonging to the subnet under exploration. The growing process is then stopped and the subnet is stepped back to its last valid state.

In some sens, traceNET works as traceroute: it is designed to detect subnets on a given path between a source and a destination. traceNET is thus subject to some traceroute limitations, such as routing. In addition, in order to be efficient, traceNET requires several vantage points. *exploreNET* [124], a traceNET extension, has been introduced to mitigate those drawbacks. In particular, exploreNET is able to discover individual subnets rather than subnets on an end-to-end path. exploreNET also presents techniques for sampling subnets in the target domain and their global characteristics (such as mean subnet degree, subnet prefix length distribution, etc.) [7].

4.1.2 Passive Methods

Although it is not its primary goal, IGMP probing (see Sec. 3.2) allows one to detect subnets [14]. The subnet inference requires to post-process the collected data by following three rules:

- *Symmetry rule.* All routers attached to the potential subnet should have the same view. On Fig. 8, router R_2 is connected to the same subnet as R_5 and R_6 through a layer-2 device. When probing R_5 and R_6 with `mrinfo`, R_2 must also appear in their `mrinfo` output.
- *Querier rule.* In a normal case, only one router per layer-2 network must be tagged as the IGMP "querier" (i.e., it won the querier election on the subnet [125]: it has the greatest IP address on the subnet). For instance, on Fig. 8, as interface `1.1.2.3` is tagged as "querier", interfaces `1.1.2.1` of R_5 and `1.1.2.2` of R_6 should not be tagged as such.
- *Subnet mask rule.* The validity of the minimum ask covering all IP addresses in the subnet is verified.

In addition, IGMP probing can provide information on the technology used in the subnet, such as ATM cloud, etc. However, the limit of IGMP probing for revealing subnet is that `mrinfo` is only able to detect subnetworks involving at least three routers.

Gunes and Sarac suggest that subnet inference can be done once data has been collected (using `traceroute`) and alias resolution has been done [8]. IP address assignment practices [126, 127] induce subnet relationships or formations. Candidate subnets are thus formed, from the data, by grouping into a subnet address range address prefix of length $/x$. Smaller subsets ($/x$, $(x+1)$, ..., $/31$) are next recursively created. In the fashion of IGMP probing, a set of rules is defined for verifying candidate subnets:

- *Accuracy rule.* IP addresses in a subnet should appear next to each other each time they appear in the same trace.
- *Distance rule.* IP addresses from a given subnet should be at similar distances to a vantage point.
- *Completeness rule.* Candidate subnets having less than a quarter of their IP addresses present in the data set should be ignored.
- *MaxFit rule.* Candidate subnets that are a subset of a larger one must be ignored after assessing the previous rules.

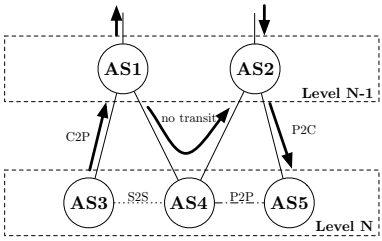


Fig. 12. AS relationships

5 AS Level Discovery

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).[5]

In this section, we describe the AS relationships 5.1) before discussing the induced AS structure (Sec. 5.2). We next describe the various data sources for inferring AS level topology (Sec. 5.3). We describe techniques for inferring AS relationships 5.4 and, finally, discuss their limit (Sec. 5.5).

5.1 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 [20].

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 recent analysis has shown that p2p relationships between adjacent Tier-1 ASes are redundant, i.e., the connections between those ASes involve several physical links [20].

[5] It is worth to notice that, since December 1st, 2006, the AS Number Registry has been expanded to 32 bit-number space [128].

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 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. Fig. 12 illustrates those AS relationships.

These relationships have a major impact on routing in the Internet, as shown by Tangmunarunkit et al. [129]. 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 [130, 131] 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.

5.2 AS Hierarchy

Relationships discussed in Sec. 5.1 suggest the existence of an AS hierarchy [133]. This hierarchy is described in Fig. 13.

Following Subramanian et al. [133] nomenclature, we can distinguish three kinds of AS. First, the *Tier-1* ASes do not have upstream provider of their own. Typically, a Tier-1 has a national or international backbone and there are full p2p connections between them. On Fig. 13, Tier-1 ASes are on the top of the hierarchy and labeled as "National Backbone Operators". There are a few Tier-1 ASes (roughly 12-20), such as UUNET, Sprint, or Level3. Second, the *Tier-2* ASes (or, simply, the *Transit* ASes) provide transit services to downstream providers. A transit AS needs, at least, one provider of its own. The Internet counts a few thousand Transit ASes. Those Transit ASes are located in the middle of the hierarchy in Fig. 13 and labeled as "Regional Access Providers" and "Local Access Providers". Finally, the *Stub* ASes do not provide transit services to others. They are connected to one or more upstream providers. Stub ASes constitute the vast majority of ASes (i.e., 85-90%). They are located at the bottom of the hierarchy in Fig. 13 and labeled as "Customer IP Networks".

5.3 Data Sources

Two sources of AS level topology data are available: *Internet registries* and *BGP routing information*. This section describe these two sources [134] along with their advantages and limitations.

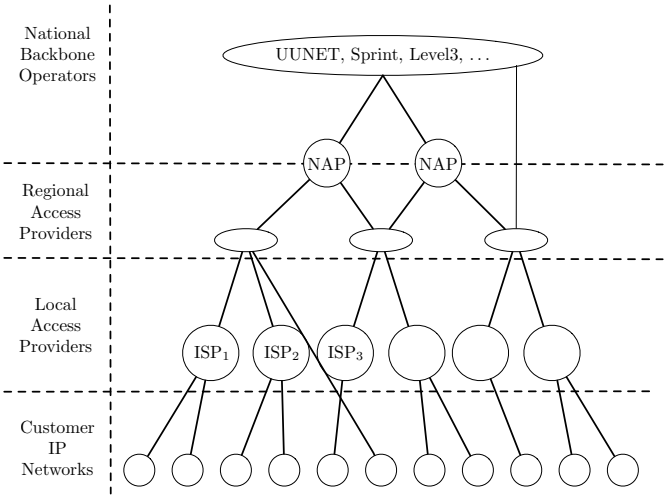


Fig. 13. Traditional AS hierarchy [132]

5.3.1 Routing Registry Information

Many publicly-available registries share information about the Internet and its topology. *Regional Internet Registries* [135] are organizations responsible for allocating AS numbers and IP address blocks, all of which are accessible using the WHOIS protocol [136]. *Internet Routing Registry* (IRR) [137] 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) [138].

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 in Sec. 2. 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 [139], 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.

5.3.2 BGP Routing Information

As opposed to link-state protocols such as OSPF [54] or IS-IS [140], 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 [141].

A second source of BGP information is *BGP dumps*. Projects such as RouteViews [142] 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 *update traces*. 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 [143].

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. [144], 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 interdomain routing as for example using node-probing, confirmed by Broido and claffy studies [145].

5.4 Inference Techniques

Early research assumed that two ASes were linked if their AS numbers were adjacent in an AS path. Gao and Rexford [146] then made a substantial advance

by noticing c2p links creating so a hierarchy. Gao went on to identify the p2p and s2s relationships [147].

Inferring these relationships is a problem of its own. In her study, Gao [147] first tackles the problem by developing an inference mechanism which extracts information from BGP tables and summarized *valley-free*[6] property of AS paths. Subramanian et al. [133] formulates AS relationship assignment as an optimization problem, a *type of relationship* (ToR) problem. Battista et al. [148] prove its NP-completeness and present an approximately optimal solution. Gao and Xia [149] evaluate then the accuracy of these algorithms and improve them by introducing techniques on inferring relationships from partial information, in particular the information coming from the BGP community attribute [150]. Dimitropoulos et al. [151] provides improvements to relationships inference. In particular, they provide techniques for inferring s2s from IRRs and heuristics for c2p and p2p relationships.

Chang et al. show that many existing links do not actually appear in BGP [152]. Therefore, they propose to infer the AS topology from Internet's router topology. Broido and claffy [145] 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.

A side issue in inferring AS topologies is to delineate the border of an AS. Indeed, border routers can be made of IP addresses belonging to the AS of interest, to a peer, or to a third party such as an Internet eXchange Point (IXP). Solutions to this issue have been proposed for AS topologies collected by IGMP probing [153] and by `traceroute` [152, 154].

5.5 Limitations

Several works [151, 155–157] demonstrate that AS level topology discovery based on current data collection efforts is limited. Indeed, for instance, Dimitropoulos et al. [151] show that BGP tables missed up to 80% of the ASes adjacencies (mainly p2p relationships). Dimitropoulos et al. [158] suggest to additionally consider BGP updates as the path exploration process may reveal new links between ASes. Others [144] suggest to actually mix the various data source (see Sec. 5.3) instead of considering each source in isolation to others.

Inferring AS level topology from `traceroute` is not exempt from limitations. It naturally comes with all drawbacks inherited from `traceroute` (see Sec. 2.3). In addition, it comes with another limitations called *third-party address*.

A third-party address is the address of a router interface that does not lie in the actual path taken by the packet [52]. Fig. 14 illustrates the problem of third-party address. Remind that, as explained in Sec. 2.1, the source address of the

[6] 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.

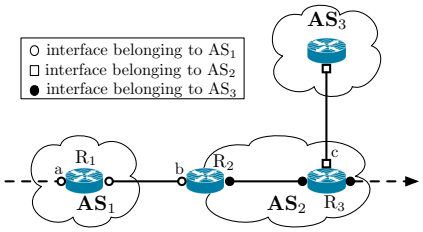


Fig. 14. Third-party address in a traceroute path [53]

`time-exceeded` message generated by a router is the outgoing interface of the reply, not the interface that received the packet generating the reply. On Fig. 14, let us assume that our `traceroute` contains the sequence `a, b, c` where `a` and `b` are incoming interfaces from router R_1 and R_2 and `c` the interface used by R_3 to send ICMP messages. `c` is a third-party address as it does belong to the actual path traversed by `traceroute` packets.

Third-party addresses have an impact on AS level topology. Indeed, when mapping IP addresses to their AS number, one can generate a false link. On Fig. 14, the third party address `c` generates a false AS link between AS_1 and AS_2.

Hyun et al. [52] suggest that third-party addresses occur generally at a few hops from the `traceroute` destination (i.e., at the destination edge of the network) and are found mainly in multihomed Stub networks. Marchetta et al. [53] use the IP timestamp option for revealing third-party addresses in a `traceroute`.

6 Conclusion

The Internet is an heterogeneous system made of interconnected entities allowing the communication between machines, from computers to smartphones. In this chapter, we have reviewed how data can be collected for obtaining the Internet topology. In particular, we focused on four levels of the topology: the IP interface, the router, the subnet, and, finally, the AS level. Each level has its own set of inference techniques (active or passive) with their advantages and drawbacks.

Does this chapter mean that everything has been done regarding Internet topology? Surely not. Several challenges are still open. For instance, large-scale distributed measurement infrastructures made of hundreds or thousands of monitors are more and more deployed (see, for instance, the recent RIPE Atlas [159]). 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.

Challenges are also on network measurement techniques and modeling. For instance, mechanisms for obtaining information about the network dynamics cannot rely on standard probing techniques. Further, current modeling approaches do not take into account network dynamics.

References

1. Pastor-Satorras, R., Vespignani, A.: Evolution and Structure of the Internet: A Statistical Physics Approach. Cambridge University Press (2004)
2. Govindan, R., Tangmunarunkit, H.: Heuristics for internet map discovery. In: Proc. IEEE INFOCOM (March 2000)
3. Teixeira, R., Marzullo, K., Savage, S., Voelker, G.: In search of path diversity in ISP networks. In: Proc. ACM SIGCOMM Internet Measurement Conference, IMC (October 2003)
4. Feldman, D., Shavitt, Y., Zilberman, N.: A structural approach for PoP geolocation. Computer Networks (COMNET) 56(3), 1029–1040 (2012)
5. Shavitt, Y., Zilberman, N.: Geographical Internet PoP level maps. In: Proc. Traffic Monitoring and Analysis Workshop, TMA (March 2012)
6. Tozal, M.E., Sarac, K.: TraceNET: an Internet topology data collector. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2010)
7. Tozal, M.E., Sarac, K.: Estimating network layer subnet characteristics via statistical sampling. In: Proc. IFIP Networking (May 2012)
8. Gunes, M., Sarac, K.: Inferring subnets in router-level topology collection studies. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2007)
9. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the Internet topology. In: Proc. ACM SIGCOMM (September 1999)
10. Haddadi, H., Iannaccone, G., Moore, A., Mortier, R., Rio, M.: Network topologies: Inference, modeling and generation. IEEE Communications Surveys and Tutorials 10(2), 48–69 (2008)
11. Alderson, D., Li, L., Willinger, W., Doyle, J.C.: Understanding Internet topology: Principles, models and validation. IEEE/ACM Transactions on Networking 13(6), 1205–1218 (2005)
12. Willinger, W., Alderson, D., Doyle, J.C.: Mathematics and the Internet: a source of enormous confusion and great potential. Notices of the American Mathematical Society 56(5), 586–599 (2009)
13. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
14. Mérindol, P., Donnet, B., Bonaventure, O., Pansiot, J.J.: On the impact of layer-2 on node degree distribution. In: Proc. ACM/USENIX Internet Measurement Conference (IMC (November 2010)
15. Palmer, C.R., Siganos, G., Faloutsos, M., Faloutsos, C., Gibbons, P.B.: The connectivity and fault-tolerane of the Internet topology. In: Proc. Workshop on Network-Related Data Management (May 2001)
16. Radoslavov, P., Tangmunarunkit, H., Yu, H., Govindan, R., Shenker, S., Estrin, D.: On characterizing network topologies and analyzing their impact on protocol design. Technical Report 00-731, Computer Science Department, University of Southern California (February 2000)
17. Jacobson, V., et al.: Traceroute. Man page, UNIX (1989), `ftp://ftp.ee.lbl.gov/traceroute.tar.gz`
18. Sommers, J., Eriksson, B., Barford, P.: On the prevalence and characteristics of MPLS deployments in the open Internet. In: ACM SIGCOMM Internet Measurement Conference (November 2011)
19. Donnet, B., Luckie, M., Mérindol, P., Pansiot, J.J.: Revealing MPLS tunnels obscured by traceroute. ACM SIGCOMM Computer Communication Review 42(2), 87–93 (2012)

20. Mérindol, P., Van den Schriek, V., Donnet, B., Bonaventure, O., Pansiot, J.J.: Quantifying ASes multiconnectivity using multicast information. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2009)
21. Gavron, E.: NANOG traceroute, ftp://ftp.login.com/pub/software/traceroute/
22. Baker, F.: Requirements for IP version 4 routers. RFC 1812, Internet Engineering Task Force (June 1995)
23. Torren, M.: Tcptraceroute - a traceroute implementation using TCP packets. Man page, UNIX (2001), http://michael.toren.net/code/tcptraceroute/
24. Luckie, M., Hyun, Y., Huffaker, B.: Traceroute probe method and forward IP path inference. In: Proc. ACM SIGCOMM Internet Measurement Conference, IMC (October 2008)
25. Mao, Z.M., Johnson, D., Rexford, J., Wang, J., Katz, R.H.: Scalable and accurate identification of AS-level forwarding paths. In: Proc. IEEE INFOCOM (April 2004)
26. Mao, Z.M., Rexford, J., Wang, J., Katz, R.H.: Towards an accurate AS-level traceroute tool. In: Proc. ACM SIGCOMM (August 2003)
27. Zhang, Y., Oliveira, R., Zhang, H., Zhang, L.: Quantifying the Pitfalls of Traceroute in AS Connectivity Inference. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 91–100. Springer, Heidelberg (2010)
28. Claffy, K., Hyun, Y., Keys, K., Fomenkov, M., Krioukov, D.: Internet mapping: from art to science. In: Proc. IEEE Cybersecurity Applications and Technologies Conference for Homeland Security, CATCH (March 2009)
29. Huffaker, B., Plummer, D., Moore, D.: claffy, k.: Topology discovery by active probing. In: Proc. Symposium on Applications and the Internet, SAINT (January 2002)
30. Luckie, M.: Scamper: a scalable and extensible packet prober for active measurement of the Internet. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2010)
31. Georgatos, F., Gruber, F., Karrenberg, D., Santcroos, M., Susanj, A., Uijterwaal, H., Wilhelm, R.: Providing active measurements as a regular service for ISPs. In: Proc. Passive and Active Measurement Workshop, PAM (April 2001)
32. McGregor, A., Braun, H.W., Brown, J.: The NLANR network analysis infrastructure. IEEE Communications Magazine 38(5), 122–128 (2000)
33. Shavitt, Y., Shir, E.: DIMES: Let the internet measure itself. ACM SIGCOMM Computer Communication Review 35(5), 71–74 (2005), http://www.netdimes.org
34. Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M., Werthimer, D.: SETI@home: An experiment in public-resource computing. Communications of the ACM 45(11), 56–61 (2002), http://setiathome.berkeley.edu/
35. Waddington, D.G., Chang, F., Viswanathan, R., Yao, B.: Topology discovery for public IPv6 networks. ACM SIGCOMM Computer Communication Review 33(3), 59–68 (2003)
36. Lang, X., Zhou, G., Gong, C., Han, W.: Dolphin: the measurement system for the next generation Internet. In: Proc. 4th International Conference on Communications, Internet and Information Technology, CIIT (November 2005)
37. Liu, Z., Luo, J., Wang, Q.: Large-scale topology discovery for public IPv6 networks. In: Proc. International Conference on Networking, ICN (April 2008)

38. Madhyastha, H.V., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., Venkataramani, A.: iPlane: An information plane for distributed services. In: Proc. USENIX Symposium on Operating Systems Design and Implementation, OSDI (November 2006)
39. Sherwood, R., Bender, A., Spring, N.: Discarte: a disjunctive Internet cartographer. In: Proc. ACM SIGCOMM (August 2008)
40. Postel, J.: Internet protocol. RFC 791, Internet Engineering Task Force (September 1981)
41. Fonseca, R., Porter, G., Katz, R., Shenker, S., Stoica, I.: IP options are not an option. Technical Report UCB/EECS-2005-24, University of California, EECS Department (December 2005)
42. Botta, A., de Donato, W., Pescapé, A., Ventre, G.: Discovering topologies at router level: Part II. In: Proc. IEEE Global Telecommunications Conference, GLOBECOM (November 2007)
43. Wide: Gulliver: Distributed active measurement project (2006), `http://gulliver.wide.ad.jp/`
44. Barford, P., Bestavros, A., Byers, J., Crovella, M.: On the marginal utility of network topology measurements. In: Proc. ACM SIGCOMM Internet Measurement Workshop, IMW (November 2001)
45. Guillaume, J.L., Latapy, M.: Relevance of massively distributed explorations of the Internet topology: Simulation results. In: Proc. IEEE INFOCOM (March 2005)
46. Shavitt, Y., Weinsberg, U.: Quantifying the importance of vantage points distribution in Internet topology measurements. In: Proc. IEEE INFOCOM (April 2009)
47. Chen, K., Choffnes, D., Potharaju, R., Chen, Y., Bustamante, F., Pei, D., Zhao, Y.: Where the sidewalk ends: Extending the Internet AS graph using traceroutes from P2P users. In: Proc. ACM SIGCOM CoNEXT (December 2009)
48. Latapy, M., Magnien, C., Ouédrago, F.: A radar for the Internet. In: Proc. International Workshop on Analysis of Dynamics Networks, ADN (December 2008)
49. Magnien, C., Ouédraogo, F., Valadon, G., Latapy, M.: Fast dynamics in Internet topology: Preliminary observations and explanations. In: Proc. International Conference on Internet Monitoring and Protection, ICIMP (May 2009)
50. Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., Teixeira, R.: Avoiding traceroute anomalies with Paris traceroute. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (October 2006)
51. Viger, F., Augustin, B., Cuvellier, X., Magnien, C., Friedman, T., Teixeira, R.: Detection, understanding, and prevention of traceroute measurement artifacts. Computer Networks 52(5), 998–1018 (2008)
52. Hyun, Y., Broido, A., Claffy, K.: On third-party addresses in traceroute paths. In: Proc. Passive and Activement Measurement Workshop, PAM (April 2003)
53. Marchetta, P., de Donato, W., Pescapé, A.: Detecting third-party addresses in traceroute IP paths. In: Proc. ACM SIGCOMM (August 2012) (extended abstract)
54. Moy, J.: OSPF version 2. RFC 2328, Internet Engineering Task Force (April 1998)
55. Callon, R.: Use of OSI IS-IS for routing in TCP/IP and dual environments. RFC 1195, Internet Engineering Task Force (December 1990)

56. Thaler, D., Hopps, C.: Multipath issues in unicast and multicast next-hop selection. RFC 2991, Internet Engineering Task Force (November 2000)
57. Quoitin, B., Uhlig, S., Pelsser, C., Swinnen, L., Bonaventure, O.: Interdomain traffic engineering with BGP. IEEE Communication Magazine 41(5), 122–128 (2003)
58. CISCO: How does load balancing work? http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml
59. Juniper: Configuring load-balance per-packet action, http://www.juniper.net/techpubs/software/junos/junos70/swconfig70-policy/html/policy-actions-config11.html
60. Luckie, M., Dhamdhere, A., Claffy, K., Murrel, D.: Measured impact of crooked traceroute. ACM SIGCOMM Computer Communication Review 41(1), 15–21 (2011)
61. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with Rocketfuel. In: Proc. ACM SIGCOMM (August 2002)
62. Augustin, B., Teixeira, R., Friedman, T.: Measuring load-balanced paths in the Internet. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2007)
63. Augustin, B., Friedman, T., Teixeira, R.: Measuring multipath routing in the Internet. IEEE/ACM Transactions on Networking 19(3), 830–840 (2011)
64. Veitch, D., Augustin, B., Friedman, T., Teixeira, R.: Failure control in multipath route tracing. In: Proc. IEEE INFOCOM (April 2009)
65. Donnet, B., Raoult, P., Friedman, T., Crovella, M.: Efficient algorithms for large-scale topology discovery. In: Proc. ACM SIGMETRICS (June 2005)
66. Donnet, B., Raoult, P., Friedman, T., Crovella, M.: Deployment of an algorithm for large-scale topology discovery. IEEE Journal on Selected Areas in Communications (JSAC), Sampling the Internet: Techniques and Applications 24(12), 2210–2220 (2006)
67. Spring, N., Wetherall, D., Anderson, T.: Scriptroute: A public Internet measurement facility. In: Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS), pp. 225–238 (March 2003)
68. Donnet, B., Friedman, T., Crovella, M.: Improved Algorithms for Network Topology Discovery. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 149–162. Springer, Heidelberg (2005)
69. Donnet, B., Friedman, T.: Topology discovery using an address prefix based stopping rule. In: Proc. EUNICE Workshop (July 2005)
70. Donnet, B., Huffaker, B., Friedman, T., Claffy, K.: Increasing the Coverage of a Cooperative Internet Topology Discovery Algorithm. In: Akyildiz, I.F., Sivakumar, R., Ekici, E., Oliveira, J.C.d., McNair, J. (eds.) NETWORKING 2007. LNCS, vol. 4479, pp. 738–748. Springer, Heidelberg (2007)
71. Beverly, R., Berger, A., Xie, G.: Primitives for active Internet topology mapping: Toward high-frequency characterization. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2010)
72. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol label switching architecture. RFC 3031, Internet Engineering Task Force (January 2001)
73. Muthukrishnan, K., Malis, A.: A core MPLS IP VPN architecture. RFC 2917, Internet Engineering Task Force (September 2000)
74. Srinivasan, C., Bloomberg, L.P., Viswanathan, A., Nadeau, T.: Multiprotocol label switching (MPLS) traffic engineering (TE) management information base (MIB). RFC 3812, Internet Engineering Task Force (June 2004)

75. Xiao, X., Hannan, A., Bailey, B.: Traffic engineering with MPLS in the Internet. IEEE Network Magazine 14(2) (April 2000)
76. Andersson, L., Minei, I., Thomas, T.: LDP specification. RFC 5036, Internet Engineering Task Force (October 2007)
77. Farrel, A., Ayyangar, A., Vasseur, J.P.: Inter-domain MPLS and GMPLS traffic engineering – resource reservation protocol-traffic engineering (RSVP-TE extensions). RFC 5151, Internet Engineering Task Force (February 2008)
78. Bonica, R., Gan, D., Tappan, D., Pignataro, C.: ICMP extensions for multiprotocol label switching. RFC 4950, Internet Engineering Task Force (August 2007)
79. Bonica, R., Gan, D., Tappan, D., Pignataro, C.: Extended ICMP to support multi-part messages. RFC 4884, Internet Engineering Task Force (April 2007)
80. Jacquin, L., Rocca, V., Kaafar, M.A., Schuler, F., Roch, J.L.: IBTrack: An ICMP black holes tracker. In: Proc. IEEE Global Communications Conference, GLOBECOM (December 2012)
81. Katz-Bassett, E., Madhyastha, H., Adhikari, V., Scott, C., Sherry, J., van Wesep, P., Krishnamurthy, A., Anderson, T.: Reverse traceroute. In: Proc. USENIX Symposium on Networked Systems Design and Implementations, NSDI (June 2010)
82. He, Y., Faloutsos, M., Krishnamurthy, S., Huffaker, B.: On routing asymmetry in the Internet. In: Proc. IEEE Global Telecommunications Conference, GLOBECOM (November 2005)
83. Su, Z.S.: A specification of the Internet protocol (IP) timestamp option. RFC 781, Internet Engineering Task Force (May 1981)
84. Morris, R.: A weakness in the 4.2 BSD unix TCP/IP software. Technical Report 117. Bell Labs (February 1985)
85. Bellovin, S.M.: Security problems in the TCP/IP protocol suite. ACM SIGCOMM Computer Communication Review 19(2), 32–48 (1989)
86. Ferguson, P., Senie, D.: Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827, Internet Engineering Task Force (May 2000)
87. Baker, F., Savola, P.: Ingress filtering for multihomed networks. RFC 3704, Internet Engineering Task Force (March 2004)
88. Jin, C., Wang, H., Shin, K.: Hop-count filtering: An effective defense against spoofed DoS traffic. In: Proc. 10th ACM International Conference on Computer and Communications Security, CCS (October 2003)
89. Leech, M., Taylor, T.: ICMP traceback messages. Internet Draft (work in progress) draft-ietf-itrace-04.txt, Internet Engineering Task Force (February 2003)
90. Snoeren, A.C., Partridge, C., Sancheq, L.A., Joes, C.E., Tchakountio, F., Kent, S.T., Strayer, W.T.: Hash-based IP traceback. In: Proc. ACM SIGCOMM (August 2001)
91. Beverly, R., Bauer, S.: The Spoofer projet: Inferring the extent of source address filtering in the Internet. In: Proc. Steps to Reducing Unwanted Traffic on the Internet Workshop, SRUTI (July 2005)
92. Yao, B., Chang, R.V., Waddington, F., Topology, D.: inference in the presence of anonymous routers. In: Proc. IEEE INFOCOM (April 2003)
93. Gunes, M.H., Sarac, K.: Resolving anonymous routers in the Internet topology measurement studies. In: Proc. IEEE INFOCOM (April 2008)
94. IANA: Special-use IPv4 addresses. RFC 3330, Internet Engineering Task Force (September 2002)

95. Cheswick, B., Burch, H., Branigan, S.: Mapping and visualizing the Internet. In: Proc. ACM/USENIX Annual Technical Conference (June 2000)
96. Broido, A., Claffy, K.: Internet topology: Connectivity of IP graphs. In: International Conference on Internet, Performance and Control of Networks, ITCom (August 2001)
97. Bilir, S., Sarac, K., Korkmaz, T.: Intersection characteristics of end-to-end Internet paths and trees. In: IEEE International Conference on Network Protocols, ICNP (November 2005)
98. Jin, X., Yiu, W.P.K., Chan, S.H.G., Wang, Y.: Network topology inference based on end-to-end measurements. IEEE Journal on Selected Areas in Communication, Sampling the Internet: Techniques and Applications 24(12), 2182–2195 (2006)
99. Gunes, M.H., Sarac, K.: Importance of IP alias resolution in sampling Internet topologies. In: Proc. IEEE Global Internet Symposium (May 2007)
100. Braden, R.: Requirements for internet hosts. communication layers. RFC 1122, Internet Engineering Task Force (October 1989)
101. Keys, K.: Iffinder A tool for mapping interfaces to routers, `http://www.caida.org/tools/measurement/iffinder/`
102. Bender, A., Sherwood, R., Spring, N.: Fixing Ally's growing pains with velocity modeling. In: Proc. ACM SIGCOMM Internet Measurement Conference, IMC (October 2008)
103. Keys, K., Hyun, Y., Luckie, M.: claffy, k.: Internet-scale IPv4 alias resolution with MIDAR: System architecture. Technical Report v.0, Cooperative Association for Data Analysis (CAIDA) (May 2011)
104. Kim, S., Harfoush, K.: Efficient estimation of more detailed Internet IP maps. In: Proc. IEEE International Conference on Communications, ICC (June 2007)
105. Zhang, M., Ruan, Y., Pai, V., Rexford, J.: How DNS misnaming distorts Internet topology mapping. In: Proc. USENIX Annual Technical Conference (May/June 2006)
106. Keys, K.: Internet-scale IP alias resolution techniques. ACM SIGCOMM Computer Communication Review 40(1), 50–55 (2010)
107. Tozal, M.E., Sarac, K.: Palm tree: an IP alias resolution algorithm with linear probing complexity. Computer Communications (COMCOM) 34(5), 658–669 (2011)
108. Fuller, V., Li, T.: Classless inter-domain routing (CIDR): the Internet address assignment and aggregation plan. RFC 4632, Internet Engineering Task Force (August 2006)
109. Sherry, J., Katz-Bassett, E., Pimenova, M., Madhyastha, H.V., Anderson, T., Krishnamurthy, A.: Resolving IP aliases with prespecified timestamps. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2010)
110. Cooperative Association for Internet Data Analysis (CAIDA): Motu dealiasing tool (October 2011), `http://www.caida.org/tools/measurement/motu/`
111. de Donato, W., Marchetta, P., Pescapé, A.: A Hands-on Look at Active Probing Using the IP Prespecified Timestamp Option. In: Taft, N., Ricciato, F. (eds.) PAM 2012. LNCS, vol. 7192, pp. 189–199. Springer, Heidelberg (2012)
112. Garcia-Jimenez, S., Magana, E., Izal, M., Morato, D.: Validity of router responses for IP aliases resolution. In: Proc. IFIP Networking (May 2012)
113. Spring, N., Dontcheva, M., Rodrig, M., Wetherall, D.: How to resolve IP aliases. Technical Report 04-05-04, UW CSE (May 2004)

114. Gunes, M., Sarac, K.: Analytical IP alias resolution. In: Proc. IEEE International Conference on Communications, ICC (June 2006)
115. Gunes, M.H., Sarac, K.: Resolving IP aliases in building traceroute-based Internet maps. IEEE/ACM Transactions on Networking (ToN) 17(6), 1738–1751 (2009)
116. Jacobson, V.: Mrinfo (1995), `http://cvsweb.netbsd.org/bsdweb.cgi/src/usr.sbin/mrinfo/?only_with_tag=MAIN`
117. Deering, S.: Host extensions for IP multicasting. RFC 1112, Internet Engineering Task Force (August 1989)
118. Pusateri, T.: Distance vector multicast routing protocol version 3 (DVMRP). Internet Draft (Work in Progress) draft-ietf-idmr-dvmrp-v3-11, Internet Engineering Task Force (October 2003)
119. Mérindol, P., Donnet, B., Pansiot, J.J., Luckie, M., Hyun, Y.: MERLIN: MEasure the Router Level of the INternet. In: Proc. 7th Euro-nf Conference on Next Generation Internet, NGI (June 2011)
120. Marchetta, P., Mérindol, P., Donnet, B., Pescapé, A., Pansiot, J.J.: Topology discovery at the router level: a new hybrid tool targeting ISP networks. IEEE Journal on Selected Areas in Communication, Special Issue on Measurement of Internet Topologies 29(6), 1776–1787 (2011)
121. Fenner, B., Handley, M., Holbrook, H., Kouvelas, I.: Protocol independent multicast - sparse mode (PIM-SM: Protocol specification). RFC 4601, Internet Engineering Task Force (August 2006)
122. Marchetta, P., Mérindol, P., Donnet, B., Pescapé, A., Pansiot, J.J.: Quantifying and mitigating IGMP filtering in topology discovery. In: Proc. IEEE Global Communications Conference, GLOBECOM (December 2012)
123. Mogul, J., Postel, J.: Internet standard subnetting procedure. RFC 950, Internet Engineering Task Force (August 1985)
124. Tozal, M.E., Sarac, K.: Subnet level network topology mapping. In: Proc. IEEE International Performance Computing and Communications Conference, IPCCC (November 2011)
125. Fenner, W.: Internet group management protocol (IGMP), version 2. RFC 2236, Internet Engineering Task Force (November 1997)
126. Hubbard, K., Kosters, M., Conrad, D., Karrenberg, D., Postel, J.: Internet registry IP allocation guidelines. RFC 2050, Internet Engineering Task Force (November 1996)
127. Retana, A., White, R., Fuller, V., McPherson, D.: Using 31-bit prefixes on IPv4 point-to-point links. RFC 3021, Internet Engineering Task Force (December 2000)
128. Vohra, Q., Chen, E.: BGP support for four-octet AS number space. RFC 4893, Internet Engineering Task Force (May 2007)
129. Tangmunarunkit, H., Govindan, R., Estrin, D., Shenker, S.: The impact of routing policy on Internet paths. In: Proc. IEEE INFOCOM (April 2001)
130. Gao, L., Wang, F.: The extent of AS path inflation by routing policies. In: Proc. IEEE Global Internet Symposium (November 2002)
131. Tangmunarunkit, H., Govindan, R., Shenker, S.: Internet path inflation due to policy routing. In: Proc. SPIE International Symposium on Convergence of IT and Communication, ITCom (August 2001)
132. Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F.: Internet inter-domain traffic. In: Proc. ACM SIGCOMM (August 2010)

133. Subramanian, L., Agarwal, S., Rexford, J., Katz, R.H.: Characterizing the Internet hierarchy from multiple vantage points. In: Proc. IEEE INFOCOM (June 2002)
134. Zhang, B., Liu, R., Massey, D., Zhang, L.: Collecting the Internet AS-level topology. ACM SIGCOMM Computer Communication Review 35(1), 53–61 (2005)
135. The Internet Society: The regional Internet registry structure, `http://www.isoc.org/briefings/021/`
136. Daigle, L.: WHOIS protocol specification. RFC 3912, Internet Engineering Task Force (September 2004)
137. Network, T.M.: Internet routing database, `ftp://ftp.radb.net/routing.arbiter/radb/dbase/`
138. Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenber, D.: Routing policy specification language (RPSL). RFC 2622, Internet Engineering Task Force (June 1999)
139. RIPE NCC: Routing registry consistency check reports, `http://www.ripe.net/projects/rrcc/`
140. ISO: Intermediate system to intermediate system intra-domain routing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473) International Standard 10589:2002
141. Kernen, T.: Traceroute organization, `http://www.traceroute.org`
142. University of Oregon: Route views, University of Oregon Route Views project, `http://www.routeviews.org/`
143. Free Software Foundation: Zebra, `http://www.gnu.org/software/zebra/`
144. Changa, H., Govindan, R., Jamina, S., Shenker, S.J., Willinger, W.: Towards capturing representative AS level Internet topologies. Computer Networks (COMNET) 44(6), 737–755 (2004)
145. Broido, A., Claffy, K.: Internet topology: Connectivity of IP graphs. In: Proc. SPIE International Symposium on Convergence of IT and Communication, ITCom (August 2001)
146. Gao, L., Rexford, J.: Stable Internet routing without global coordination. In: Proc. ACM SIGMETRICS (June 2000)
147. Gao, L.: On inferring autonomous system relationships in the Internet. In: Proc. IEEE Global Internet Symposium (November 2000)
148. Di Battista, G., Patrignani, M., Pizzonia, M.: Computing the types of the relationships between autonomous systems. In: Proc. IEEE INFOCOM (April 2003)
149. Xia, J., Gao, L.: On the evaluation of AS relationship inferences. In: Proc. IEEE Global Communications Conference (GLOBECOM (November 2004)
150. Quoitin, B., Bonaventure, O.: A survey of the utilization of the BGP community attribute. Internet Draft draft-quoitin-bgp-comm-survey-00, Internet Engineering Task Force (February 2002) (work in progress)
151. Dimitropoulos, X., Krioukov, D., Fomenkov, M., Huffaker, B., Hyun, Y., Claffy, K., Riley, G.: AS relationships: Inference and validation. ACM SIGCOMM Computer Communication Review 37(1), 29–40 (2007)
152. Chang, H., Jamin, S., Willinger, W.: Inferring AS-level Internet topology from router-level path traces. In: Proc. SPIE International Symposium on Convergence of IT and Communication, ITCom (August 2001)
153. Pansiot, J.-J., Mérindol, P., Donnet, B., Bonaventure, O.: Extracting Intra-domain Topology from `mrinfo` Probing. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 81–90. Springer, Heidelberg (2010)

154. Huffaker, B., Dhamdhere, A., Fomenkov, M., Claffy, K.: Toward Topology Dualism: Improving the Accuracy of AS Annotations for Routers. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 101–110. Springer, Heidelberg (2010)
155. Oliveira, R., Pei, D., Willinger, W., Zhang, B., Zhang, L.: In search of the elusive ground truth: The Internet's AS-level connectivity structure. In: ACM SIGMETRICS (June 2008)
156. Oliveira, R., Pei, D., Willinger, W., Zhang, B., Zhang, L.: The (in)completeness of the observed Internet AS-level structure. IEEE/ACM Transactions on Networking 18(1), 109–122 (2010)
157. Willinger, W., Maennel, O., Perouli, D., Bush, R.: 10 lessons from 10 years of measuring and modeling the Internet's autonomous systems. IEEE Journal on Selected Areas in Communications (JSAC) 29(9), 1810–1821 (2011)
158. Dimitropoulos, X.A., Krioukov, D.V., Riley, G.F.: Revisiting Internet AS-Level Topology Discovery. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 177–188. Springer, Heidelberg (2005)
159. RIPE NCC: Atlas (2010), `https://atlas.ripe.net/`