Travelling Without Moving: Discovering Neighborhood Adjacencies

Jean-François Grailet, Benoit Donnet
Université de Liège, Montefiore Institute, Belgium

***Abstract*—Since the early 2000's, the research community has explored many approaches to discover and study the Internet topology, designing both data collection mechanisms and models.**

In this paper, we introduce `SAGE` (Subnet AGgrEgation), a new topology discovery tool that infers the hop-level graph of a target network from a single vantage point. `SAGE` relies on subnet-level data to build a directed acyclic graph of a network modeling how its (meshes of) routers, a.k.a. neighborhoods, are linked together. Using two groundtruth networks and measurements in the wild, we show `SAGE` accurately discovers links and is consistent with itself upon a change of vantage point.

By mapping subnets to the discovered links, the directed acyclic graphs discovered by `SAGE` can be re-interpreted as bipartite graphs. Using data collected in the wild from both the PlanetLab testbed and the EdgeNet cluster, we demonstrate that such a model is a credible tool for studying computer networks.

I. Introduction

For the past twenty years, multiple approaches to study the Internet topology have been investigated [1], [2]. Besides traditional active probing techniques revealing IP interfaces [3], [4], [5], [6] and routers [7], [8], [9], [10], several intermediate levels have been explored over time, such as Internet eXchange Points (IXPs) [11], [12], Points-of-Presence (PoPs) [13], [14], and subnets (short for subnetworks) [15], [16].

Among these intermediate levels, the subnet-level showed potential for broader topology mapping. In particular, `TreeNET` [17] collects routes towards subnets first discovered via `ExploreNET` [16] in order to build a tree-like map of the target network, an approach which can notably be used as a space search reduction scheme for alias resolution [10]. However, both `ExploreNET` and `TreeNET` rely on assumptions that can be easily violated by load balancing architectures [18]: asymmetrical paths typically induce `ExploreNET` into chunking subnets, while symmetrical paths are difficult to model in the tree-like maps built by `TreeNET`.

`WISE` [19] handles the subnet inference challenges induced by load balancers by carefully reviewing the interfaces it discovers in a target domain prior to aggregating them in subnets. Additional researches with `WISE` thoroughly assessed the potential of ***neighborhoods***, i.e., network locations bordered by subnets located at most one hop away from each other, for topology discovery [20]. An individual neighborhood corresponds, in practice, to a single (mesh of) router(s) behaving as a single hop in `traceroute` measurements. Though `WISE` only discovers individual neighborhoods, it is possible to use them as the building blocks of a hop-level graph.

In this paper, we introduce `SAGE` (**S**ubnet **AG**gr**E**gation), a new topology discovery tool that systematically builds the hop-level graph of a target domain as a neighborhood-based DAG (Directed Acyclic Graph), using a single vantage point. In such a graph, vertices model neighborhoods while edges account for the links that connect them together. `SAGE` infers the DAG of a network by building neighborhoods with subnet-level data first collected with `WISE`, by finding their adjacencies with backward `traceroute` probing, and by using alias resolution to detect convergence points of load balanced paths. It also identifies, when possible, the subnets acting as the links between adjacent neighborhoods. We designed `SAGE` to capture exhaustive maps of intra-domain topologies and study their hop- and subnet- levels.

This paper provides two key contributions. First, we describe `SAGE` and the challenges it addresses and demonstrate it can capture consistent pictures of intra-domain topologies, regardless of the vantage point. Second, we propose to turn neighborhood-based DAGs into ***bipartite graphs***. In a bipartite graph, vertices are divided into two disjointed sets (or parties), $\top$ and $\bot$, so that every edge connects a vertex in $\top$ to one in $\bot$. Bipartite graphs have been widely studied by the scientific community [21], [22], [23], and in particular, a Layer-2 (data link) device – router bipartite model for the Internet has been previously explored [24]. With `SAGE` data, networks can be studied as neighborhood – subnet bipartite graphs, and we argue that the properties of such graphs are consistent with previous research and the types of the measured networks. The source code of `SAGE`, our tools for bipartite analysis, and our dataset are all publicly available on GitHub. [1]

The rest of this paper is organized as follows. First, Sec. II presents `SAGE` in details: its core concepts, the challenges it addresses, and its workflow. Second, Sec. III validates `SAGE` on two groundtruth networks and shows it can discover similar graphs for a given network when the vantage point changes. Third, Sec. IV discusses the potential of the neighborhood – subnet bipartite model using data collected with `SAGE` in the wild. Finally, Sec. V concludes this paper by summarizing its main contributions.

II. SAGE

A. Ideas and Previous Work

Initially, the subnet-level was considered by the research community as a complementary view to the more traditional

978-3-903176-40-9

[1] https://github.com/JefGrailet/SAGE

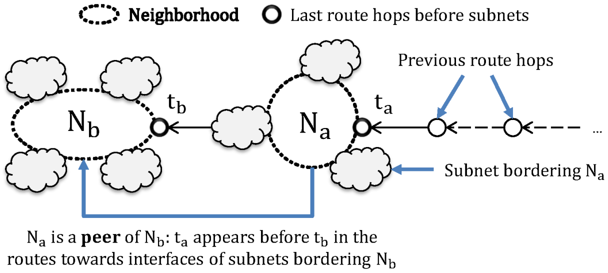


Fig. 1. The concepts of neighborhood and peer.

router-level view of the Internet [15], [16]. However, recent researches showed that subnet-level data can hint at the underlying topology of a target network, i.e., its hop-level adjacencies [17], [10], [20]. A common observation is that the `traceroute` records collected for separate subnets, usually by probing one or a few interfaces of each, can be very similar. More precisely, the last hop(s) of said routes (i.e., before reaching the subnet interfaces) tend to be identical, therefore implying that the corresponding subnets are reached through the same path(s) and at the very least through the same last hop (mesh of) router(s). This common last hop is conceptualized as a ***neighborhood***, and is formally defined as a network location bordered by subnets located at most one hop away from each other. In practice, a neighborhood might be either a single router or a mesh of routers, potentially involving Layer-2 (data link) equipment, and is typically identified by the router interface that appeared as the last hop towards its surrounding subnets (i.e., before the subnet interfaces).

Neighborhoods were first explored with `TreeNET` [17], which uses the complete Paris `traceroute` [3] records towards each subnet it previously inferred to build a tree-like map of a target domain where leaves correspond to subnets while internal nodes model neighborhoods. However, a collection of `traceroute` paths is usually closer to a directed acyclic graph rather than a tree [25]. To overcome the limitations of the tree model, the notion of neighborhood can be complemented with the concept of ***neighborhood peer*** (which will be subsequently simply denoted as ***peer***): given two neighborhoods N_a and N_b identified by the last hops towards their subnets t_a and t_b (respectively), N_a is a ***peer*** of N_b if and only if the hop prior to t_b in the routes to subnets of N_b is t_a. Fig. 1 illustrates this definition, as well as the concept of neighborhood, i.e., a single hop delimited by subnets located at most one hop away from each others.

The definition of peer naturally allows a single neighborhood to have multiple predecessors, while an implication of the tree-like model of `TreeNET` is that a neighborhood can only have one predecessor in the tree. When a neighborhood has no peer at a distance of one hop prior to the router interface identifying it, a best effort approach consists in looking for one or several ***remote peers***, i.e., the closest route hops which the associated IP interfaces identify other neighborhoods.

Both the concepts of neighborhood and peer have been thoroughly evaluated with `WISE`, a subnet inference tool initially designed to take account of various effects of traffic engineering to accurately discover subnets in linear time [19], [20]. `WISE` was implemented to conduct Paris `traceroute` measurements towards subnet interfaces (up to five in each subnet) to subsequently infer neighborhoods and their peers, both algorithmical steps being run after the end of subnet inference. Not only `WISE` allowed to thoroughly assess the consistency of the concept of neighborhood on groundtruth networks, but it also led to one major result: using data collected in the wild from the PlanetLab testbed with `WISE`, it was showed that neighborhoods have high odds of having peers located at most one hop away, implying the corresponding (meshes of) routers are separated by a single network link. In particular, for a subnet picked at random, the odds of the bordered neighborhood having a peer at most one hop away ranged from 80% to more than 90% depending of the target network. A considerable number of neighborhoods having several peers was also observed.

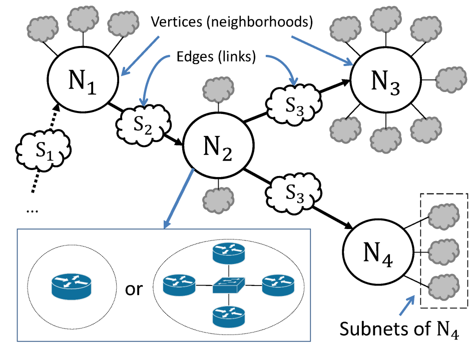


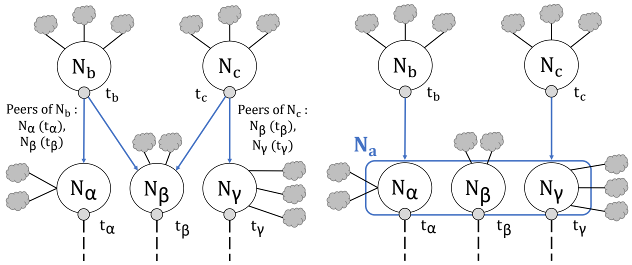
Fig. 2. Toy network viewed as a neighborhood-based DAG. Edges are mapped to subnets, with S_3, a small LAN, being mapped twice.

The study of neighborhoods and their peers with `WISE` [20] showed that both concepts are consistent enough to consider building a ***neighborhood-based DAG***. In such a graph, vertices model neighborhoods while the edges model how they are located in respect of each other. I.e., given two neighborhoods u and v, if u is a peer of v, then the DAG including u and v will include the edge $u \rightarrow v$ as well. Ideally, as subnets can correspond to links or LANs in the Internet, it is even possible to enrich this neighborhood-based view by associating a subnet with each edge, resulting in a neighborhood – subnet topology that can be used to study the internal routing of a network. Fig. 2 shows a toy network viewed as a neighborhood-based DAG, where subnets S_1, S_2, and S_3 are mapped to edges. The side frame shows an example of what a single neighborhood (vertex) can be made of.

B. SAGE Challenges

We designed `SAGE` (**S**ubnet **AG**gr**E**gation) to concretize the systematic construction of neighborhood-based DAGs by reusing the concepts previously explored with the help of `WISE` [20]. `SAGE` is named after the way it first creates the neighborhoods, i.e., by aggregating subnets on the basis of the last hop(s) observed prior to their interfaces.

Naturally, such an objective brings several challenges. The first challenge arises from the inference of neighborhoods



(a) Before alias resolution (b) After alias resolution

Fig. 3. Discovering a convergence point with `SAGE`.

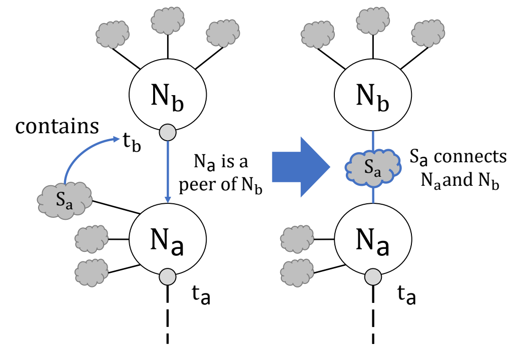


Fig. 4. Mapping links and subnets in a neighborhood-based DAG.

themselves: in some cases, the end of the route towards interfaces of a given subnet can consist, for instance, of one or several anonymous hops or of cycling interfaces. It is therefore impossible to aggregate the corresponding subnet with other subnets on the basis of a single router interface belonging to the last hop (mesh of) router(s). For such subnets, `SAGE` relies on a best effort approach: it aggregates subnets on the basis of the part of the routes towards their interfaces containing the last non-anonymous and non-cycling hop. For example, if several such routes consist of A, B, X, X where X is an anonymous interface, a neighborhood can be built with subnets whose the routes end with B, X, X. We denote such neighborhoods as ***best effort neighborhoods*** in opposition to regular neighborhoods that are identified by the last router interfaces observed before reaching their surrounding subnets (such as t_a and t_b in Fig. 1). Due to not being denoted only by router interfaces, best effort neighborhoods cannot be peers by design and can only act as endpoints of the final graph.

The second issue `SAGE` needs to address comes from load balancing architectures. When a neighborhood features several peers, there is indeed no guarantee that the interfaces associated to them are strictly from distinct devices. For instance, a neighborhood can first appear as several neighborhoods because the probes reached the respective associated interfaces from distinct paths instead of always reaching the same interface. In other words, a neighborhood can include a ***convergence point***, i.e., a device where distinct paths converge that is best identified by an alias list. To address this challenge, `SAGE` identifies router interfaces that might belong to convergence points and conduct alias resolution on them. After listing the peers of all neighborhoods, `SAGE` gathers the lists with multiple peers and merges them when they share common interfaces, relying on the property of ***alias transitivity***. This property works as follows: if we have three interfaces A, B, and C, if A and B are aliases and if B and C are aliases too, then A and C are aliases as well. This trick allows `SAGE` to resolve the largest hypothetical alias lists to avoid testing the same alias multiple times. The alias resolution itself is performed with a fingerprint-based framework [10]. When a convergence point is discovered, `SAGE` proceeds to merge together the associated neighborhoods.

Fig. 3 shows a toy topology where `SAGE` is able to discover a convergence point by applying subsequently alias transitivity and alias resolution. In this toy example, the two neighborhoods N_b and N_c initially have respectively the lists of peers t_α, t_β, and t_β, t_γ (Fig. 3a). By aggregating those peers to create the largest hypothetical alias $\{t_\alpha, t_\beta, t_\gamma\}$, `SAGE` can recover the (initially split) neighborhood N_a depicted in Fig. 3b as long as t_α, t_β, and t_γ are aliased together.

The final issue, which is more an opportunity than a challenge, lies in mapping subnets to the edges of the final graph. Indeed, in the real world, subnets can notably act as point-to-point links between two devices (typically, with a /30 or /31 subnet). For each edge it infers, `SAGE` tries to find the subnet that could correspond to the practical network link. By doing so, `SAGE` not only outputs a true picture of the hop-level adjacencies, but also complements it with subnet-level data by design, resulting in a neighborhood – subnet perspective of the network. This task simply consists of looking at the router interface(s) denoting a neighborhood and finding the subnet whose the prefix encompasses (one of) said interface(s). Fig. 4 shows a simple example: given two juxtaposed neighborhoods N_a and N_b, respectively identified by router interfaces t_a and t_b, N_a being closer to the vantage point, a subnet S_a bordering N_a encompasses t_b. As a consequence, S_a can be considered as the subnet connecting N_a and N_b together. In practice, the subnet that connects two neighborhoods does not always appear around the neighborhood closest to the vantage point, and can also border a third neighborhood, depending on the subnet size (a large subnet can be mapped to several edges) and/or the internal routing of the target network.

C. SAGE Workflow

In order to build a neighborhood-based DAG, `SAGE` requires subnet-level data to begin with. It is therefore built on top of `WISE` [19], [20] so that the former can reuse the subnet inference methodology of the latter. A key advantage of such a methodology, in addition to handling various effects of traffic engineering, is its ability to scale linearly with the number of target IP addresses. As a result, it can easily discover subnets on a target network using a single vantage point. Previous work is either not as fast [19] or involves multiple vantage points, such as Cheleby [26]. Having efficient subnet inference makes it possible to run `SAGE` from a single vantage point as well, as it only requires some lightweight additional probing to eventually build a graph.

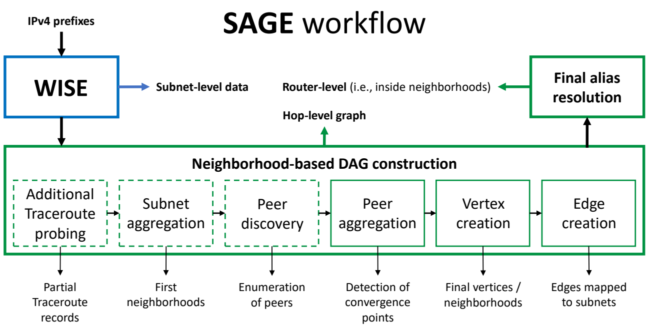


Fig. 5. SAGE workflow. Green blocks show our contributions, with dashed green blocks corresponding to steps previously experimented with WISE [20].

Having discovered subnets with the WISE methodology, SAGE first collects additional traceroute data before it starts building the DAG. It therefore first processes the subnet-level data to make a census of all router interfaces that could act as identifiers for (regular) neighborhoods, i.e., all final hops before subnet interfaces that are neither anonymous nor cycling hops, also called ***peer addresses***. It subsequently schedules *partial* backward (Paris) traceroute measurements towards a subset of interfaces from each subnet [2] in order to collect paths that will help it discover peers. I.e., for each subnet interface, SAGE sends a first probe with a TTL value corresponding to one hop prior to the last non-anonymous and non-cycling hop towards the target. If a reply arrives, SAGE checks if it comes from a peer address. If so, it already has the data it needs to discover a peer and can stop probing, meaning the measurement is *partial*. Otherwise, it keeps probing by decreasing the TTL value until it reaches 0 or until a remote peer can be found. By doing so, SAGE reduces the traffic it sends towards the target network, and rarely collects full paths. For efficiency's sake, this preliminary traceroute probing is also performed with multithreading.

Once the traceroute data is available, SAGE builds the graph step by step. It first aggregates subnets based on their last hops into neighborhoods and uses the additional traceroute data to discover their respective peers. If some neighborhoods feature multiple peers, SAGE applies the alias resolution scheme described in Sec. II-B to identify convergence points. Once peers and convergence points are clearly identified, SAGE builds the final vertices of the neighborhood-based DAG and uses the peer data to create the edges. Special *remote* edges are inserted to account for remote peers, these edges storing the intermediate hops in an effort to provide exhaustive data. For non-remote peers, SAGE identifies the subnet which could correspond to each edge (as mentioned in Sec. II-B). Fig. 5 illustrates the workflow of SAGE, with a focus on the sub-steps involved in the construction of a neighborhood-based DAG. It is worth noting that, after building the DAG, SAGE also performs alias resolution on the interfaces identified for each final neighborhood (re-using a fingerprint-based scheme [10]) to find out whether it consists of one or several routers (also shown in Fig. 5). The processing of this additional router-level data is left for future work.

[2]Up to 5 interfaces per subnet; this can be tuned in SAGE.

	Metrics	Academic	ISP
Subnets	Covered prefixes	95.93%	88.95%
	Exact inferred prefixes	57.55%	46.98%
	Differing by a most one bit	79.94%	74.09%
Graph	Neighborhoods	51	121
	Discovered links	34	113
	Accurate discovered links	34 (**100.0%**)	108 (**95.58%**)
Subnets as links	Matched the groundtruth	34	108
	Matched the exact prefix	33 (**97.06%**)	93 (**82.3%**)
	Mean prefix difference	1	1.2

TABLE I
SAGE VALIDATION (ACADEMIC NETWORK AND ISP BACKBONE).

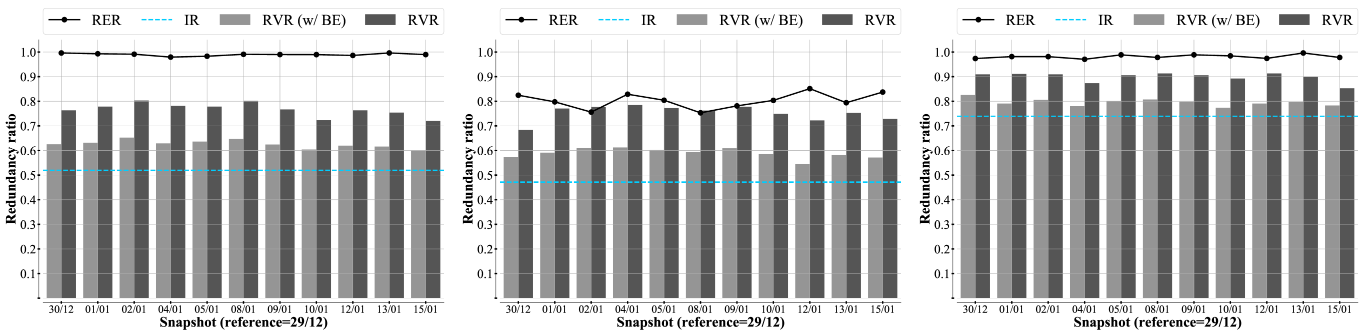
III. EVALUATION

A. Validation

In order to validate SAGE, we measured two groundtruth networks: an academic network roughly the size of a /16 prefix and the backbone of a national ISP whose prefixes cover hundreds of thousands of attributable IP addresses. As SAGE re-uses previous work to discover subnets, neighborhoods, and peers that has been already assessed [19], [20], we are only interested here in verifying whether the graphs discovered by SAGE are true to the measured networks.

One way to assess the soundness of SAGE consists of checking whether the edges found in its graphs match real life links. To do so, we first measured each groundtruth network in late September 2020. Then, for each network, we generated a series of line commands our groundtruth partners had to run on given routers. The routers they had to log onto were those bearing the interfaces that SAGE used to identify neighborhoods while the commands were `show ip ro IP_next` where `IP_next` identifies the next neighborhood from the perspective of SAGE. For instance, if the graph contained the edge `N32 -> N33 via x.y.z.0/30` where `N32` is identified by `a.b.c.1` and `N33` by `x.y.z.2`, then our contact had to log onto the router bearing `a.b.c.1` and run the command `show ip ro x.y.z.2`. If the output of the command returned that the interface was directly connected, this proved the edge discovered by SAGE matched a groundtruth link. We also parsed the output of each command to check whether the connecting subnet matched the one discovered by SAGE.

The results of our validation, provided in Table I, demonstrate almost all links discovered by SAGE were true to the topology for both networks, with 100% of the links discovered on the academic groundtruth being correct and 95% of the links found on the national ISP backbone being true as well. The subnets mapped to links were accurate too: only one subnet had a prefix greater by one bit as for the academic network, while a dozen of subnets mapped to links featured prefixes longer by one bit as for the national ISP backbone (only three prefixes were longer by two bits). A careful analysis of our results for the national ISP backbone revealed that the five incorrect links corresponded to subnets that were mislocated by traceroute probes due to a BGP configuration error. Interestingly, the subnets themselves still matched the subnet-level groundtruth data (though not counted in our metrics). It should be noted the graphs provided slightly more neighborhoods than edges: this is due to having a few neighborhoods without peers (often because they were the



(a) AS6453 (TATA Communications Inc.) (b) AS3257 (GTT Communications Inc.) (c) AS224 (UNINETT)

Fig. 6. Comparison of snapshots collected by SAGE between December 29, 2019 and January 15, 2020 from the PlanetLab testbed (BE = Best Effort).

closest to the vantage point), but also due to having several connected components in the case of the academic network.

It is worth noting that, by design, SAGE will not necessarily discover all links within a network. Discovering all links would notably require probing the same network from different vantage points and combining the resulting graphs. However, we can have an idea of whether a graph is representative of a network by checking whether the subnet-level data covers exhaustively the groundtruth data. This is why Table I also provides subnet metrics to show the subnet data exploited by SAGE is representative, with the ratio of inferred prefixes differing by at most one bit being close to the ratio of covered prefixes. Such a ratio accounts for subnets which the (lack of) *live* interfaces prevents finding the exact prefix. [3]

Finally, it should also be noted that SAGE correctly detected convergence points within the national ISP backbone (not shown in Table I). Interestingly, SAGE missed one such device within the academic network because the corresponding interfaces never appeared in the same list of peers for any neighborhood. However, the two neighborhoods that should have been merged into the missed convergence point had a few links mapped with a subnet from the other half (and vice versa). This suggests both neighborhoods were related and could have been merged by post-processing the graph.

B. Graph Isomorphism

While our validation show promising results, both groundtruth networks mostly consist of end systems and implement little traffic engineering, despite discovering multiple convergence points within the national ISP backbone. Moreover, in the wild, not all subnets and routers of a specific Autonomous System (AS) will be visible to probes, meaning the graphs built by SAGE will not always be as complete as with our two groundtruth networks.

To assess SAGE in the wild, the graphs obtained for the same target network but on different days and from distinct vantage points can be compared. First, let us define a ***snapshot*** as the whole data captured by SAGE for a given target network, usually an entire AS, on a given date from a given (and single) vantage point. If two snapshots of the same AS captured from distinct vantage points on separate dates provide comparable vertices and edges, then the graphs are isomorphic to some extent, showing the graph inference of SAGE is consistent. Of course, two snapshots will likely never provide identical graphs, but a large amount of similarities will imply that the same architecture was captured each time with a few differences due to the change of vantage point.

We quantify similarities between neighborhood-based DAGs as follows. We first select a set of snapshots of a given AS so that each snapshot was captured on a different date and from a different vantage point. We then select the first snapshot as a reference for comparing with subsequent snapshots. Second, for each pair, we quantify how many neighborhoods appear in both snapshots by comparing the interface(s) used to identify each, and divide the total by the number of neighborhoods found in the reference to get a ***redundant vertex ratio (RVR)***. We focus on regular neighborhoods in particular, clearly associated to one or several router interfaces, as best effort neighborhoods (cf. Sec. II-B) are less likely to appear identically in two distinct snapshots. A redundant vertex ratio close to 1 means both snapshots provide (mostly) the same hop-level nodes. To complete this metric, the number of regular neighborhoods found in all snapshots can be divided by the number of regular neighborhoods of the reference snapshot to obtain an ***intersection ratio (IR)***.

We proceed by quantifying how many edges can exist in both snapshots of a pair, i.e., for any edge $u \rightarrow v$ found in the reference snapshot, neighborhoods u and v exist in both snapshots. We then check if $u \rightarrow v$ exists in the second snapshot as well, and divide the number of such redundant edges by the total number of edges that can appear in both snapshots to obtain a ***redundant edge ratio (RER)***. When close to 1, this ratio implies both snapshots provide the same links between the neighborhoods they have in common. If both the redundant vertex ratio and the redundant edge ratio are close to 1, this means both snapshots provide comparable topologies.

Fig. 6 shows our quantification of graph isomorphism for three distinct ASes, using 12 snapshots collected for each from PlanetLab between December 29^{th}, 2019 and January 15, 2020^{th}. Each snapshot was collected from a different vantage point on each date. We selected AS6453 (TATA Communications Inc.) and AS3257 (GTT Communications

[3] Thorough validation of WISE is provided in previous work [19], [20].

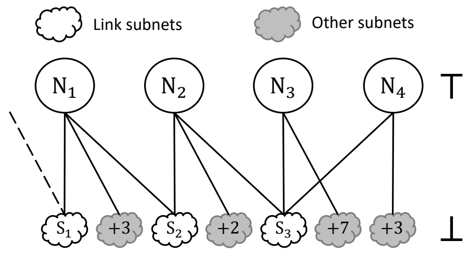


Fig. 7. Fig. 2 re-interpreted as a bipartite graph.

Inc.) because both are Tier-1 ASes from the top 10 of CAIDA's AS ranking [27] that we could measure on a daily basis. For comparison's sake, we provide a third figure for AS224 (UNINETT), a Norwegian Stub AS. Additional figures for other ASes can be browsed at our public GitHub repository. [4]

All figures show that more than 70% of the regular neighborhoods appeared in both snapshots of each pair, while the redundant edge ratios were close to 1, especially with AS6453 (Fig. 6a) and AS224 (Fig. 6c). When best effort neighborhoods were included, the redundant vertex ratios remained above 0.5, while the intersection of regular neighborhoods (i.e., excluding best effort) found in all snapshots was around 0.5 for both AS6453 and AS3257 (both Tier-1 ASes) and even above 0.7 for AS224. These results clearly suggest `SAGE` captured a similar hop-level architecture in each snapshot, though the case of AS3257 (Fig. 6b) seemed more difficult due to this network being a major Tier-1 AS: its redundant edge ratios were noticeably lower than in other cases. However, since they remained around 0.8, it could be possible to combine the links of both snapshots of a pair since they still shared many common neighborhoods. Future work with `SAGE` could consist of merging snapshots of the same network captured from different vantage points to build more exhaustive graphs.

IV. Bipartite Modeling

A. *Interpreting `SAGE` DAGs as Bipartite Graphs*

We introduce a novel bipartite model where one party consists of neighborhoods ($\top$) while the second party accounts for subnets ($\perp$). Indeed, because they model either a single router or a full mesh, neighborhoods are by definition connected together via network links that can be mapped to subnets (as explained in Sec. II-A). Not only this model is intuitive, but it also constitutes a tool to study the hop-level adjacencies of a network as well as the role played by subnets. In particular, the subnet mappings in a DAG can reveal new hop-level adjacencies when a same subnet is mapped to multiple edges. Fig. 7 shows our initial toy network from Fig. 2 re-interpreted as a bipartite graph: because S_3 was mapped to two edges in the initial neighborhood-based DAG, it can be considered as being a hub between N_2, N_3, and N_4, which means N_3 and N_4 are also adjacent to each other.

This bipartite formalism is therefore well suited for studying the subnet degree, i.e., how many devices a subnet connects together (e.g., S_3 in Fig. 7 has a degree of 3), along with its effects on the neighborhood degree, i.e., how many adjacent (one hop away) neighborhoods a given neighborhood features. This is made possible with ***bipartite projection***: given a bipartite graph, a graph with either $\top$ or $\perp$ vertices can be built by creating an edge between a pair of vertices of one party each time they share a common neighbor vertex from the other party. Therefore, projecting a neighborhood – subnet bipartite graph on $\top$ vertices (also known as $\top$***-projection***) produces a hop-level graph.

Interpreting a neighborhood-based DAG as built by `SAGE` as a bipartite graph is almost immediate, although a few special cases must be addressed. An edge between two adjacent neighborhoods which could not be mapped with a subnet is notably accounted for by an *hypothetical* subnet vertex to model the idea that, in theory, a link must exist between both neighborhoods. Likewise, remote edges are replaced by special subnet vertices to show they model more than one hop. Doing so avoids us from adding too many *hypothetical* vertices (neighborhoods or subnets) to model multiple hops adjacencies, which could needlessly complexify the final graph.

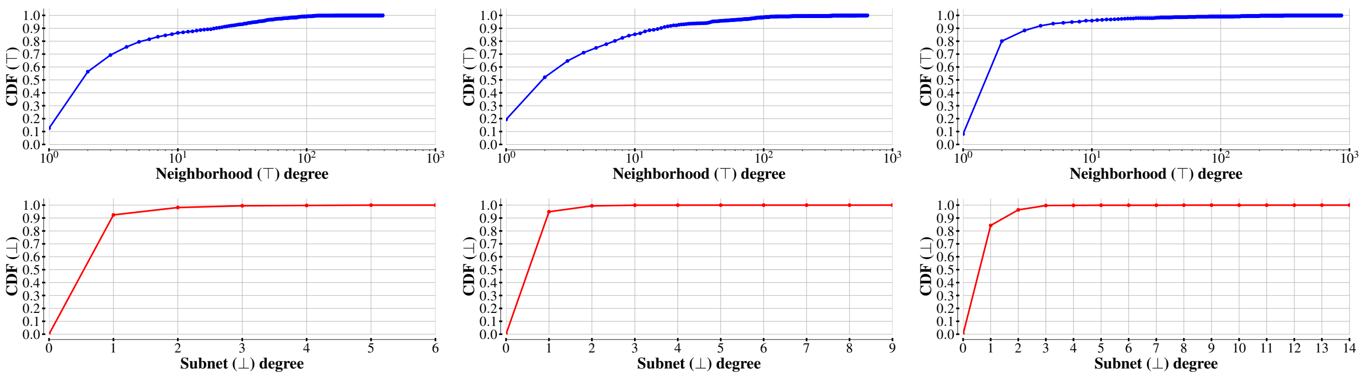
B. *Bipartite Graphs in the Wild*

In 2020, we deployed `SAGE` from both the PlanetLab testbed and the EdgeNet cluster [28] to capture snapshots (as defined in Sec. III-B) of various intra-domain topologies, which we subsequently converted into bipartite graphs to study the topological features of our target networks.

We first take a look at the distributions of both the neighborhood degree and the subnet degree. Fig. 8 shows cumulative density functions (CDFs) of the degree of both neighborhoods ($\top$) and subnets ($\perp$) found in snapshots collected on AS6453 (TATA Communications Inc.), AS6939 (Hurricane Electric LLC), and AS1241 (Forthnet) which are respectively Tier-1, Transit, and Stub ASes. These snapshots are representative, i.e., they are not outliers among snapshots of the selected ASes in the sense that all snapshots provided us with comparable topological features. Among these figures, the CDFs of AS1241 (Fig. 8c) stand out: the $\perp$ CDF shows a bit more than 15% of the discovered subnets (3,052 in this snapshot) had a degree of two or more, with a maximum of 14 (corresponding to an inferred /20 subnet), while the $\top$ CDF shows there were a few very high degree neighborhoods, most neighborhoods (i.e., above 90%) having a degree smaller than 10 (included).

The CDFs for AS6453 (Fig. 8a) and AS6939 (Fig. 8b) rather show their neighborhoods had higher degrees overall, but smaller maxima. Likewise, their subnets had smaller maxima too, and the shares of subnets having a degree of two or more were noticeably smaller, though they hid dozens of such subnets due to the large number of discovered subnets (8,188 for AS6453 and 10,645 for AS6939). These differences in the distributions hint at different types of topologies: AS1241 appeared to be very centralized, while both other ASes seemed to be much more distributed. Such observations can be explained by the roles played by these ASes in the Internet: AS1241 is indeed operated by a Greek ISP, while AS6453 and AS6939 are respectively Tier-1 and Transit ASes found in the top 10 of

[4]https://github.com/JefGrailet/SAGE/tree/master/Python/Isomorphism



(a) AS6453 (09/20/2020) (b) AS6939 (09/20/2020) (c) AS1241 (09/12/2020)

Fig. 8. CDFs of the degrees of neighborhoods (⊤) and subnets (⊥) using three snapshots collected from the EdgeNet cluster.

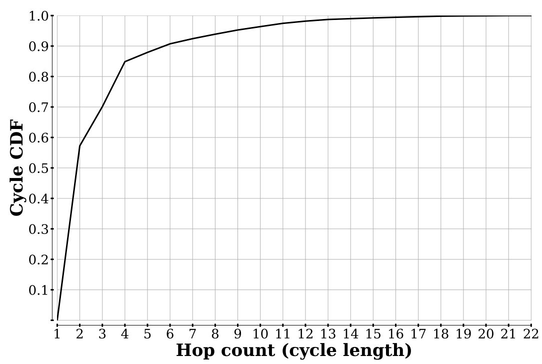


Fig. 9. CDF of the length of cycles in bipartite graphs (all 2020 snapshots).

CAIDA's AS ranking [27]. Interestingly, AS6453 featured the smallest maxima in Fig. 8, suggesting its observed topology was the most distributed among all three snapshots. Overall, these distributions suggest that the properties of our bipartite graphs are consistent with the role assumed by each target network in the Internet. Additional figures for various ASes can be browsed at our public GitHub repository. [5]

Another interesting application of our bipartite formalism lies in the study of ***topological cycles***, i.e., cycles observed within the (inferred) graph of a network. Such cycles do not appear at probing time due to the nature of forwarding: at best, `SAGE` can discover *diamonds*, i.e., subgraphs delimited by a divergence point (e.g., where load balanced paths start) followed by a convergence point [18]. As such, topological cycles are not strictly equivalent to routing loops experienced with `traceroute`, notably because the latter can be explained by various network mechanisms such as MPLS tunnels [29].

Fig. 9 shows a cumulative density function of the length of topological cycles found in bipartite graphs built from all snapshots collected during 2020, from both the PlanetLab testbed and the EdgeNet cluster. This figure only takes account of ***base cycles***, i.e., cycles that cannot be decomposed in smaller ones, and omits cycles that encompassed special vertices modeling remote edges (cf. Sec. IV-A). It also omits cycles encompassing subnets whose the prefixes were too small for their degree (e.g., a /29 subnet cannot have a degree of 10), as such a high degree may be the result of undetected convergence points (as defined in Sec. II-B), though the subnet may also be undergrown. Despite these precautions, the figure still accounts for 25,095 cycles found within 345 `SAGE` snapshots after bipartite conversion. In particular, a majority of the cycles were two hops long and may account for back-up links, since they involved two distinct subnets connecting the same neighborhoods. Fig. 9 also highlights that a large share of the cycles ranged from three to a dozen of hops, which may correspond to structures such as load balancers. We leave a detailed characterization of these cycles for future work.

[5] https://github.com/JefGrailet/SAGE/tree/master/Python/INSIGHT/Results

C. *Projections in the Wild*

Finally, let us take a look at the neighborhood degree after projecting bipartite graphs on neighborhoods (⊤). Though neighborhood-based DAGs built by `SAGE` already reveal adjacencies, the mappings between subnets and edges can reveal new adjacencies, and as a result, the ⊤-projections of our bipartite graphs can better account for the hop-level.

Fig. 10 shows the distribution of the neighborhood degree both in bipartite graphs (dashed line) and in their ⊤-projections (plain line) as a complementary cumulative density function (CCDF), using again the bipartite graphs built from all 2020 snapshots (i.e., the same data as for Fig. 9). This figure shows the neighborhood degree distribution is shaped like a power law for 99.9% of the neighborhoods in bipartite graphs and for 99% of neighborhoods in the ⊤-projections. The research community has debated many times on whether the router degree in the Internet, i.e., the number of adjacent routers of a given router, was distributed in this manner, among others by discussing the impact of Layer-2 equipment [30]. Hopefully, our observations corroborate these claims rather than they invalidate them: indeed, a neighborhood can consist of either a single router or a full mesh, possibly involving Layer-2 equipment (cf. Sec. II-A). Future work includes discovering routers within neighborhoods with alias resolution and inferring the presence of Layer-2 equipement.

Interestingly, the neighborhood degree in bipartite graphs covers a larger range of values. This should not be too surprising, as we expect many subnets to consist of end systems or to act as links towards unprobed parts of the Internet. It should be

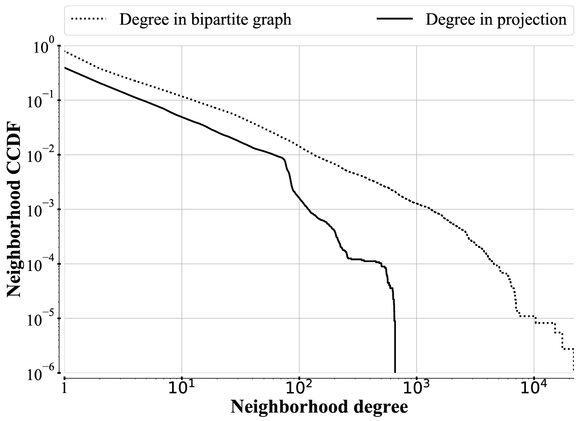


Fig. 10. CCDF of the degree of neighborhoods (all 2020 snapshots).

noted, however, that the extrema shown for bipartite graphs are measurement artefacts. In particular, degree-1 neighborhoods (in bipartite graphs) were typically bordered by a single subnet and could not be directly peered to any other neighborhood by SAGE. The highest degree neighborhoods also resulted from measurement issues such as unusually undergrown subnets.

V. Conclusion

In this paper, we introduced SAGE, a new topology discovery tool able to infer the hop-level adjacencies of a target network from a single vantage point thanks to subnet-level data. SAGE builds a directed acyclic graph where vertices model single hops while edges model the links, which are also mapped to the discovered subnets. Using two groundtruth networks and data collected in the wild, we showed that SAGE can discover intra-domain topologies that are both true to the targets and consistent upon changing the vantage point.

We also introduced a new bipartite formalism to study network topologies. Using data collected from both the PlanetLab testbed and the EdgeNet cluster, we discussed various applications of our bipartite formalism and showed that the topological properties they highlight are consistent with the measured networks or previous studies. In particular, we demonstrated that the degree of vertices in hop-level graphs, obtained by projecting our bipartite graphs on the hop-level, follows a power law shape, therefore corroborating previous work on router- and hop-level. We also showed that our bipartite graphs can be a tool for discovering back-up links and other network structures by analyzing their cycles.

References

[1] B. Donnet and T. Friedman, "Internet Topology Discovery: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 9, no. 4, December 2007.

[2] H. Haddadi, G. Iannaccone, A. Moore, R. Mortier, and M. Rio, "Network Topologies: Inference, Modeling and Generation," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 2, pp. 48–69, April 2008.

[3] 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 Internet Measurement Conference (IMC)*, October 2006.

[4] R. Beverly, "Yarrp'ing the Internet: Randomized high-speed active topology discovery," in *Proc. ACM Internet Measurement Conference (IMC)*, November 2016.

[5] K. Vermeulen, J. P. Rohrer, R. Beverly, O. Fourmaux, and T. Friedman, "Diamond-Miner: Comprehensive Discovery of the Internet's Topology Diamonds," in *Proc. USENIX Symposium on Networked Systems Design and Implementations (NSDI)*, February 2020.

[6] Y. Huang, M. Rabinovich, and R. Al-Dalky, "FlashRoute: Efficient Traceroute on a Massive Scale," in *Proc. ACM Internet Measurement Conference (IMC)*, October 2020.

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

[8] M. H. Gunes and K. Sarac, "Importance of IP Alias Resolution in Sampling Internet Topologies," in *Proc. IEEE Global Internet Symposium*, May 2007.

[9] K. Keys, Y. Hyun, M. Luckie, and k. claffy, "Internet-scale IPv4 Alias Resolution with MIDAR," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 383–399, April 2013.

[10] J.-F. Grailet and B. Donnet, "Towards A Renewed Alias Resolution with Space Search Reduction and IP Fingerprinting," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, June 2017.

[11] B. Augustin, B. Krishnamurthy, and W. Willinger, "IXPs: Mapped?" in *Proc. ACM Internet Measurement Conference (IMC)*, November 2009.

[12] G. Nomikos and X. Dimitropoulos, "traIXroute: Detecting IXPs in traceroute paths," in *Proc. Passive and Active Measurements Conference (PAM)*, April 2016.

[13] D. Feldman, Y. Shavitt, and N. Zilberman, "A Structural Approach for PoP Geo-Location," *Computer Networks (COMNET)*, vol. 56, no. 3, pp. 1029–1040, February 2012.

[14] Y. Shavitt and N. Zilberman, "Geographical Internet PoP Level Maps," in *Proc. Traffic Monitoring and Analysis Workshop (TMA)*, March 2012.

[15] M. E. Tozal and K. Sarac, "TraceNET: an Internet Topology Data Collector," in *Proc. ACM Internet Measurement Conference (IMC)*, November 2010.

[16] ——, "Subnet Level Network Topology Mapping," in *Proc. IEEE International Performance Computing and Communications Conference (IPCCC)*, November 2011.

[17] J.-F. Grailet, F. Tarissan, and B. Donnet, "TreeNET: Discovering and Connecting Subnets," in *Proc. Traffic and Monitoring Analysis Workshop (TMA)*, April 2016.

[18] B. Augustin, R. Teixeira, and T. Friedman, "Measuring Load-Balanced Paths in the Internet," in *Proc. ACM Internet Measurement Conference (IMC)*, October 2007.

[19] J.-F. Grailet and B. Donnet, "Revisiting Subnet Inference WISE-ly," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, June 2019.

[20] ——, "Virtual Insanity: Linear Subnet Discovery," *IEEE Transactions on Network and Service Management (TNSM)*, vol. 17, no. 2, pp. 1268–1281, June 2020.

[21] D. J. Watts and S. H. S.H. Strogatz, "Collective Dynamics of Small-World Networks," *Nature*, vol. 393, pp. 440–442, June 1998.

[22] M. Newman, "Scientific Collaboration Networks. Network Construction and Fundamental Results," *Physical Review E*, vol. 64, no. 1, June 2001.

[23] A. Iamnitchi, R. Matei, and I. Foster, "Small World File-Sharing Communities," in *Proc. IEEE INFOCOM*, April 2004.

[24] F. Tarissan, M. Latapy, P. Mérindol, J.-J. Pansiot, B. Quoitin, and B. Donnet, "Towards Internet Topology Modeling Through Bipartite Graphs," *Computer Networks (COMNET)*, vol. 57, no. 11, pp. 2331–2347, August 2013.

[25] R. Pastor-Satorras and A. Vespignani, *Evolution and structure of the Internet: A statistical physics approach*. Cambridge University Press, 2007.

[26] H. Kardes, M. Gunes, and T. Oz, "Cheleby: a Subnet-Level Internet Topology Mapping System," in *Proc. International Communications Systems and Networks and Workshops (COMSNETS)*, January 2012.

[27] CAIDA, "CAIDA's Ranking of Autonomous Systems (ASRank)," https://asrank.caida.org/.

[28] B. C. Senel, M. Mouchet, J. Cappos, O. Fourmaux, T. Friedman, and R. McGeer, "EdgeNet: A Multi-Tenant and Multi-Provider Edge Cloud," in *Proc. International Workshop on Edge Systems, Analytics and Networking (EdgeSys)*, April 2021.

[29] P. Marchetta, A. Montieri, V. Persico, A. Pescapé, I. Cunha, and E. Katz-Bassett, "How and how much traceroute confuses our understanding of network paths," in *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN) Conference*, June 2016.

[30] P. Mérindol, B. Donnet, O. Bonaventure, and J.-J. Pansiot, "On the Impact of Layer-2 on Node Degree Distribution," in *Proc. ACM Internet Measurement Conference (IMC)*, November 2010.