A Hybrid Tool for Discovering Router Level Core of Multicast ASes

Pietro Marchetta*, Pascal Merindol‡, Benoit Donnet†, Antonio Pescapé*, Jean-Jacques Pansiot‡
∗ Dipartimento di Informatica e Sistemistica, University of Napoli Federico II – Italy
‡ LSIIT, University of Strasbourg – France
† Montefiore Institute, University of Liége – Belgium

Abstract

The recent introduction of IGMP probing allows to natively discover the router level topology of the multicast enable part of the network. However, IGMP probes are subject to filtering, leading so to the fragmentation of the collected multicast graph into several disjoint connected components. In this paper, we first quantify the effects of IGMP filtering in large tier-1 ISPs and show that resulting topologies are heavily fragmented. Using traceroute data, we construct a hybrid graph and estimate how far each IGMP fragment is from each other. Based on the distance distribution resulting from this analysis, we demonstrate that most IGMP components can be merged into a large connected multicast component. We thus propose and evaluate an efficient approach for reconnecting IGMP components at the router level. The key idea is to recursively use IP level information and alias resolution to reassemble disjoint fragments and progressively extend the mapping of the targeted AS.

1 Introduction

Since the late 90's, the Internet topology discovery has known a growing interest, leading to several papers proposing new tools for collecting data [Don07]. The Internet topology can be seen at different levels. In this paper, we focus on the *router level*. At this level, the topology is seen as a graph where routers are nodes. Such a representation is generally obtained by aggregating IP interfaces (collected via the traceroute tool) through a technique called *alias resolution* [Key10].

While inferring the router level topology of IP networks is an important aspect, in particular to study routing characteristics and design new efficient protocols, tools for capturing the router vision of ASes come with a cost. On the one hand, traceroute is known as being redundant [Don06] and collected information can be biased. On the other hand, active alias resolution (i.e., based on additional probing) can be intrusive and prone to error as false positives (i.e., two IP addresses are declared as aliases while they do not belong to the same router) and negatives [Key10].

The recently introduced Merlin [Mer11] [Mar11], which is based on IGMP recursive probing, can natively discover multicast topologies at the router level with a low probing cost [Mer09, Pan10, Mer10]. Thus, Merlin does not rely on any alias resolution techniques: soliciting an IGMP `NEIGHBORS_REPLY` message with an IGMP `ASK_NEIGHBORS` probe, Merlin is able to obtain all multicast interfaces and IP neighbors of the targeted router. While the resulting vision may be incomplete (because limited to the multicast part), it is also less subject to false positives than common topology discovery techniques. In practice, note that major transit ASes (such as Tier-1 or Tier-2 provider networks) are more likely to deploy multicast capabilities than private networks such as CDNs. Indeed, since major transit ASes have numerous clients they are more induced to deploy many services such as IP multicast routing. In constrast, CDN or local access providers may deploy multicast IP routing depending on their own needs (for example to provide IP TV channels). It is also worth to notice that multicast support for MPLS VPN needs to run native multicast on the provider network; MPLS backbone multicast routers should thus reply to Merlin as standard

Copyright © 2012 by the paper's authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors. Published at http://ceur-ws.org

multicast routers. Unfortunately, some routers do not reply to IGMP probes sent by MERLIN, leading to an anonymous behavior that is similar to the one observed with traceroute [Gun08]. We call this phenomenon *IGMP filtering*. As a consequence, the topology obtained with MERLIN is incomplete and disconnected. The main purpose of this paper is to reassemble these disjoint components at the router level.

In this paper, we quantify the impact of IGMP filtering on the multicast topologies collected with MERLIN (Sec. 2). First, we propose a methodology to evaluate how IGMP filtering impacts the collected topologies (Sec. 2.1); then, we provide results (see Sec. 2.2) of the proposed methodology on three different large ASes. Based on traceroute traces and, thus, the use of a hybrid graph, we suggest a technique for estimating how far each fragment is from others and demonstrate that most of them can be reconnected using only two IP hops sequence. As multicast fragments are "close" to each others, we can expect to efficiently reassemble them: we show how it is possible to merge most IGMP components into a larger connected multicast component. To this aim, in Sec. 3, we propose a recursive approach that comes with the advantage of limiting the complexity of the topology reconstruction as well as potential errors introduced at each step. Finally, in Sec. 4 we present and discuss the results of our reassembling approach. Our final topologies are available at `http://svnet.u-strasbg.fr/merlin/`.

2 IGMP Filtering

MERLIN could suffer from the multicast graph "disconnection" due to IGMP filtering: some multicast routers do not reply to IGMP probes (*local filtering*) while some other do not forward IGMP queries (*transit filtering*). While the second problem can be somehow overcome with the use of multiple vantage points in a cooperative distributed platform, 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. Note how a low proportion of non-responding routers may result in an huge disconnection of the multicast graph.

This "disjoint state" may be exacerbate by unicast adjacencies lacks. In practice, a multicast router can be configured at the interface granularity: each interface can independently support multicast or not. Nevertheless, an ISP supporting IP multicast should enable multicast everywhere in its network to ensure the correct PIM tree establishment. Some exceptions may arise at inter-area border routers and AS border routers. An area border router does not need to support multicast adjacencies with routers belonging to non-multicast areas. Between ASes, the BGP routing protocol can use specific multicast forwarding entries to disseminate PIM messages. Thus, although it is likely that a multicast border router will not enable multicast on all its interfaces, it is also likely that the multicast graph should be connected.

2.1 Methodology

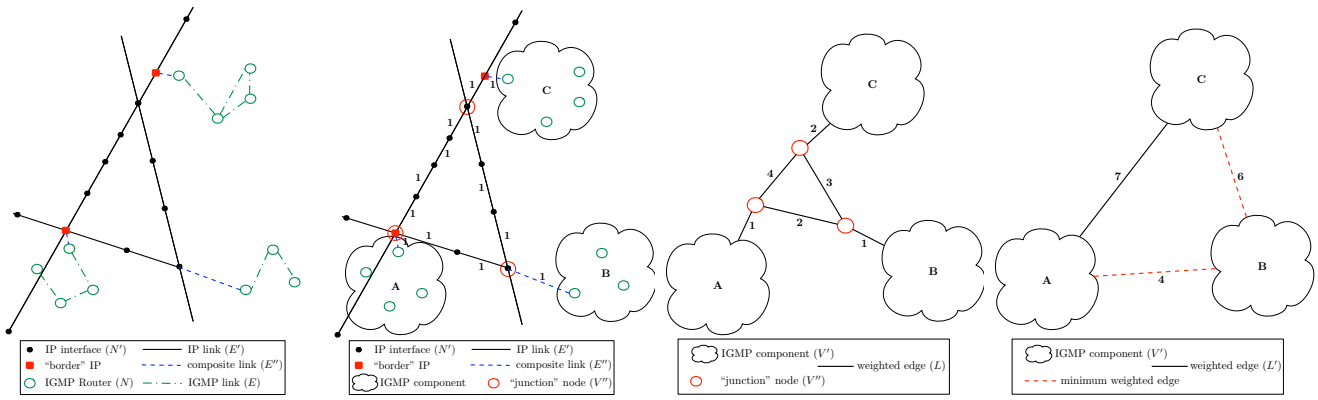
To evaluate how IGMP filtering impacts the collected topologies, we considered three large ISPs: Sprint (AS1239), Level3 (AS3356), and Global Crossing (AS3549). We selected those ASes among our set of experiments since they are representative of difficulties to obtain a fully connected topology.

We launched a MERLIN probing campaign towards each AS from five vantage points: Strasbourg (France), Napoli (Italy), Louvain-la-Neuve (Belgium), Hamilton (New Zealand), and San Diego (USA)[1]. In addition to IGMP probing, we performed a large Paris Traceroute [Aug06] campaign targeting each IGMP router previously discovered with MERLIN. In particular, we launched one Paris Traceroute per /24 prefix per router. The combination of IGMP and ICMP replies leads to a hybrid 2-tier graph where some nodes are routers (the IGMP view) or IP interfaces (the ICMP view), as illustrated in Fig. 1(a).

In this section, we are interested in the connected components size distribution and in the "distance" between connected component distributions. While evaluating the connected component size is straightforward, obtaining the distance between the components is not that easy. Fig. 1 illustrates the procedure described in this section on a small example. In the remainder of the paper the notation (V, L) refers to an undirected graph composed of a set of vertices, V, and a set of bidirectional links, L. Apart when explicitly specified, the valuation of links is uniform such that the path distance metric only relies on the number of hops (in terms of IP level links). We have a graph $G^1(\{N, N'\}, \{E, E', E''\})$ where: N is the set of IGMP routers; N' is the union of the ICMP IP interfaces set and the IGMP border IP interfaces set (only neighbor IP interfaces - see set B in Sec. 3); E is the IGMP adjacencies set (router level links between nodes in N); E' is the IP level links set (links between nodes in N'); E'' is the hybrid connections set (such that they inter-connect a router level node and an IP interface one).

The set $\{N, N'\}$ describes nodes in our hybrid graph and the set $\{E, E', E''\}$ provides edges between them.

[1] Measurements were done between April, 4$^{\text{th}}$ 2011 and April, 9$^{\text{th}}$ 2011.



(a) Original 2-Tier graph: G^1 (b) Reduction (c) Intermediate format: G^2 (d) Final graph: G^3

Figure 1: Compute minimal distance between disjoint IGMP components

The set E'' is composite since it describes edges linking nodes of different types (router level or IP level nodes). An edge $a \leftrightarrow b$ belongs to E'' if and only if a and b do not belong to the same level nodes set. This corresponds to dashed lines in Fig. 1(a). The set E'' is the key point of the analysis since it describes the interaction between the two nodes levels, being therefore the starting point for reconnecting disjoint IGMP components. An edge is added to $E'' = E''_b \bigcup E''_n$ according to two possible cases: (i) an IGMP router reports a neighbor IP address that is not locally attached to another IGMP router (this subset is denoted E''_b), (ii) a traceroute intersects a node belonging to N (this subset is denoted E''_n). Note that a node in N is a set of local IP interfaces, an IGMP alias, such that E''_n is almost equivalent to the intersection between IGMP and ICMP probing coverage. Moreover, it is worth to notice that we have no guarantee that G is connected (it mainly depends on the utility of traceroute traces), so that the distance distribution analysis may be incomplete.

For the specific purpose of our analysis, G^1 can be reduced to a weighted graph $G^2(V, L, w)$ where nodes in V are either connected components of IGMP routers in the graph (N, E) (such a connected component becomes a node in the set V') or IP interfaces in N' whose degree is strictly greater than two in G (this set of nodes is denoted V'', $a \in V'' \Leftrightarrow deg_{G^1}(a) \geq 3$). Thus, we have $V = V' \bigcup V''$. The valuation w of an edge in L is the hop distance between nodes in the graph $(V, \{E', E''\})$. Since nodes whose degree is lower or equal than two are "removed" from N' to form V'', we keep track of this distance information: $\forall a, b \in V$, $w(a \leftrightarrow b) - 1$ is equal to the number of nodes $\in N'$ removed from the shortest path between a and $b \in (V, \{E', E''\})$ if any, $w(a \leftrightarrow b) = \infty$ otherwise. Note that this reduction operation preserves distances computed in the initial graph. Fig. 1(b) illustrates the reduction operation: after such an operation, nodes in N' whose degree is still greater than 3 become "junction nodes", i.e articulation points of the new graph. Moreover, nodes belonging to the same IGMP connected component are merged so that they become a "IGMP cloud". Fig. 1(c) provides the resulting graph G^2: distances between nodes in V are updated to reflect the number of hops between them.

Then, the graph G^2 can be reduced to a third graph $G^3(V', L', w')$ where V' is the set of connected components of IGMP routers, L' are links between them, i.e.,

$$\forall e = a \leftrightarrow b \in L' \ (a, b \in V'), \ w(e) = min(d_{G^2}(a, b)).$$

The metric d_{G^2} provides the distances of all existing paths (in the graph G^2) between nodes in V' that do not contain any "intermediate" nodes in V'. Thus, $min(d_g(a, b))$ describes the shortest path distance between a and b using intermediate nodes only in $V'' = V \setminus V'$. For this purpose, we use a modified version of the *Dijkstra* algorithm on $g(V', L, w)$ where the `extract_min_distance` operation is limited to nodes in V''.

From the last reduced graph $G^3(V', L', w')$, we compute its resulting minimal weighted tree with the *Kruskal algorithm*. This final computation allows us to obtain a minimal distance distribution between disjoint IGMP components. Fig. 1(d) illustrates the final result: $\{4, 6\}$ is the minimal distances distribution for IGMP connected component A, B, and C.

The weight of edges belonging to the resulting minimal weighted tree describes the *a priori* required minimal effort to reconnect the topology. This metric has several advantages but also suffers from the interface level view provided by traceroute. On the one hand, it offers insights on the required effort to reconnect the topology: the more important the distances, the more intense the reconnection. On the other hand, although this metric is *a priori* stable to analyze the evolution of the topology reconnection (the reconnection of two

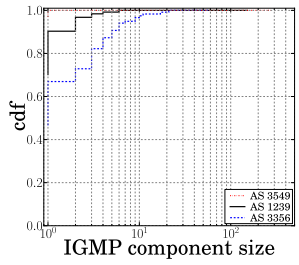


Figure 2: IGMP connected component size distribution

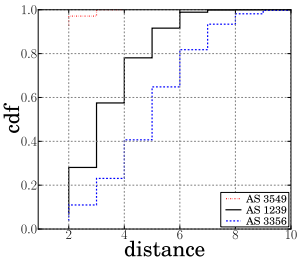


Figure 3: Distance distribution

disjoint components does not impact other distances than those between them), it may suffer from the lack of IP alias resolution. Indeed, nodes that describe different IP interfaces (and so different nodes in N') may belong to the same router, and thus falsely increase distances between disjoint components. Hence, this metric provides a worst case scenario to reconnect the topology without relying on any aliasing knowledge.

2.2 Evaluation

Fig. 2 provides the IGMP connected component size distribution for the three ASes of interest. The horizontal axis, in log-scale, is the component size (i.e., the number of routers included in a given IGMP component), while the vertical axis is the cumulative distribution. Although a very low proportion of IGMP components are quite large (larger than 200 for AS 3549), we see that the vast majority of IGMP components are made of a single router (70% for AS1239, 46% for AS3356, and 96% for AS3549): although MERLIN is able to capture one or two reasonably large connected components within an AS, most of the time, it discovers information about isolated IGMP routers. Table 2.2 provides most relevant information about the studied graphs (for instance the total number of collected IGMP routers, N). Note that in the scope of the distance analysis provided here and for each targeted ISP, we consider all the IP interfaces collected by Paris Traceroute towards the AS of interest whereas, in Sec. 3.1, we apply a conservative IP2AS filtering .

Analyzing the final graph G^3, we observe two notable properties. First, on the three explored ASes, we notice that most of disjoint IGMP components are "reconnectable" thanks to our dataset of traceroute traces, i.e., there exists at least one path in G^3 connecting the vast majority of the pairs of nodes in V'. Only (respectively for AS3549, AS1239 and AS3356) 2, 6, and 8 IGMP components (made of single router) are disconnected from the remainder of the graph (among 33, 118, and 124 nodes in V'). Second, Considering

Table 1: General statistics

		AS1239	**AS3356**	**AS3549**
IGMP cmp	$\#cmp : \|V'\|$	124	118	33
	largest cmp	153	58	276
G^1 graph	$\|N\|$	328	386	308
	$\|N'\|$	5,064	10,610	7,934
	$\|E'\|$	6,859	15,856	12,667
	$\|E''\|$	2,342	3,158	1,342
graph reduction	$\|V''\|$	1,680	3,907	3,366
	$\|V''\|/\|N'\|$	0.33	0.37	0.42

the minimal weighted tree obtained on G^3, we discover that all the edges involved in its construction have a weight of two. This is of the highest importance since it implies that we can reconnect multicast components using only ICMP neighbor and IGMP border IP addresses: those two hop distances correspond to two edges in the set E'' made of composite links.

In order to better understand distances and path diversity in the "meshed logical graph" G^3 before applying Kruskal, we also study the distance distribution between nodes in V'. Fig. 3 provides such a distribution. The horizontal axis gives the distance, while the vertical axis shows the cumulative mass. We observe different behaviors depending on the AS: for AS3549, all computed distances are lower than three hops but its density ($\frac{2\times L'}{V'\times(V'-1)}$) is quite limited (0.14). In AS1239 and AS3356, the collected hybrid graphs are quite dense[2] (0.95 and 0.88, respectively). On the one hand, it potentially implies that using such an additional ICMP information we are able to produce a qualitative inference of the backbone that is likely to be much more connected than a tree. On the other hand, considering the quite large distances (we observe paths up to ten hops long), it also potentially means that MERLIN possibly misses a non-marginal part of the AS due to IGMP filtering. However, (i), large distances may correspond to combination of traceroutes traces (instead of a direct shortest forwarding path), (ii), those results are subject to the potential presence of aliases (implying an overestimation of distances), (iii), it is possible that a small amount of non-responding routers may impact a large amount of shortest paths between IGMP components, and, (iv), there probably exists better paths going through IGMP connected components (here the paths are pure IP level link ones). In a worst case, the existence of a large chain of non-responding multicast routers between distant PoPs in the AS may explain those large distances. This analysis will be extended and deepened in Sec. 4 after applying the alias resolution phase.

[2]Note that density values given here does not have the same meaning than in standard graph theory analysis: indeed, this number rather means (when it tends to 1) that there exists an IP level path between almost each IGMP component pair but those paths may share a common subset of IP level links.

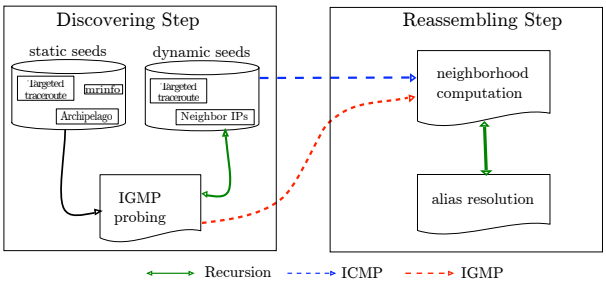


Figure 4: IGMP probing and reassembling - the overall process

3 Reassembling Components

This section aims at describing our strategy for dampening IGMP filtering. Our main objective is to merge the maximum possible number of disjoint IGMP components into a large one. For that purpose, after an IGMP probing phase, we use traceroute like exploration and alias resolution: IP level links and *aliased* IP addresses - forming so routers - fill the gap among disjoint components discovered during IGMP probing.

Except the potential impact of their bias, the choice of a given alias resolution technique should not affect the reassembling strategy described in this section. Indeed, although we consider alias resolution to maintain a router level view of the topology, our reassembling technique can work with any alias resolution mechanism. Future work should reveal how a particular mechanism influences the resulting topology. As we pay great attention to the complexity of alias resolution, we assume that a generic alias resolution process consists in checking sequentially IP addresses pairs. To describe each approach complexity, we introduce the following definitions and quantities: *The IGMP local IP set* M ($|M| = m$): Local IP addresses belonging to collected IGMP routers, $m \approx 2 \times |E| + |E''_b|$; *The IGMP border IP set* B ($|B| = b$): Neighbor IP addresses collected via IGMP routers such that $x \in B$ iff $x \notin M$, $b \approx |E''_b|$; *The ICMP-IGMP neighborhood set* N ($|N| = n$): IP addresses discovered with traceroute that belong to the neighborhood of IGMP routers, $n \leq |E''_n|$; *The ICMP set* T ($|T| = t$): All IP addresses collected with traceroute that do not belong to N or to M, $t \leq |E'|$.

The cardinality of those four sets (m, b, n, t) allows us to accurately describe the cost of the alias resolution phase. We organize those four sets such that they become disjoints. Indeed, in practice, an IP address can belong to several sets, for example an IP address in N comes by definition from T. Thus, we decide to use the following order to classify IP addresses: $M > B > N > T$ meaning that if an IP address belongs to several sets, we classify it uniquely in its first ranked set. Note that an address in B may be used to describe several links when this address is also an ICMP neighbor for another router. Moreover, in this section, we apply an IP2AS filtering on T to focus on the AS of interest and limit the alias space exploration.

Considering the original graph G^1 described in Sec. 2.1, our goal is to progressively "transfer" nodes from N' to N using alias resolution to qualitatively reconnect all original nodes in N between themselves. Thus, we use alias resolution mechanisms to gather IP level nodes in N' in order to provide a connected router level graph. Alias resolution allows for both checking and anti-checking a set of IP interfaces pairs so that we can also easily conclude when IP level nodes are independent in the router level graph.

Fig. 4 summarizes the whole topology collection process. Two steps are required: *Discovering* and *Reassembling*. The Discovering step is based on IGMP probing with Merlin. Merlin is fed with a list of static seeds coming from both the Archipelago dataset [Cla09] and targeted traceroute (using the reachable prefix method described in Rocketfuel [Spr02]) filtered to fit with the AS of interest. Then, to obtain new seeds for Merlin, we launch Paris Traceroute [Aug06] to dynamically discover new seeds (i.e., dynamic seeds on Fig. 4). Those probes target an IP address per each /24 prefix of each router collected with Merlin. This set of traces is of the highest importance for the Reassembling phase since it precisely targets discovered IGMP routers. Once the IGMP router level topology has been collected (the recursive process ends), we have a scattered topology made of disjoint IGMP components, as explained in Sec. 2. More details on the Discovering step are reported in [Mar11].

The next step, Reassembling, aims at reconnecting the IGMP components in order to, at best, obtained a single large and highly connected component. Using IP level links discovered with our traceroutes between distinct IGMP components, the alias resolution step can start. The main challenge here is to identify IP addresses pairs that are good candidates for alias resolution in order to efficiently expand IGMP components and so reassemble them. We do not want to test all possible pairs: only selected candidate IP addresses pairs (using a Neighborhood Computation, see Sec. 3.2) are tested using a standard alias resolution technique, Ally [Spr02]. The alias resolution recursion continues until no new candidates are found. At the end of the process, we can expect to achieve our goal: providing a single large and highly connected graph at the router level.

3.1 Alias Resolution Complexity

In this section, we study the theoretical complexity of alias resolution for reassembling isolated components. Let us denote θ the total number of IP interfaces collected through a topology discovery campaign (in our case, the union of IGMP and ICMP IP addresses). Using a sequential and per addresses pair alias resolution method such as Ally, the total number of pairs to check is $\frac{\theta\times(\theta-1)}{2} = \binom{\theta}{2}$, requiring so an overall complexity in $O(\theta^2)$. It is worth to notice that, in practice, we can rely on the following assumption: "when Ally declares an addresses pair as being aliases, it is sufficient to pick one given of those IP address to represent the alias for the remainder of the sequential process", i.e., cross-validation is useless. Based on this "representative assumption" the overall complexity becomes:

$$\sum_{i=0}^{p} \theta - 1 - \theta_i \leq \binom{\theta}{2}. \quad (1)$$

where θ_i is the cumulative number of IP addresses belonging to valid aliases generated during steps $1, \ldots, i$ and p is the total number of "cluster/alias" in the list (including clusters having a unique IP interface). The number of required steps depends on the number of clusters and their sizes. If the list contains many large aliases, this reduction may be significant. We decide to base our alias resolution campaign on this assumption to limit the number of probes injected in the network and manage alias resolution campaign duration.

Table 3.1 provides those numbers (we use a strict ordering and classification between sets to ensure their empty intersection). Note that we introduce a new notation, $H = B \cup N$, to understand the total neighborhood size of IGMP components (using both the ICMP neighborhood collected through the traceroute campaign and the pure IGMP neighborhood collected through IGMP campaign). The size of T is computed as follows: from the traceroutes we launched towards a given AS, we consider all IP addresses falling in it with a classic IP2AS mapping. We further consider, on each traceroute path, the last IP hop before entering the AS and the first after the AS. This approach is conservative since a border router of a given AS may use the IP address allocation space of its neighbor [Pan10]. Note that in the previous section the T set was not filtered to verify such conditions.

Considering the values given in Table 3.1, if we apply a naive and brute force alias resolution phase, the overall complexity will be equivalent to $\frac{(m+h+t)^2}{2}$ ($\approx$ 24.2 million pairs to investigate for AS3356).

In the following we describe how we designed an efficient and network friendly recursive process that scales

Table 2: ASes of interest with respect to quantities

Sets			AS number		
			AS1239	AS3356	AS3549
$\lvert M\rvert$	=	m	1,789	2,891	2,339
$\lvert B\rvert$	=	b	398	507	418
$\lvert N\rvert$	=	n	898	1,295	718
$\lvert T\rvert$	=	t	554	2,271	589
$\lvert H = B \cup N\rvert$	=	h	1,296	1,802	1,136
$\lvert H \cup M \cup T\rvert$			3,639	6,964	4,064

with our problem. To limit this probing overhead, first we assume that IGMP native aliases (i.e., local interfaces returned by MERLIN) are correct [Mer11] such that the $m \times (m + b)$ sub-cost is useless. It comes that an almost total alias resolution phase requires approximatively $\frac{(h+t)^2}{2} + m \times (t + n)$ alias operations, i.e., $\approx$ 18.6 million of pairs.

Another subset of such space, $m \times (t + n)$, was already investigated thanks to the *IGMP unicast alias resolution*. As shown in Fig. 4, IP addresses extracted from the traceroute traces and mapped to the AS of interest were IGMP probed during the Discovering phase. Although a router provides information only about its multicast enabled links and interfaces, when probed through an unicast interface the router still answers providing the same IGMP reply but with a different source address. Analyzing the collected IGMP replies, it is possible to detect duplicated answers generated by different source addresses: merging those router instances and adding each unicast IP address as an additional interface allows us to consider the space $m \times (t + n)$ already investigated. If inside our IGMP replies dataset there does not exist a reply coming from an address extracted from the traceroute traces, it is reasonable to consider this IP interface as router level independent of any routers collected with IGMP. Note that false negatives are still possible since the IGMP reply could have been filtered in transit. For this reason, we decided to partially re−explore the $m\times n$ space to avoid most likely problems as it will described in Sec. 3.3. Hence the complexity is reduced to $\frac{(h+t)^2}{2}$, i.e, $\approx$ 8.3 million of pairs.

Considering results from Sec. 2.2, we expect that a large portion of the reconnection phase should come from H so that it is possible to ignore T to initiate our alias resolution campaign. Indeed, we showed that it is possible to reconstruct at least a tree capturing almost all IGMP disjoint components using only path offering edges with a distance of two. Moreover, we decide to develop a recursive approach starting from the borders of IGMP components and then recursively extend this approach by considering created alias as new routers. The ICMP neighborhood of these newly created routers allows us to progressively inject IP addresses coming from T in the alias resolution phase: a part of T may be explored when it potentially allows for extending and merging disjoint connected compo-

nents. This approach, developed and formalized in Sec. 3.2, allows us to minimize the impact of T in order to avoid a "flat exploration" of this set. We use IP addresses from T only when they become the border of an extension of the initial graph. At the initialization, our approach just explores the h^2 alias resolution space, i.e., requiring a campaign of ≈ 1.6 million of pairs. In practice, these values represent the worst case: the computation considers all the possible pairs when all targeted IP addresses do not reply to probes, leading each time to timer expiration.

3.2 A Recursive Alias Resolution Approach

In order to reduce the alias resolution space, we decide to not consider all the IP addresses extracted from traceroute traces but only those that are located "close" to routers in the already discovered topology. Hence, our method starts by trying to alias ICMP neighbors and IGMP neighbors (the set H) to generate new routers and, thus, expand each connected component. Then, considering the neighborhood of each new aliased routers, we recursively re-apply the same alias resolution mechanism, progressively expanding the current topology.

The main advantage of such an approach is the alias resolution space reduction by carefully using IP addresses belonging to T. To formally describe our recursive approach, let us introduce some notations. We define the sets H_i as follows:

$$\begin{array}{ll} H_0 = H = B \cup N & \textit{step 0} \\ H_i = N_i \setminus K_{i-1} & \textit{step } i. \end{array} \quad (2)$$

where N_i is the set of IP addresses depicting the neighborhood of new aliases generated during the $(i-1)^{\text{th}}$ step and a set K_i is computed such that $K_i = \bigcup_{j=0}^{i} H_j$. Hence, H_i only contains the new ICMP neighborhood N_i that has not been already considered in the previous steps. Such a set allows us to progressively inject T IP addresses in the process if they belong to the neighborhood of newly generated aliases. Since already explored IP addresses pairs are not checked again, we use K_i to depict the union of all discovered neighborhood since the beginning, i.e., $K_i = B \cup N \cup N_1 \cup ... \cup N_i$. Using those notations, we can easily describe the exploration complexity at each step i, as $O(|H_i| \times |K_i|)$, with the very first step (i.e., $i = 0$) being $O(h^2)$. Hence, the overall complexity becomes:

$$O(h^2) + \sum_{i=1}^{R} O(|H_i| \times |K_i|). \quad (3)$$

where R is the total number of steps performed. It is difficult to predict the value of R since the recursion continues as long as new aliases are generated and their neighborhood is not empty, i.e., while $N_i \neq \emptyset$. In practice, since we prefer to limit the alias resolution propagation error, we decide to use a fixed and constant value for R (see Sec. 4.1).

Note that the representative assumption given in Eqn. 1 is used during the recursive alias resolution phase to fasten the process but it is not taken into account in Eqn. 3 because of its unpredictable nature. Without cross-validation to ensure transitive clusters, the notation $|K_i|$ simply depicts the number of already computed IP clusters in previous iterations.

3.3 Practical Details and Improvements

We provide here some technical details allowing us to improve the alias resolution phase. Two kinds of improvements are made: (i), fasten the alias resolution process with information collected by active measurements, (ii) reducing the complexity of the alias resolution between recursive stages.

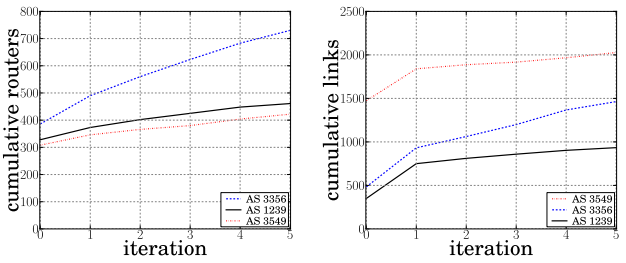
First, (i), as described by Pansiot and Grad [Pan98] we complete the alias resolution with an address-based method: the source sends a UDP probe with a high port number to the routers interface X. If the source address of the resulting "Port Unreachable" ICMP message is Y, then X and Y are *aliased* in the same router. This kind of information can be retrieved from our traceroute campaign. We further modify Paris Traceroute so that it clearly indicates whether the destination has been reached or not. This way, when targeting one interface of a known IGMP router, the corresponding traceroute clearly shows that the router is reached and reports a final IP address X. If X does not appear among the set of know interfaces, we can safely add it as a new interface of the targeted router. Although limited in number, by analyzing those modified traceroute traces we were able to discover additional unicast interfaces of IGMP routers: 8 for AS3549, 22 for AS3356, and 4 for AS1239.

Second, (ii), we make use of two *non−checking reduction*'s strategies: we avoid checking a pair of IP addresses if (a) they appear in the same traceroute trace, (b) or they are contained in the same subnet */31*. Future works will investigate more sophisticated approaches (for example exploiting the IP subnet allocated to IGMP layer−2 devices [Mer10]). Moreover it is also possible to carefully manage the alias resolution exploration space between each recursive stage. Between each recursive step, an additional linking stage is performed: when a traceroute reveals a direct connection between two router level nodes, a new link is added in E. Since the neighborhood information obtained with IGMP queries could be incomplete (uni-

cast lacks - see [Mer11] - or even empty for ICMP aliases), it is possible that two consecutive IP addresses in the traceroute traces belong to two known router level nodes without these routers revealing the link by themselves. In such a case, even if we do not know one of the IP interfaces involved in the link, we can infer a new link between two router level nodes and possibly reconnect IGMP disjoint components. Let us denote $N(r)$ the ICMP-IGMP neighborhood set of a given router level node r. When traceroute discovers an ICMP link between two router level nodes a and b thanks to the IP i on b, then the set $N(a)$ excludes i. Between multicast nodes, this kind of links reveals unicast lacks as explained in Sec. 4.2. Moreover, for each connected component A, we apply a second process: for each IP address belonging to the set $N(a),\ a \in A$, we try to alias IP addresses in $N(a)$ to the M_A set (i.e., a representative set of IGMP local IP addresses belonging to IGMP routers of the component A and linked to a). This set is specific for routers in A and, to reduce the alias resolution complexity, we only pick one IP address per considered router. When an alias is inferred with another router $b \in A$ ($x \in N(a),\ y \in b \mid x \in b$), we add x to b and we exclude it from $N(a)$. Indeed, since multicast IGMP alias may miss a unicast interface, it is possible that the IP address x is "internal" to the connected component A (it is not necessary to consider it for the reconnection between disjoint connected components). This step allows us to possibly reduce the IP level neighborhood of a and, so, the overall complexity of subsequent alias resolution procedures (in particular when $|M_A| < N_i$ and if an alias is generated). It also allows for enforcing the exploration of the $m \times n$ alias space and in a lower extent the $m \times t$ one.

4 Reassembling Evaluation

For this evaluation, we targeted the same ASes of Sec. 2. We followed for each AS the overall methodology depicted in Fig. 4. First, a MERLIN probing campaign is launched towards each AS from all the vantage points. While Paris Traceroute campaigns were launched from all the vantage points towards multiple interfaces of each router (a single IP address for /24), the alias resolution phase, that makes use of the information retrieved from the traceroutes, is performed for each AS by a single monitor. The main reason is to avoid interference among the monitors when they try to infer alias in the same AS topology: it could result in the exceeding of the ICMP rate-limiting threshold and makes the routing domain silencing our probes. In our implementation of the reassembling strategy, we make use of Ally for performing the alias resolution.



(a) routers (b) links

Figure 5: Routers and links created at each step of the reassembling process

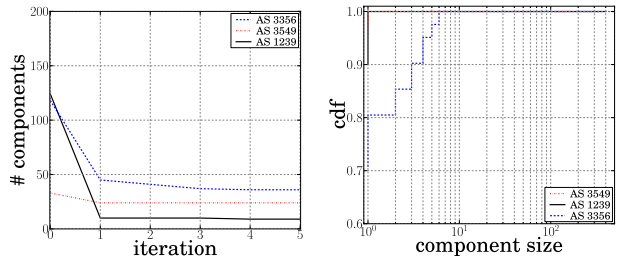
4.1 Alias Resolution Stage

In this section, we focus on the number of generated aliases and their impact on the reassembling process. We consider a network component as a node only if it has at least two IP addresses. It implies that, when analyzing the graph evolution during recursive iterations, we do not consider IP addresses that are not aggregated in an alias (only positive ones are considered in the router level node set).

We aim at demonstrating that most of the alias phase benefit comes from first recursion stages and requires a reasonable amount of time. In Sec. 2.2, we demonstrated that using only a set of minimal distances of two hops between IGMP components makes possible to reassemble almost all disjoints fragments. We should thus be able to provide a fully connected topology after a single iteration of the reassembling process. Indeed, since we create new routers/aliases all along the border of disjoint components, we should be able, in the best case, to reduce the distance between components by two hops per iteration.

However, as we only consider new aliases in the graph reconnection, results given here represent a lower bound to study the "quality" of our reassembling scheme. Indeed, IP interfaces involved in two hops or longer paths (providing so minimal distances between IGMP fragments) must generate positive aliases in order to be considered in the graph reconnection. Sec. 4.2 provides a more friendly perspective by considering as potential nodes all network components tested during the alias exploration phase. Our goal here is rather to focus on positive alias resolution performance. We also aim at showing that our reconnection strategy is able to quickly reduce the number of disjoint components by generating aliases and revealing so lower distances between IGMP fragments.

Fig. 5 shows the cumulative number of nodes (Fig. 5(a)) and links (Fig. 5(b)) created at each step of our recursive process (the horizontal axis). Note that "iteration 0" on Fig. 5 refers to the situation be-



(a) Disjoint components evolution (b) Component size distribution

Figure 6: Connected components analysis

fore applying the reassembling process: it provides the original IGMP graph after adding some traceroute IP interfaces to IGMP routers (IGMP alias unicast resolution) and after correcting one hop distance with the process explained in Sec. 3.3.

The number of new routers and links created at each iteration seems to slowly decreases: in particular, for all evaluated ASes, at least as many links are introduced in the first iteration as in subsequent iterations. AS3356 shows a specific behavior: it seems more subject to positive alias generation. For other ASes, the gain seems to become marginal after three or four iterations: the number of new alias slows down and most of the links have been discovered earlier. Based on this observation and the cumulative bias introduced by Ally, we decide to stop our recursive process after five iterations. Note that a number of k iterations is able to ideally solve distances of $2 \times k$ hops. Intuitively, a distance of k corresponds to a potential reconnection path made of k hops (i.e., a path of k links allowing to merge several IGMP components). In Sec. 4.2, we decide to cap the number of iterations to $k = 2$ in order to limit the number of false information potentially generated by Ally, and a priori, reconnect all IGMP fragments. Most of the benefits comes from the first two iterations: it allows for solving four hops distances.

Fig. 6 provides an analysis of the evolution of disjoint IGMP components over the recursive process. In particular, Fig. 6(a) shows how the number of disjoint components decreases over time. According to results obtained in Sec. 2.2, the first iteration should be almost sufficient to reconnect the graph. However, here we consider only reconnection paths involving aliases created during the previous iterations. Thus, results provided here gives a lower bound describing only the impact of generated alias. It allows for understanding whether reconnection paths are subject or not to potential alias. Considering this point of view, the number of IGMP components decreases from 33 to 24 for AS3549, 118 to 45 for AS3356, and 124 to 10 for AS1239. The reduction level highlights each network specificity regarding our measurements: AS1239, and in a lower extent, AS3356 offers a good alias performance, a significant number of alias is generated during the first step allowing so to fix most of two hop distances. On the contrary, AS3549 does not provide such efficient results: either a small amount of IP addresses we retrieve forms aliases, or Ally does not work within this AS. The impact of the alias resolution phase reveals the level of dependency among forwarding paths discovered through our traceroute campaigns. For AS1239, it seems that almost all two hop distances are subject to alias favoring so the almost complete reconnection during the first iteration. Although limited, further iterations still induce the merging of IGMP components using solely alias.

Since our goal is to reconnect the components while preserving the *reliability* of the preliminary topology provided by the IGMP probing phase, it is important to not underestimate the bias introduced by the aliasing resolution technique: Ally may generate false negatives and false positives [Key10]. Although in both cases we face inaccuracies, the false positives have the worst impact on the rebuilt topology. Indeed, if a new aliased router consists of false positive IP addresses, the error will affect its neighborhood and thus the way the overall topology grows. Due to the cumulative nature of this error, we decide to consider as final topologies the ones obtained after only two iterations. Note that this seems also reasonable in the light of results given in this section: most of the reconnection paths involving useful alias results from the first iteration. Compared to Fig. 2, i.e., the same distribution but before applying the reassembling process, Fig. 6(b) shows the efficiency of the alias process used for our reassembling technique. For instance, before applying it, the largest component in AS1239 was made of 153 routers. After the second iteration, the largest component is made of 393 nodes and only 9 components (made of a single router) are still isolated. On other ASes such as AS3356, we can notice that some low distance reconnection paths do not seem to involve alias so that we still have a significant number of isolated IGMP fragments after studying four hops paths. Fig. 5 and Fig. 6 showed us that strategic reconnection paths (the ones exhibiting short distances lower than four hops) involve generally a great number of alias demonstrating so the good coverage and the dependence among IP interface level nodes we consider. In particular, it highlights the efficiency of using H: it allows for considering new low distances and eventually merging previous dependent ones thanks to a great number of generated aliases during the first iteration. In AS1239, we observe that a large proportion of aliases are created using B involving so multicast neighbors. Finally, it seems also to indicate good den-

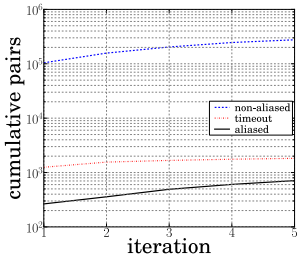


Figure 7: Reassembling AS3356: pairs investigated

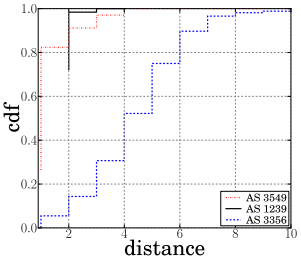


Figure 8: Distance Analysis

sity properties considering the inter-connection subgraph. In the next section, we will refine this analysis by considering all independent IP interfaces as nodes to better understand the reconnection efficiency of our approach.

Fig. 7 gives an insight into the practical efficiency of our reassembling technique, focusing on AS3356 (the worst case). The figure provides the cumulative number of IP addresses pairs tested for alias at each step of the process. We label as *alias* (plain line) pairs of IP addresses that are declared as aliases by Ally. On the contrary, *non-alias* (dashed line) are declared as anti-aliases by Ally. Finally, *timeout* (dotted line) refers to IP addresses pairs triggering a timeout: no decision has been made by Ally due to the absence of replies by one of the targeted addresses. Note that once an IP address generates a timeout expiration, we remove it from the set of IP addresses to explore. While positive or negative aliases are generally quickly inferred (between one and two second), timeout expirations implies to wait between two and four seconds. Obviously, the number of positive alias is lower than the number of negative alias or even timeout probes. We can observe that the number of pairs investigated remains significant in last iterations. It seems that the recursion process continues to discover an important IP neighborhood on newly introduced alias. The number of IP pairs to investigate stay almost stable after the first iteration (it only slowly decreases after the first iteration): while the first iteration costs obviously a great part of the alias exploration, we did not expect such a "costly evolution" during subsequent steps. Note that Fig. 7 also helps us understanding the save in the alias space exploration due to reduction offered using the representative assumption (Eq.1), our set our "non-checking" rules and timeouts. Indeed, only $105,739$ pairs are explored among $h \times (h-1)/2 = 1,622,701$ theoretical possible pairs leading so to a save of almost 95%.

4.2 Final Topology Analysis

On the contrary to Sec. 4.1, here we take into account all network components that appear to be independent: router level nodes such as IGMP routers and generated aliases as before, but we also consider a subset of single remaining IP interfaces (the ones that have been involved in the alias exploration). Indeed, IP addresses that do not form aliases while they have been tested with others should be separate entities forming so independent nodes in the final graph. Our goal is then twofold: (i) show that we provide large and well connected graphs using both positive and negative alias, and (ii) validate our expectations about the multicast use in several Tier-1 backbone.

Fig. 8 shows the impact of our recursive alias resolution approach on preliminary distances computed between native IGMP components. For this analysis, we consider the final resulting graph and apply the methodology described in Sec. 2.1 to obtain the G^3 graph. Although, most of IGMP components are now reconnected, we continue to distinguish IGMP native disconnected components from the rest of the graph (newly introduced alias and IP level nodes). Compared to Fig.3, we notice a great shift towards lower distances: even for the worst case (AS3356), we observe that almost 80% of distances are now lower than six hops instead of approximately 60% before alias computation. It is also worth to notice that the alias resolution phase allows one to compute new distances and can make the G^3 graph denser. When several IP addresses are merged into a given alias, the distance resulting from a combination of traceroute traces may decrease: on the contrary, when it results from a unique direct forwarding trace, the distance is fixed. On AS3356, although most of distances decreases, maximal distances are incompressible: they result from direct and unique forwarding traces. However, note that this "distance oriented graph" is so dense that large distances are not necessary to make the reconnected graph meshed.

Table 3 gives an overview of the final graph characteristics. In particular, we focus on its multicast/unicast structure. Indeed, from information retrieved through IGMP probing, we can classify routers into several category: ***IGMP*** stands for native IGMP routers, ***MA*** for Multicast Alias, ***UA*** for Unknown Alias, ***MIP*** for Multicast IP (coming from the B set), and ***UIP*** for Unknown IP (coming from the $N \cup T$ set. Note that we consider a subset of IP level nodes (coming from the $H \cup T$ set) as router level nodes when they have been tested during the alias resolution phase. Indeed, it does not generate false positive nodes (i.e., we do not consider two IP addresses as being separate

Table 3: Global statistics on the final graph

AS number	Routers #total: 1076, 1958, 1363 IGMP	MA	UA	MIP	UIP	Links #total: 1692, 4091, 4929 $IGMP_1$	$IGMP_2$	$ICMP_1$	$ICMP_2$	Graph analysis D	Δ	$\varnothing$	Inter-cmp. Vision d	δ
AS1239	328	34	40	51	623	335	57 (337)	957	63	3.145	0.003	10	3.628	0.005
AS3356	386	50	124	418	980	372	1,320 (1,489)	2,028	202	4.179	0.002	14	4.732	0.003
AS3549	308	17	41	376	621	567	568 (602)	2,610	1,150	7.233	0.005	9	8.269	0.008

nodes as long as they belong to the same router) because all those IP addresses have been checked between themselves[3]. A Multicast Alias (MA) is computed as such only if at least one IGMP IP address belonging to it (coming from the B set), otherwise the resulting alias is considered as an UA (we cannot infer its nature when we only find IP addresses in the set $N \cup T$).

In Table 3, we also provide a detailed analysis about the nature of retrieved links. We classify links according to four categories: (i) IGMP links between two IGMP native router ($\boldsymbol{IGMP}_1$), (ii) IGMP links (resulting from IP addresses in B) between multicast components (IGMP, MA, MIP nodes but with one IGMP node involved at maximum - $\boldsymbol{IGMP}_2$: the number in brackets being the total number of such links and the value given as first is related to links involving MIP nodes), (iii) ICMP links between nodes whose nature is unknown (UIP and UA nodes, $\boldsymbol{ICMP}_1$), and, (iv), ICMP links between multicast nodes (IGMP and MA connections involving at least one IGMP router, $\boldsymbol{ICMP}_2$). Depending on the AS of interest, those values are really fluctuant. In particular, considering unicast links between multicast nodes ($ICMP_2$), we can observe that, while their proportion is lower than 4% for AS1239 and AS3356, they represent more than 20% of links for AS3549. Moreover, among the 20% of such links discovered in AS3549, the vast majority of them are unicast links between IGMP native routers that we can discover with the IGMP unicast alias resolution process. This large difference suggests that the multicast is not deployed in the same way within this AS. A such large proportion implies that the multicast and unicast forwarding tables seem to diverge. We can also observe that this AS graph is much more dense (inside and outside IGMP native components) than the two other ASes graphs. Indeed, the two last parts of Table 3 shows the connectivity of resulting topology at the global scale (*Graph analysis*) and at the inter-component scale (*Inter-cmp vision*). We provide average degree (D, d for respectively global and inter-component scales), density (Δ, δ for respectively global and inter-component scales), and diameter ($\varnothing$) computed on the three ASes. The AS3549 graph seems sufficiently dense to deploy multiple topology in order to distinguish unicast and multicast traffic.

Based on the overall analysis of Table 3, we can conclude that, while best results in terms of multicast structure are achieved on AS1239 (we have a low number of unicast links and a high proportion of border IGMP IP addresses involved in terms of generated alias), we obtain a very meshed networks for AS3549 that seems to present high redundancy patterns to increase failure tolerance but that is also more subject to false positives (we need to intensify the $m \times n$ alias exploration to better integrate the large presence of unicast links).

To push further our connectivity analysis, we decide to remove unicast links and links not involving multicast components (at least on one extremity). We want to understand if the final topology is able to provide at least a multicast tree (we consider $ICMP_1$ links as potential multicast links if they inter-connect MA aliases or MIP). In practice, we only use ICMP links when we cannot determine their nature because of the lack of native IGMP information: when we extend the topology, MA nodes do not provide their multicast neighborhood such as IGMP nodes, so that we cannot draw any conclusion on their connectivity nature.

Considering the three ASes of interest and their resulting reduced edges set, we still have connected graphs that are likely to be fully multicast. It seems to confirm our first expectation: there exists at least a connected multicast structure in the backbone of targeted ASes. Depending on the routing strategy, this structure is then more or less similar to the complete network backbone that includes unicast-only components.

The number of unicast routers (UA) and even more the number of unicast links between multicast routers ($ICMP_2$) could indicate that there are some unicast-only components inside the multicast core. This could be explained by the following reasons: (i) ECMP: multipath next hop entries dedicated to the forwarding of multicast traffic can be a subset of the unicast table. (ii) IGP weight and "backup links": according to the destination, some forwarding tables can diverge because of links only used to reroute unicast traffic (and

[3]In practice, we may have some false positive nodes because we do not explore all the $m \times n$ space. However, if unicast links between multicast routers that we do not discover thanks to IGMP unicast alias resolution are considered as exceptions, false positives cases should be very marginal.

probably break the PIM tree in case of failures while preserving unicast traffic to support lower capacity). (*iii*) Multi-topology of IS-IS: it is possible that some IS-IS routing domains use two routing maps to distinguish multicast and unicast traffic. (*iv*) Unicast Border Routing: links to only unicast nodes/leafs are not multicast enabled. The BR (Border Router to limit internal areas) and ASBR (AS Border Router dealing with BGP routing) belonging to a given routing domain may distinguish multicast from unicast traffic. A non multicast LAN could be connected to several multicast routers (for redundancy reasons) by unicast only interfaces. (*v*) Traceroute limitations (MPLS, load balancing, etc): traceroute may reveal false links. (*vi*) Ally limitations: Ally may reveal false aliases [Key10].

5 Conclusion

In this paper, we improved IGMP based topology discovery by presenting a new hybrid tool based on Merlin for circumventing IGMP filtering: a subset of multicast routers does not reply to IGMP probing causing therefore the partitioning of the collected graph into several disjoint connected components. While the Merlin probing stage collects a set of disjoint IGMP components, we use traceroute and alias resolution techniques to reassemble them at the router level and, thus, extend the mapping of the targeted routing domain. After having defined a hybrid graph model capturing the heterogeneity of the collected data (at both router and link level views) to better understand the impact of IGMP filtering, we develop a recursive reconnection approach based on the neighborhood proximity to limit the complexity of the alias resolution phase. Our strategy allows for reducing the amount of false inferred links and routers introduced by current topology discovery techniques: although it is likely that the inferred graph forms a partial view of the real targeted network, we favor false negatives amongst false positives. Our approach is particularly efficient for discovering the multicast enabled backbone of large ASes. Indeed, our probing campaigns show that IGMP probing is a relevant approach to initiate the capture of the core of large multicast ASes.

References

[Mer11] P. Mérindol, B. Donnet, J.-J. Pansiot, M. Luckie, and Y. Hyun. MERLIN: MEasure the Router Level of the INternet. In *Proc. 7th Euro-NF NGI*, 2011.

[Aug06] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with Paris traceroute. In *Proc. ACM/USENIX IMC*, 2006.

[Cla09] k. claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov. Internet mapping: from art to science. In *Proc. IEEE CATCH*, 2009.

[Don07] B. Donnet and T. Friedman. Internet topology discovery: a survey. *IEEE Communications Surveys and Tutorials*, 9(4):2–15, 2007.

[Don06] B. Donnet, P. Raoult, T. Friedman, and M. Crovella. Deployment of an algorithm for large-scale topology discovery. *IEEE JSAC*, 24(12):2210–2220, 2006.

[Gun08] M. H. Gunes and K. Sarac. Resolving anonymous routers in the Internet topology measurement studies. In *Proc. IEEE INFOCOM*, 2008.

[Key10] K. Keys. Internet-scale IP alias resolution techniques. *ACM SIGCOMM CCR*, 40(1):50–55, 2010.

[Mar11] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, and J.-J. Pansiot. Topology discovery at the router level: a new hybrid tool targeting ISP networks. *IEEE JSAC*, 29(6):1776–1787, 2011.

[Mer10] P. Mérindol, B. Donnet, O. Bonaventure, and J.-J. Pansiot. On the impact of layer-2 on node degree distribution. In *Proc. ACM/USENIX IMC*, 2010.

[Mer09] P. Mérindol, V. Van den Schriek, B. Donnet, O. Bonaventure, and J.-J. Pansiot. Quantifying ASes multiconnectivity using multicast information. In *Proc. ACM/USENIX IMC*, 2009.

[Pan98] J. J. Pansiot and D. Grad. On routes and multicast trees in the Internet. *ACM SIGCOMM Computer Communication Review*, 28(1):41–50, 1998.

[Pan10] J.-J. Pansiot, P. Mérindol, B. Donnet, and O. Bonaventure. Extracting intra-domain topology from `mrinfo` probing. In *Proc. PAM*, 2010.

[Spr02] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *Proc. ACM SIGCOMM*, 2002.