Discovering and Mapping Subnet Level Topology

Wei Yao, Hai Zhao, Jing-Jing Chen*

School of Computer science and Engineering, Northeastern University, China
yaow.neu@gmail.com, zhaoh@mail.neu.edu.cn, cjjzm0903@163.com

Abstract

The research of Internet measurement has promoted the development of router-level topology discovery, while the subnet of the network layer can provide a more detailed intermediate complementary view. To improve the current level of router-level topology discovery, some researchers have proposed subnet discovery methods. However, the existing methods do not perform well in terms of efficiency and accuracy, which leads to the failure of the final resulting topology map. In this paper, we propose a new approach to discover subnets, which consists of a network pre-processing stage and a subnet inference stage. Given a set of target IP prefixes, the network pre-processing stage introduces a minimal probing overhead to improve probing efficiency. Based on the IP address allocation principle, the subnet inference stage utilizes a set of complementary inference rules to infer subnets. The experimental results show that our method can achieve a higher accuracy in discovering subnets while keeping lower probing overhead compared with existing methods. Finally, we utilize the proposed approach to discover the subnets of six geographically dispersed public Autonomous System (AS) networks and analyze their various subnet characteristics, including degree distribution.

Keywords: Internet measurement, Subnet discovery, Subnet inference, Network topology

1 Introduction

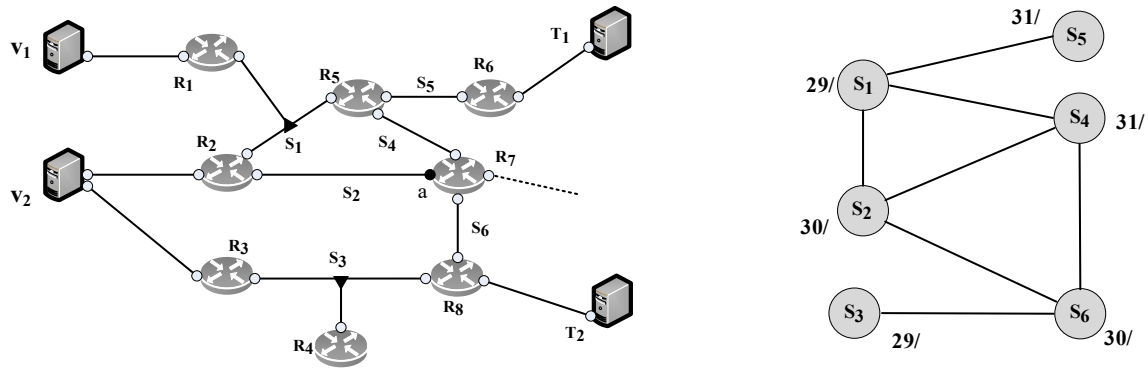
As a large and spontaneously growing complex network structure, Internet has been studied and analyzed systematically from various perspectives [1]. During the past fifteen years, many successful projects and research efforts have been focused on topology discovery and analysis of the Internet [2]. These efforts aim to reveal different levels of the Internet topology, including autonomous system (AS) level, router level, or IP interface level maps [3-5]. Understanding these Internet topologies and structural properties would enable us to design better network protocols and Internet-like synthetic models [6]; optimize the allocation of network structure and resources [7]; and improve the performance and fault tolerance of Internet services to a certain extent [8]. Existing network topology mapping works mainly focus on router level maps. These maps might be then studied to learn about coefficient and degree distribution various characteristics of routers, including clustering coefficient and degree distribution.

Recent studies have suggested extending the traditional network topology to the point of presence level (PoP) [9], Internet eXchange Points (IXPs) [10], or subnets [11-12], which helps to obtain a representative and accurate Internet topology. This paper is in the scope of subnet level. Similar to routers, subnets are composite structures. A subnet denotes a set of devices that are located on the same connection medium and that can directly communicate with each other at the link layer (layer 2) [13]. All IPs in a subnet are addressed with a common most-significant bit-group (IP prefix). Subnet level maps describe subnets as vertices, and routers are described as links to these subnets. As shown in Figure 1, v_1, v_2, and T_1 are hosts with unweighed links, and S is a subnet associated with router R between path traces. It is a way to enrich the router layer mapping by using subnet level connection information. In other words, if subnets are regarded as a simple link in the process of network topology mapping, the fidelity of the resulting mapping would be reduced.

The measurement of Internet topologies has become a challenging task due to the lack of public and systematic information about subnets and subnet interconnections [14]. To facilitate subnet level topology studies, several subnet discovery approaches have been developed to collect the required information on subnets [12, 15-20]. Most of these approaches utilize well-known debugging tools such as *ping* to directly check whether a destination IP address is in use or not, and *traceroute* to obtain subnets information on a given path between a source and a destination [21].

However, despite a decade of advancement, subnet discovery has made a significant progress, there still have some drawbacks. For those proposed methods, it is difficult to guarantee the soundness and completeness of inferred subnets (especially for the case when only a few responsive IPs are in the subnet). More importantly, these approaches are challenging to evaluate subnets comprehensively. For example, IGMP probing [18] has been utilized to discover subnets by employing rules (e.g., routers must be connected through the same Layer-2 device) [19]. However, due to increasingly filtering done by network operations, it becomes unusable in practice, making this inferencing approach is obsolete [22]. XNet has a higher probing overhead and also tends to fragment large subnets into several smaller and incomplete ones [16]. [12] introduces a refinement stage and represented the topology of the subnetworks as a tree-like structure, which can clearly display the relative positions between networks.

*Corresponding Author: Wei Yao; E-mail: yaow.neu@gmail.com
DOI: 10.53106/160792642023032402008



(a) Internet topology example (b) Subnet level topology

Figure 1. A network layer topology map (left) is represented as a subnet graph (right)

Nevertheless, this approach does not take into account other subnet boundry situations. Hence, these inaccuracies would significantly affect the representativeness of the resulting subnet topology. Even though researchers have investigated several solutions, discovering the subnet topology of the Internet is still a challenging task because (i) In terms of accuracy, most methods use a single inference rule to discover subnets and cannot consider comprehensive subnet boundary situations, resulting in a lower accuracy; (ii) In terms of topology completeness, there are often only a small fraction of responding IP addresses in real-world situations, which can affect the result of subnet inference; (iii) In terms of efficiency, it is difficult to achieve efficient probing due to high probing overhead. Therefore, it is necessary and imperative to develop an efficient and effective method to solve the above problems.

In this paper, we propose NSTology, a new subnet discovery approach to collect and manipulate subnet topology information. Given a list of targeted IPv4 prefixes, NSTology attempts to discover the subnets that accommodate all alive IP addresses. To improve probing efficiency, NSTology introduces a network pre-processing stage to check which IP addresses are alive and reachable. By listing responsive IP addresses, this stage significantly reduces probing overhead for subsequent process steps. To infer the subnet between IP addresses, NSTology utilizes a subnet inference stage to check if each observed IP address meets inference rules. Specifically, we analyze IP addresses within the collected path traces and present a set of complementary inference rules to infer subnets accurately. Note that similar to other probed-based approaches, if an inferred subnet interfaces remain silent to probe packets, or the border routers discard the corresponding response packets, NSTology cannot obtain the subnet. In order to evaluate the performance of NSTology comprehensively, we conducted an experimental study on a public Internet network. The experimental results show that compared with the existing approaches, NSTology can achieve higher accuracy to discover subnets while using less probing overhead. In addition, we utilize NSTology to discover the subnet information of six AS networks existing in different regions of the world, and study the subnet level maps to identify common and discrete subnet characteristics among these ASes. The main contributions of this paper are listed as follows:

(1) We propose a novel lightweight subnet discovery method, called NSTology, that can accurately collect and manipulate subnet topology information. This method solves the limitations of current methods.

(2) We introduce a pre-processing startegy and a set of relaxed inference rules to quickly find more reliable and reasonable subnets.

(3) We evaluate the effectiveness of proposed method by conducting a comprehensive experiment in a real-world ground truth. We also use NSTology to discover the subnet information of six dispersed AS networks.

The remainder of the paper is organized as follows. Section 2 introduces related work. Then, we present complementary inference conditions to infer subnets in Section 3. Section 4 presents the detail process of NSTology. In Section 5, we evaluate the performance of NSTology and compare it to the state-of-the-art techniques. We collect subnet level maps of six common AS networks and study their topology characteristics in Section 6. Finally, Section 7 concludes this paper.

2 Related Work

2.1 Topology Discovery for Different Levels

To facilitate topology measurement studies, many successful topology discovery technologies and projects have been proposed for collecting a complete and accurate Internet topology maps at IP interface, router, subnet, and AS levels.

IP interface level. Various methods have been proposed to discover IP level maps. [5] introduces a set of methods for discovering the IP topology, which uses SNMP, broadcast ping, DNS information, and traceroute to collect alive IP addresses. Nevertheless, the inherent limitations of these methods, they cannot correctly infer the connection relationship when IP interfaces that do not have characteristic traffic. From a different viewpoint, a recent work in [25] develops a new topology discovery method using equipment-alarm information to find more IP addresses.

Router level. To discover accurate router level maps, researchers have developed a number of topology inference techniques. However, as routers may contain multiple IP interfaces, how to identify IP interfaces within the same router is a crucial problem (i.e., also called alias resolution).

Previous efforts in probing-based [26] and inference-based [27] are available for alias resolution. However, neither of them can comprehensively infer router behaviors. Recent works [12, 28-29] focus on the accuracy and efficiency problems for collecting router-level maps, which enables a quick and efficient router-level measurement.

AS level. Compared to router level maps, producing AS level topology maps is relatively simple. AS-level topology maps utilize various sources of information, including BGP routing tables, traceroutes, and Internet registry databases, to create more accurate higher-level Internet maps [30]. Another latest IP-to-AS mapping techniques have been discussed in [3].

Subnet level. Different from the above studies, subnet level mapping has emerged as a new intermediate way to improve the understanding of the Internet topology. This is also the focus of our work. To better map the subnet view of Internet topology, a number of techniques have been proposed for subnet discovery, which can be classified into two categories. The passive-based approaches require to send multiple probes in the network but with additional post-process (i.e., without probing) to discover subnets. For example, lightweight IGMP probing [18] exploits the characteristic of silently collecting all multicast interfaces of a router in a single probe [19]. However, it becomes outdated as filtering is now widely applied by operators [22].

On the other hand, most studies use active-based tools (i.e., done as traceroute meanwhile or shortly after, with additional probes). In particular, [23] utilizes broadcast addresses to determine whether an address is the boundary of a subnet. With IP address assignment practices, the work [14] develops a set of conditions to determine the subnet boundary. Then, Cheleby [24] enhances the method in [14] by exploiting only the distance preservation condition, which reduces the computational complexity. Another technique called TraceNet [15] employs some tight judgment conditions to infers subnets on a given path between the source and the target. XNet [16] improves the TraceNet tool to discover individual subnets. However, these approaches still cannot comprehensively and accurately identify subnets.

Recent works on subnet discovery either try to reduce the complexity of the probing process [12, 17], or try to increase accuracy with additional probing and inference steps [20, 31]. Particularly, the authors of [12] reduce the complexity by introducing a refinement stage and a tree-like structure able to show the relative positions between the subnets. [17] enhances the probing overhead by introducing a network pre-scanning step. While these approaches have lower runtime complexity than previous solutions [14-16], it still produces relatively low accuracy. The work in [20] develops a target scanning stage to find a minimal TTL (i.e., Time-To-Live) value of the target distance, which does not perform a complete traceroute towards each target IP address. Then, it employs five inference rules by considering different scenarios to infer alive subnets. Apart from the basic inference steps in [20], [31] also employs a post-processing step to check if an IP address belongs to the current subnet. This approach studies the potential for inferring a comprehensive map of the target domain. Although this results in a more accurate inferred topology, it requires additional probing and inference costs. Another of studies [32] aims to build a directed acyclic graph of a subnet by modeling its (meshes of) routers, which can be utilized to discover back-up links and other network structures. In addition, [33] presents a subnetwork generator (SubNetG) that reflects the subnet-level characteristics of the Internet backbone, which captures the subnetwork distributions.

2.2 Literature Comparison with Subnet Discovery

A number of studies about subnet discovery have been developed and most of them use Traceroute and Ping tool to discover subnets. However, neither of the studies [11, 13-16] comprehensively consider the actual demand for subnet discovery in real environments. Real-world complications (e.g., only a small fraction of routers and lack of proper vantage points) will necessitate more substantial evaluation efforts to provide the sort of completeness, accuracy, and efficiency properties that are required before the tools' results can be trusted and used in practice. For example, Cheleby [23] only adopts the distance condition in [11] to infer subnets, which requires traceroute paths collected from a large number of vantage points. This probably incurs a low accuracy and additional probing overhead. While XNet offers an attractive method for inferring subnets, it produces high probing complexity and cannot guarantee the completeness of the inferred subnet. Moreover, this approach tends to divide the larger subnet into several smaller and incomplete subnets. Although TreeNet [17] built upon XNet [16] improves its subnet inference and execution time by adding a refinement stage and a set of heuristics, it still does not consider some boundary conditions that satisfy the subnet. Similarly, the work in [22] can not significantly improve the accuracy of subnetting. The latest works in [20, 31] provide great enhancements for subnet inference in terms of accuracy, completeness, and efficiency. However, these methods require more computation complexity than ours and its implementation in real environments is relatively complex.

An effective subnet discovery should achieve a high accuracy and a high completeness. Additionally, to meet the high probing efficiency of real world, an efficient method should have low computational complexity and high implementation speed. Thus, four important properties, are implemented in our proposed NSTology to improve the efficiency and accuracy of subnet discovery.

To solve the above issues, we propose a new lightweight subnet discovery approach, which introduces a fast pre-process stage and several relaxed inference rules to find reliable and accurate subnets. Table 1 clearly compares the existing literature to the proposed NSTology based on the properties that an effective and efficient subnet discovery method should have.

Table 1. Comparison of different subnet discovery methods

Method	Year	Probe way	Range of applications	Implementation difficulty	Completeness	Efficiency	Topology accuracy
[14]	2007	Active	Big	Easy	Low	Low	Low
[15]	2010	Active	Big	Easy	Low	Middle	Low
[18]	2011	Passive	Small	Easy	Low	High	Low
[23]	2011	Active	Small	Easy	Low	Middle	Low
[16]	2011	Active	Big	Easy	Low	Low	Low
[24]	2012	Active	Big	Easy	Low	Middle	Low
[12]	2017	Active	Big	Easy	Middle	Middle	Middle
[20]	2019	Active	Big	Middle	High	High	High
[31]	2020	Active	Big	Hard	High	High	High
[32]	2021	Active	Big	Hard	High	High	High
Our	2022	Active	Big	Easy	High	High	High

3 Subnet Discovery Analysis

In this section, we give some definitions used (Section 3.1) and present inference rules to discover subnets (Section 3.2). Subnet discovery is the process of listing all surviving IP addresses of a subnet and annotating the subnet with the observed IP address space (i.e., subnet mask). A subnet holds an IP address range for assignment to the connected router interfaces. Nevertheless, from the view of the network layer, the subnet is independent of any configuration below layer-3. It can be point-to-point links and multi-access links including Ethernet, virtual MPLS tunnels, FDDI, etc.

3.1 Foundations

Definition (Sunbet): From the perspective of a single vantage poing v, a subnet S can be identified as a set of responsive interfaces: $S = \{l_1, l_2, ..., l_n\}$, where n is the number of interfaces of l. The interface l has a related IP address shown as l_{ip} and a hop distance from a vantage v, defined as, l_v^h. When the vantage point is obvious in the context, we use l^h to substitute l_v^h. The subnet S_x^p denotes its subnet address is x and the length of subnet mask is p. The degree of subnet S^d represents the number of interfaces that that S can accommodate. As shown in Figure 2, the degree of subnet S is 3.

Definition (Contra-Pivot Interface) [16]: Assuming that there always exists a stable routing path in the network, the ingress router of subnet S, with respect to a vantage point v, is the last router that packets destined to S are delivered through. The contra-pivot interface of subnet S is the interface located on the ingress router of the subnet. As shown in Figure 2, R_1 is the ingress router of S and data packets that are sent from v to R_1.b enter S through R_1. Interfaces {R_3.c, R_5.d} are the pivot interfaces and R_1.b is the contra-pivot interface.

Definition (Trace): A trace, (s, T: s, ..., x, y, T) or Trace (s, T), defines a path from source s to destination T, which contains a series of visited IP interfaces along the path. x and y represent the IP interfaces of last two routers to the target, respectively. In practice, due to network strategies, Trace (s, T) may not be equal to Trace (T, s). If not specified, we drop s and use Trace (T).

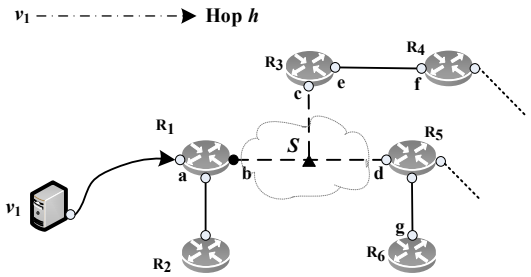


Figure 2. A subnet S of size three located at hop distance h from a vantage point v_1

3.2 Subnet Inference Rules

In order to discover subnet comprehensively and accurately, we need to analyze topological observations within the same subnet. The existing approaches assume a fixed subnet boundary, which may be too fine (wasteful probes) or too rough (lack of information). Based on the definitions mentioned above, we present a set of complementary rules to help us perform the necessary verification during subnet inference process as follows.

Inference Rule 1: A subnet contains at least one contra-pivot interface that belongs to the ingress router, and its assigned IP addresses share a common subnet mask (prefix).

In fact, a subnet can have several contra-pivot interfaces. This may be due to network problems (i.e., redirection) or network policies. For example, the router has a backup interface for the subnet in case the first one fails. In such a case, the subnet contains at least one contra-pivot interface. In other words, given a stable ingress router to a subnet of degree n, there are at most n-1 pivot interfaces and at least one contra-pivot interfaces. Note that the lack of a contra-pivot interface is most likely due to network issues. In addition, according to IP address allocation guideline (RFC 2050), the IP addresses of all interfaces within a same subnet have the same maximum x prefix, i.e., the subnet address.

It should be emphasized that when a large subnet lacks of responsive interfaces, most existing approaches tend to divide it into several smaller subnets containing a few responsive interfaces, where only one of them involves a valid contra-pivot interface. This indicates that only this subnet can be considered as sound. To solve this problem, we make verifications to ensure that any inferred subnet at least has

a valid contra-pivot interface. Formally, we utilize $<t^{ip}|\ ttl> \leftrightarrow <Response\_Message>$ to indicate that probing of an IP address t^{ip} with TTL value of *ttl* results in a response message of type *Response_Message*. If the probing does not reply any response message, the *Rseponse_Message* is set to *nil*. In particular, we send an ICMP probe to a given target IP address *t* and obtain its TTL = t^h value from the ICMP response message. Then, we determine a contra-pivot interface by building a small /*m* subnet (/31 or /30) that containes the target *t*. For each interface *l* other than target *t* in the subnet, two probes are done. We divide the probing results into three cases. (1) If *t* is a contra-pivot interface and *l* is a pivot interface, an ICMP probe to *l* with TTL = t^h should return an ICMP TTL Exceed message, i.e., $<l^{ip}|\ t^h> \leftrightarrow <ICMP\_TTL\_Exceeded>$ and another probe with TTL = $t^h + 1$ also returns an ICMP Echo Reply, i.e., $<l^{ip}|\ t^h+1> \leftrightarrow <ICMP\_Echo\_Reply>$. (2) If the first probe with TTL = t^h receive ICMP Echo Reply message, i.e., $<l^{ip}|\ t^h> \leftrightarrow <ICMP\_Echo\_Reply>$, and the second one with TTL = $t^h - 1$ also results in ICMP Echo Reply message, i.e., $<l^{ip}|\ t^h-1> \leftrightarrow <ICMP\_Echo\_Reply>$, then *l* is defenitely a contra-pivot interface and *t* is a pivot interface. (3) If *l* and *t* both are contra-pivot interfaces, the first probe with TTL = t^h should return ICMP Echo Reply message, i.e., $<l^{ip}|\ t^h> \leftrightarrow <ICMP\_Echo\_Reply>$, and the second one with TTL = $t^h - 1$ shoud results in ICMP TTL Exceed message message, i.e., $<l^{ip}|\ t^h> \leftrightarrow <ICMP\_TTL\_Exceed>$. The principle behind these situations is that two consecutively adjacent IP addresses are most likely to be hop apart. In this way, we ensure that each inferred subnet contains of at least a valid contra-pivot interface.

Inference Rule 2: The IP addresses within the same subnet should have similar path traces to a vantage point and only the last hop or two hops are different. The path hop distances of these addresses differ at most by one unit. All interfaces in the same subnet have the same subnet prefix, thus their IP addresses have similar path traces information. For instance, let us assume that for the network topology example in Figure 3, R_1 and R_5 are routers on the FDDI ring. The hops before interface c are the same in the trace collection from Trace (a, c, e) to Trace (a, c, g). For the path Trace (a, d), their last two hops are different. Moreover, the TTL distances between the interfaces {d, e, f, g} differ at most one hop. This rule helps to eliminate the inaccurate candidate subnets.

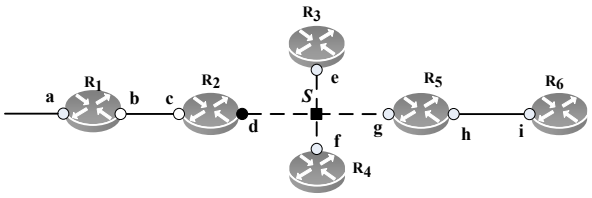


Figure 3. A sample of network topology with a subent *S*

For clarity, we give a notion of key interface. The key interfaces denote the last two interfaces in the route before a given target address. For example, in Figure 3, interfaces {g, h} are the key interfaces in router R_6. In fact, the key interfaces towards a given target address are not always visible (i.e., "*" anonymous IPs). This situation may be the result of network filtering, such as rate-limiting. Therefore, as long as the IP interface is at the same TTL distance from other interface in the possible subnet, we still consider them to be part of the same subnet.

Inference Rule 3: If there exits a loop-free trace, IP addresses in a subnet cannot appear in any path trace unless they have a neighborhood relationship with each other. If the IP addresses from the subnet appear in the same track, they should appear next to each other.

This situation arises from the fact that nodes in the same subnet are directly connected, and there should be a hop between each other in the path tracking. For example, without knowledge of the network topology in Figure 3, the path trace (c, j, i) from source R_1 to target R_6 indicates that interface R_2.c and R_6.i cannot be in the same subnet as they are two hops away from each other. According to RFC 1812, routers usually send response messages to the traceroute source along with the IP address of the outgoing interface. In this case, all IP addresses in the trace will come from different subnets. However, in practice, IP addresses of other interfaces (i.e., incoming interface, default interface, or the interface on the shortest path from itself) may be returned, and these different practices may result in path tracking of two IP addresses from the same subnet. In this case, such IP addresses are at most one hop away from each other.

In summary, the above inference rules introduce the relationships between the IP addresses allocated by the interfaces in the same subnet. Therefore, we utilize the inference rules to verify the authenticity of the inferred subnet. This will not only help to find more reasonable subnets but also relax the definition of subnets to a certain extent, allowing more peculiar network configurations to be discovered.

4 NSTology

In this section, we present a new approach called **N**etwok **S**ubnet **T**op**ology** (NSTology), which aims to discover observable subnets in the target network. Given a list of target IPv4 prefixes belonging to the target network, NSTology consists of two stages: the network pre-processing (Section 4.1) and the subnet inference (Section 4.2). Figure 4 summarizes the typical deployment of NSTology (left) and the design flow chart of NSTology (right). Particularly, NSTology can send probing towards a target network running from a single vantage point.

4.1 Network Pre-processing

In the initial stage, NSTology lists all IP addresses of the target domain, using single address or IPv4 prefixes as input. The purpose of this step is, by using IP address assignment principle, to check which IP addresses in the target list are alive and reachable. NSTology works by sending a single probe (usually an ICMP probe, but UDP and TCP can also be considered) to each possible IP address contained in the initial target prefixes and waiting for a response. Instead of simply decompose a prefix into its constiuent subnets, NSTology attempts to quickly discover alive IP addresses of the internal subnets of a given prefix. Intuitively, the two numerically consecutive IP addresses are more likely to share a path. However, probing the consecutive addresses in the prefix

would increase network delays. Hence, to improve probing efficiency, NSTology performs a binary search way on the address space represented by the input prefix. The motivation is to adjust the number of probes according to the degree of subnetting in the prefix to avoid wasted probing. Given a subnet S_x^p, NSTology divides the subnet prefix into two parts, and then continuously detects the central address of each part from the vantage point. More formally, the two addresses are:

$$c_1 = x + 2^{32-p-2} + 1 . \quad \textbf{(1)}$$

$$c_2 = x + 3 \cdot 2^{32-p-2} + 1 . \quad \textbf{(2)}$$

For each input prefix, NStology maintains a set of discovered interfaces in the target network. Let I denotes the list of all unique IP addresses belonging to the target network domain. Let Q_i represents the list of IP addresses in the target network domain discovered by the ith probing. Then, NStology splits the input prefix in half and performs recursively on those two smaller prefixes if the following conditions are met:

$$|Q_i \setminus I| \geq \varphi . \quad \textbf{(3)}$$

where we set φ =1 such that the probing stops only if no new IP addresses is found. Then, we update the target IP address list. Meanwhile, the overall process is sped up with multiple threads. Since some IP addresses may not reply in the first measurement stage (i.e., network issues), NSTology also conducts a second pre-processing stage. At this stage, the initial timeout delay value being doubled is utilized to obatin as many responsive addresses as possible. This guarantees that unresponsive addresses are indeed dead and not be inaccessible due to certain network conditions. Note that in general, two rounds of pre-processing are enough. As unresponsive IP interfaces do not appear in the IP addresses set, they will not be probed again in the next stage, i.e., subnets inference. Thus, by listing the only responsive IP and a preliminary concurrent probing operation, NStology saves time for the subsequent steps. At the end of the network preprocessing, NSTology sorts the probed IP addresses with regard to the IP range (i.e., according to their values as 32-bit integers).

4.2 Subnet Inference

Once all alive addresses have been found, subnet inference phase consists in discovering all subnets accommodating consecutive addresses by relying on the inference rules described in Section 3.2. NSTology uses an iterative way to form all candidate subnets, which starts from /31 subnets to /x (x > 31) subnets containing the target IP address. More specifically, NSTology first removes a target address from the sorted list, builds a /31 subnet for it, and then iteratively decreases its prefix length (i.e., /30, …/x+1, /x subnets), and retrieves all interfaces encompassed in this expanded subnet from the initial list. Then it continues to check whether accommodated addresses are indeed in the same subnet, and also check if some contra-pivot(s) is (are) among them. This step involves additional probes during which the TTL of the probe packets changes to confirm the position of the target IPs. From the inference rule 2 presented in Section 3.2, we know that the path information of the interfaces under the same subnet differs only in the last two hops at most. In other words, knowing the full path of each subnet interface is useless for the subnet inference step. Therefore, NSTology does not perform a complete traceroute for each target IP address, but uses heuristic way to minimize the number of probes. When NSTology knows the TTL distance required to reach a given IP address, it utilizes this TTL in the first probe to the next IP address in the list. According to the result of the first probing, it estimates the TTL distance and obtain path trace from the address of key interfaces to the target address by performing some forward/backward probing (i.e., with increasing and decreasing TTL values) [34]. The probing is still multi-threaded to further speed up the whole process.

Next, this stage evaluates the entire situation to determine whether the prefix length should be further reduced or whether the expansion of the subnet should stop, with or without increasing subnet prefix size (i.e., subnet shrinkage). This obviously happens if the newly included address is not compatible with the subnet being inferred. In order to check whether the new addresses are indeed part of the current subnet, NSTology selects the first contra-pivot as the reference interface. Then, the newly accommodated IP addresses are compared with this reference interface to ensure that the new interfaces and the reference interface are on the same subnet. In other words, the candidate subnet needs to be determined whether it corresponds to the real subnet. To this end, NSTology applies the three inference rules presented in Section 3.2 to determine whether the probed candidate IP address is on the subnet. The checked IP address violating one of the rules means that it is not on the subnet being built (i.e., the found interface is located on the pivot TTL+1). Therefore, the subnet growth process stops immediately, and the subnet shrinks to its last known valid state (i.e., the previous subnet prefix) by removing all interfaces that are not in valid state.

When all interfaces have been checked, NSTology also check how many new interfaces have been identified in the subnet. This helps NSTology to correctly infer continuous subnets (e.g., a consecutive of /30 subnets found in the same /24 prefix). As long as these subnets all have at least one contra-pivot interface, the pivot interfaces sharing the same path can all be seen at the same TTL distance. If more than 20% of the interfaces in subnet are contra-pivot interfaces, then the subnet is shrunk by one level to avoid keeping an inconsistent and unauthentic subnet in the inference results. In fact, these contra-pivot interfaces may be the pivots of another subnet.

In addition, subnet growth immediately stops if the amount of responsive IP addresses within the new subnet is equal to or less than one-third of the total number of IP addresses that the current subnet prefix can accommodate. A subnet S_x^p can encompass $2^{32-p} - 2$ IP addresses for assignment, thus we require a fraction of thses alive addresses (e.g., one third of them) appear in the subnet. This completeness requirement helps us increase the probability in the accuracy

of the inferred subnet. Without such requirement, it is easy to use a few IP addresses belonging to the same subnet range to form a candidate subnet (probably a smaller one). Depending on the completeness, only considering this condition may result in discarding a real subnet, that is, /28 and instead consider one or more smaller subnets of size /29, /30, or /31 to meet the completeness condition.

However, due to the limited number of alive IP addresses in the candidate subnet, it is still difficult to verify whether the corresponding real subnet exists. After satisfying the above completeness condition, we make further verification checks on the current subnet to solve this issue. (1) if the candidate subnet contains at least a contra-pivot interface, NSTology stops the inference and returns the subnet with its prefix length; (2) if the candidate subnet does not contain a contra-pivot interface, NSTology continues to expand the subnet until one or several contra-pivot interfaces are found, then subnet growth stops and save the subnet. Otherwise, NSTology will stop when the current subnet overlaps other subnets and the contra-pivot interface cannot be found. By in-depth consideration of whether the candidate subnet contains contra-pivot interfaces, NSTology can guarantee the accuracy of the inferred subnets.

Finally, when NStology receives all the final subnets, it merges the results with the previously inferred subnets covering the same address range to ensure the uniqueness of each subnet at the end of the inference.

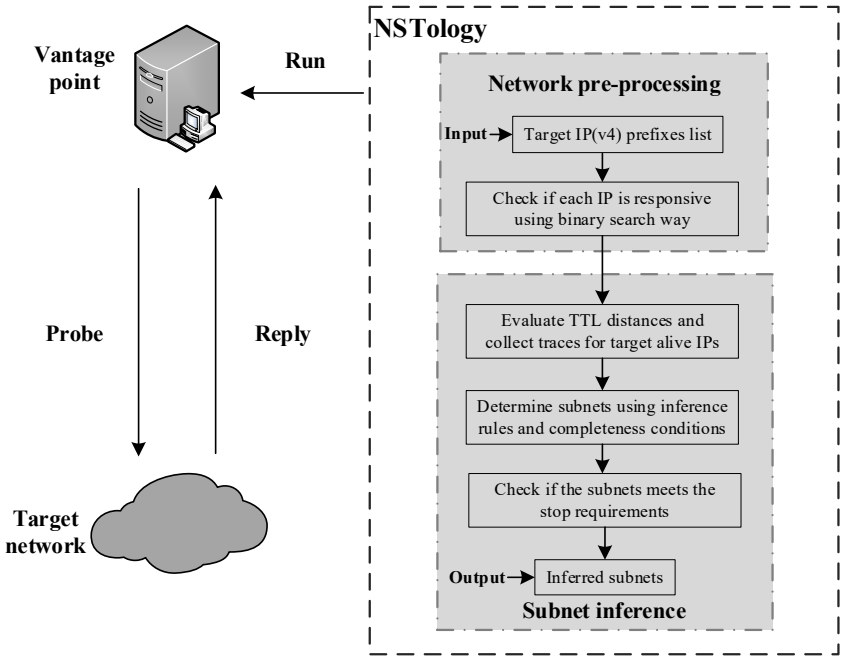


Figure 4. The schematic diagram of NSTology, which contains network pre-processing and subnet inference stages

4.3 Probing Complexity

In network preprocessing stage, since two rounds of running are repeated, the probing overhead at this stage is $O(M)$ (where M is the number of IP addresses in the target network). For subnet inference stage, there are wo worst-case scenarios resulting in a given interface to be compared multiple times. The first case is that the subnet S is a point-to-point link with only one interface (for example, the /31 prefix). The IP addresses will be compared with the next subnet for a second or even a third time. In this scenario, all addresses are considered at most three times. The other worst scenario is that each subnet S found is a multi-access link that accommodates $|S| > 2$ interfaces. The network includes a lot of consecutive sunbets whose prefix length gradually increases (e.g., /26, /27, then /28, all of which are included in a /25 prefix). The interfaces of the current subnet includes all the interfaces of the previous subnet and then they will be repeatedly put back in the list while the size of the subnet keeps getting smaller. Then, the interfaces of the smallest subnet will be compared based on how many times they are subnets. Consequently, the total number of comparisons is $\sum_i N/i \approx 2N$ (where N is the number of alive IP addresses). In summary, the probing complexity of NSTology is linear.

5 Experimental Results

In this section, we first evaluate the accuracy and completeness of NSTology, and then compare it with state-of-the-art subnet discovery techniques.

5.1 NSTology Effectiveness

In order to evaluate NSTology, we implemented it on a AS4711 network (INET Inc.), and compare the collected subnets with the list of subnets that we use the information provided by reasearch networks. AS4711 network is a stub AS (i.e., all traffic going in and out of it goes through one path), with a scale of 34,816 hypothetical IP addresses. Due to the privacy and commercial nature, we cannot access the real topology information of AS4711. Therefore, we use the results collected from PlaneLab testbed [35] as ground truth

and compare our results with them. NSTology dynamically collects subnets when tracking target IPs. However, we cannot control the path of data packets to pass through a certain subnet or router. Thus, we construct the target IP addresses list by selecting random IP addresses from each original subnet prefix in AS4711 [36].

First, in the stage of pre-processing, NStology finds 934 responsive IP addresses. After subnet inference, NStology discovers 348 subnets in the collected topology. Moreover, among the discovered subnets, there are 232 subnets with only one appropriate contra-pivot interface, 56 subnets for two or more contra-pivot interfaces, and 60 subnets without a contra-pivot interface. To quantify the results, we define accuracy as the corretly percentage of subnets w.r.t. the real subnets. Similary, precision can be denoted the corretly percentage of subnets w.r.t. the whole collected subnets.

Table 2 compares the real subnet with the subnet inferred from the collected data. In the table, the first row (*real*) shows the number of each /*x* subnet in the original topology. The second row (*exact*) shows the distribution of identified subnets, which is exactly the same as the real topology. The third (*miss*) and fourth (*under*) raws refer to missing subnets whose IP addresses could not be observed at all and underestimated subnets that are inferred to be smaller than the real ones. The fifth (*over*) raw shows the overestimated subnet distribution. Finally, ∅ line indicates the number of real subnets whose IP addresses are not in any of inferred subnets.

Table 2. Comparison of real and inferred subnets

	/24	/25	/26	/27	/28	/29	/30	/31	Total
real	2	0	4	6	25	74	138	127	376
exact	1	0	2	4	12	36	92	98	245
miss	0	0	0	1	4	16	28	24	73
under	1	0	2	1	8	18	15	0	45
over	0	0	0	0	1	3	1	0	5
∅	0	0	0	0	0	1	2	5	8

Among the 348 subnets inferred in the collected topology, we can verify 245 subnets that are part of the real subnet topology. The accuracy of NSTology is 65.16% and the precision is 70.4%. For most subnets sizes, we can identify the subnets accurately and completely. However, in the address ranges that we sent probes to, there are some subnets that have not been identified. After collecting subnets, we further probe each IP address in the address range of missing subnets and underestimsated subnets to identify unresponsive subnets. This situation occurs when there are partially and totally unresponsive IP addresses. For example, even though the correct size with /30 subnets is high, NSTology still underestimates/misses most of /29 and /28 subnets. Studying those missed or underestimated subnets further by probing each possible IP address in the real subnet indicates that those subnets mostly do not respond to probing. When analyzing one of underestimated /28 subnets, we found that only a small number of consecutive IP addresses are observed to be used for inferring subnet. Similarly, some subnets, i.e., ∅ set, have fewer observable IP addresses in the collected traces. By checking the results, we found that almost all unidentified subnets remain unresponsive to any ICMP messages. In other words, either probing packets or their responses are filtered out, or those subnets are not implemented even though they are announced as existing. In addition, some subnets are inferred to be larger than they actually are. This usually happens when two subnets share a common router and a common prefix.

Then, we analyze the completeness rate of each inferred-subnet in NSTology. Completeness rate is defined as the proportion of surviving IP addresses to the prefix length (or subnet mask) of a subnet. Figure 5 shows the completeness distribution of the /24 to /29 subnets (after meet completeness condition). Note that, all /31 and /30 subnets are always 100% complete. Hence, these two subnets are not shown in the figure. In addition, NSTology do not find the /25 subnet. Clearly, the /29 subnet has the highest completeness rate and the largest number of subnets, and the /24 subnet has the lowest integrity rate, which is only 37%. As the prefix length of the subnet decreases, the completeness rate of the subnet is also less. In addition, the completeness of the /27 subnet is more evenly distributed. More importantly, if the subnet selection threshold is set to 50%, none of the subnets of size /24 to /27 can be utilized.

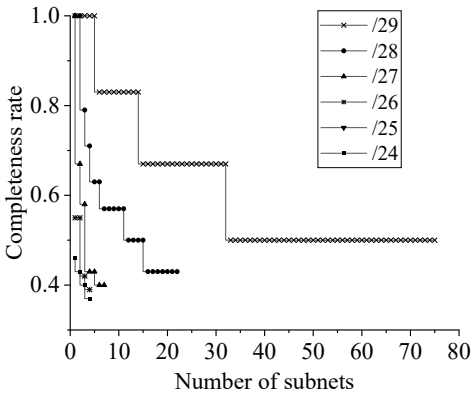


Figure 5. Completeness distribution of subnets

As a comparison, we applied the most widely used methods, namely XNet and TreeNet, to identify the subnets. The distribution of discovered subnets is summarized in Table 3. The first row represents the distribution of real subnets, and the last column is the total number of subnets obtained by each method. Overall, NSTology finds the least number of subnets. This mainly happens because NSTology can not find a part of the point-to-point links containing the contra-pivot interface, and misses the probing for some subnets. The XNet assumes that a subnet necessarily holds only a contra-pivot interface, Therefore, it tends to divide a large subnet into multiple smaller (incomplete) subnets. For example, our NSTology can discover 348 subnets and 245 of them are inferred correctly. However, the subnets found by the XNet method are not suitable for various sizes of subnets in real situation. The real larger subnets are divided into /27, /28, /29, and /30 subnets, resulting in the underestimation of the subnets. For point-to-point links, XNet also merges some smaller subnets into larger ones. Although TreeNet can accurately find larger subnets by introducing further refinement, it

also tends to divide a large subnet into multiple smaller ones when only a few alive IP interfaces in the subnet.

Table 3. Comparison of the number of subnets identified by different methods

	/24	/25	/26	/27	/28	/29	/30	/31	Total
Real	2	0	4	6	25	74	138	127	376
XNet	0	12	16	35	55	83	154	132	487
TreeNet	1	1	2	16	39	78	106	110	353
NSTology	1	0	2	8	32	86	116	103	348

Table 4 indicates the results of different methods in terms of accuracy, precision and running time. It can be seen that NSTology performs overall better than state-of-the-art approaches. XNet will stop the subnet inference once there exists a backup contra-pivot interface in the network. This case prevents XNet from correctly inferring a real larger subnet. Although TreeNet uses a refinement stage to increase the coverage of the subnet, it is inclined to the overgrowth of the subnet. NSTology can perform even better in the case of only partial responsive addresses in the subnet. However, NSTology still fails to identify a lot of subnets. By checking the tracking data results, we found that some routers discarded all ICMP probe packets. Under this scenario, the designed inference rules may not work. We believe this is one of the most difficult challenges in the research of topology discovery based on *traceroute*.

Table 4 also shows the overall execution time of three methods on the network. It can be seen that NSTology completes subnet discovery faster than TreeNET. In fact, NSTology only takes about 61% of the execution time of completes subnet discovery faster than TreeNET. In fact, NSTology only takes about 61% of the execution time of TreeNet, mainly because TreeNET considers a single subnet at a time in a given thread (the larger the subnet, the slower the execution speed), while NSTology utilizes the same thread to probe and infer IP addresses with a steady speed. However, XNet costs the longest time to probe and infer subnets. This is mainly because it neither considers preemptively multithreaded probing nor effectively filters the wrong subnets. Obviously, NSTology is also more probing intensive, mainly due to the way it scans target IP addresses and schedules probing work (i.e., re-probing target IP addresses that cannot be successfully scanned) to get all the subnets it does as fast as possible data needed for inference, but still relatively reasonable.

Table 4. The perfomance comparison of XNet, TreeNet and NSTology

	Accuracy	Precision	Execution Time
XNet	48.92%	53.75%	34m15s
TreetNet	61.85%	68.26%	18m39s
NSTology	65.34%	70.4%	11m16s

6 Topology Evaluations

In this section, we exploit NSTology to discover the subnet-level topology of six geographically dispersed AS networks, and use these topologies to analyze their subnet characteristics.

6.1 Measurement

First, we use the BGP Toolkit of Hurricane electric [37] to select ASes of different sizes and roles in the Internet topology. The six different ASes and their respective IPv4 prefixes are listed with a number of potential addresses varying from slightly less than 100,000 to a bit more than 1 million. Next, in order to guarantee we have different configuration files in the list, we utilize the AS relationship provided by CAIDA [38]. Table 5 lists all the ASs we have probed, their respective names, types (i.e., the levels in the AS hierarchy), and the number of potential addresses. For clarity, we also assign a number to each AS to represent them in subsequent figures. Finally, we exploit NSTology to form the set of alive IP addresses and collect inferred subnets given in Table 5.

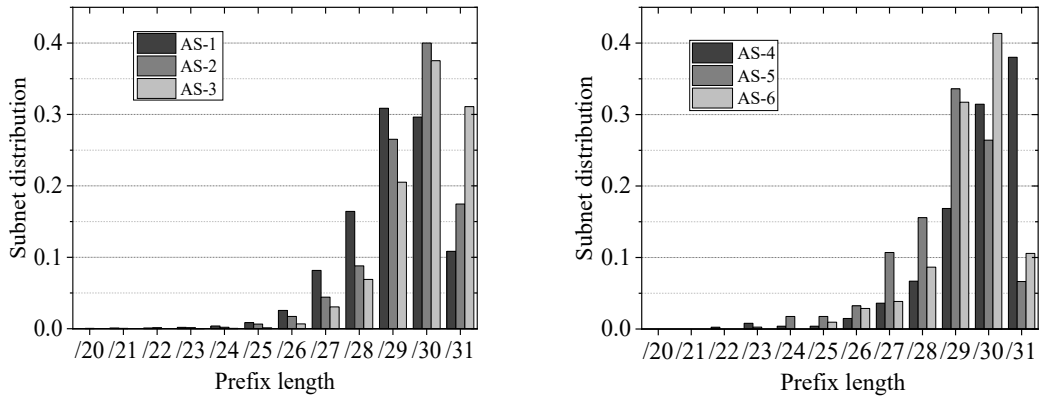
Table 5. Target ASes of measurement

Index	ASN	Name	Type	#IPs	Alive IPs	Subnets
1	109	Cisco Systems	Stub	1,165,568	11,580	1054
2	2764	AAPT Limited	Transit	844,544	75,484	4007
3	6453	TATA Com.	Tier-1	784,384	52,520	747
4	5511	Orange S.A.	Transit	551,680	21,617	3792
5	13789	Internap Net.	Transit	93,952	8,131	738
6	24369	CERNET2 IX	Stub	464,896	2,322	114

6.2 Prefix Length Prefix Distribution

The subnet prefix length denotes a quantitative measure of the capacity of a subnet. In this experiment, we analyzed the subnet prefix distribution patterns of the target ASes and their summary statistics.

Figure 6(a) and Figure 6(b) show the prefix length distribution of target ASes. The ordinate represents the proportion of subnets with a certain prefix length collected in a specific AS network. Brieftly, the results show that 73.4% of the subnets in the backbone of ASes are point-to-point links consisting of /31 and /30 subnets, while these subnets only accommodate 21.1% of successful subnet IP addresses. Nevertheless, since multi-access links can hold a large number of IP addresses (78.9%), subnets with smaller prefixes (larger capacity) constitute an important part of the backbone of these AS networks. The /31 point-to-point link subnet (RFC 3021) was introduced a few years after the standard subnetization procedure (RFC 950) with the purpose of improving the utilization of IP addresses. However, research on the prefix length distribution shows that, with the exception of TATA Communications (AS-3) and Orange S.A. (AS-4), the /31 subnet does not dominate the point-to-point links. When analyzing the prefix length trends of ASes, it can be found that as the prefix length decreases, the number of observed subnets decreases faster. However, many ASes break this trend at /24. The reason behind this may be that compared with /25 or /23, the /24 prefix length is a popular choice for constructing subnets and exploring the Internet periphery may reveal more /24 subnets.



(a) AS-1, AS-2 and AS-3 subnet distribution (b) AS-4, AS-5 and AS-6 subnet distribution

Figure 6. Subnet prefix length distribution of each AS

As can be seen from Table 6, the average prefix length of six ASes is 29.77 and the median is 30, which shows that the majority of the subnets are point-to-point links. The mean of subnet prefixes of AS-3 and AS-4 are approximately /31, while AS-1, AS-2, and AS-6 are closer to /29. The relatively large standard deviations of AS-4 and AS-5 indicate the variability in the utilization of the subnet, and therefore the size of the subnet is high, while TATA Communications (AS-3) prefers a more stable subnet deployment strategy.

Table 6. Subnet prefix length statistics

	AS-1	AS-2	AS-3	AS-4	AS-5	AS-6
Median	29	30	30	30	29	30
Mean	29.03	29.47	30.84	30.77	29.15	29.34
Std	1.36	1.24	1.1	1.47	1.47	1.18

6.3 Subnet Degree Analysis

The degree of a subnet is the number of interfaces (or alive IP addresses) that the subnet can accommodate. In this subsection, we carefully study the degree cumulative distribution and statistics of the six ASes, and identify common and discrete subnetization practices.

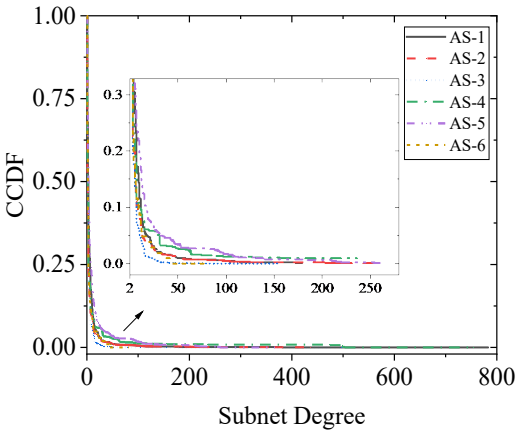


Figure 7. Degree distribution CCDFs for public ASes

Figure 7 presents the subnet degree complementary cumulative distribution functions (CCDFs) for each of the six ISPs. Due to scalability issues, we zoomed at least 80% of the subnet in the figure captured by each AS. The enlarged part is also inclined in Figure 7. It can be seen that subnet with degree of two constitute most of the subnets. The CCDF curve of AS-1 starts with 51% at degree two, and the one of AS-5 starts at 77%, while the remaining CCDF curves take values in between. This indicates that the median degree (50th percentile) of all ASes is 2, and their degree distributions are highly skewed.

Table 7 shows the statistical results of each AS, including the maximum, minimum, median, mean and standard deviation. Among six ASes, Orange S.A. (AS-4) and Internap Network (AS-5) have the highest mean degree values. This is because these two AS networks have a small number of subnets and also contain subnets with a larger prefix length. Furthermore, the number of large subnets in AS4 and AS5 is smaller than that of small subnets, which explains the high standard deviations in the subnets of these two ASes. In TATA Communications (AS-3) network, 68% of subnet degrees have a degree of 2, and there has only one /23 subnet. Thus, the mean degree and median degree of AS-3 are very close. The small standard deviation indicates that the degree distribution in the network is more stable. The statistical indicators of subnets for AS-6 are lower than other ASes. This is because the number of subnets discovered is small and the maximum subnet prefix length is only /25.

Table 7. Subet degree statistics

	AS-1	AS-2	AS-3	AS-4	AS-5	AS-6
Max	784	3009	497	755	637	82
Min	2	2	2	2	2	2
Median	4	3	3	3	4	3
Mean	7.68	6.69	4.25	11.56	11.13	5.73
Std	29.12	53.8	9.51	54.38	28.3	10.04

In order to verify the accuracy of the subnet with the largest degree in AAPT Limited (AS-2), we randomly chosen a large number of target IP address from subnets by DNS name resolution query. We found that all these subnets belonged to Akamai Technologies, which is an online content distribution service provider in the Internet. Since Akamai deploys its content server in AS network, these hosts appear on AS networks instead of Akamai network. Moreover, the DNS name of all IP addresses in the subnet share the same prefix as a72 in a72-247-183-101.deploy.akamaitechnologies.com. These

large subnets are part of the Akamai data center and can be implemented at the data link layer, where interfaces communicate through bridges, or at the network layer.

6.4 IP Address Space Utilization

In this section, we analyze IP address space utilization pattern of the target ASes. The number of IP addresses utilized in a subnet can measure the effective utilization of the subnet, which is of great significance for the construction of the subnet and topology analysis. Hence, the entire utilization rate U for each AS network can be defined as

$$U = \frac{\sum_{S_i} D_i}{\sum_{S_i} C_i},\ S_i \in AS\ . \tag{4}$$

where D_i represents the degree of a subnet, i.e., the number of current alive IP addresses utilized, and A_i denotes the prefix length (or subnet mask) of a subnet S_i.

Figure 8 shows the overall subnet utilization rate of each target AS. Obviously, AS-4 (Orange S.A.) has the highest subnet utilization rate. By combining Figure 6, the utilization rate of each subnet in AS-4 (Orange S.A.) is higher than that of subnets of other ASes. By contrast, the subnet utilization rate of AS-1 (Cisco Systems) is the lowest. By cross checking the prefix length distribution of AS-1 (Cisco Systems) in Figure 6, it can be seen that the existence of an excessive /29 subnet in AS-1 is the main reason for its low utilization percentage. In fact, removing /30 subnets and recalculating the utilization rate for AS-1 (Cisco Systems) can lead to an increase of 6 percentages. Finally, in addition to the 73.4% importance of point-to-point links (/30 and /31) in Figure 6, Figure 8 indicates that these point-to-point links only hold 21.1% of the IP addresses that have been subnetted. In other words, the multi-access links also constitute an important part of the AS network, which carries 78.9% of the subnet addresses.

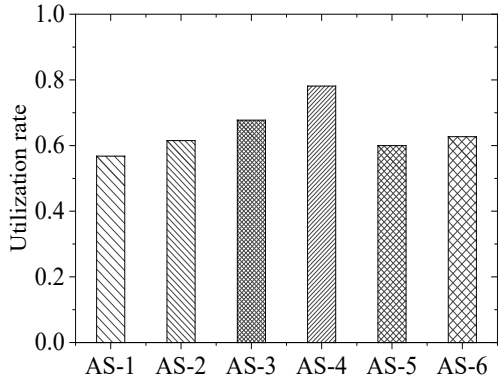


Figure 8. The overall subnet utilization rate of each AS

7 Conclusion

In this paper, we proposed a novel lightweight subnet discovery approach called NSTology, which collects and derives subnet topology information. NSTology introduces a pre-processing stage to reduce probing overhead and makes use of complementary inference rules to discover subnets. The experimental results show that compared with the state-of-the-art approaches, NSTology can achieve 65.16% accuracy and 70.4% precision and is also capable of outperforming them in terms of probing time due to its linear complexity. In addition, we exploit NSTology to discover the subnet information of six ASes operating in different parts of the world. The statistics show that although 73.4% of the subnets are point-to-point links (/30 and /31), these point-to-point links only hold 21.1% of alive IP addresses of the subnet. The rest of the IP addresses are hosted by multi access links with different sizes. In the future, we plan to combine alias resolution technique and consider extending NSTology for IPv6 to establish a suitable model for router-subnet topology.

Acknowledgements

This paper was supported by research grant from the Fundamental Research Funds for the Central Universities (no. 2020GFZD014) and the National Key Research and Development Program of China (no. 2019JSJ12ZDYF01).

References

[1] G. Tilch, T. Ermakova, B. Fabian, A multilayer graph model of the internet topology, *International Journal of Networking and Virtual Organisations*, Vol. 22, No. 3, pp. 219-245, March, 2020.

[2] R. Motamedi, R. Rejaie, W. Willinger, A Survey of Techniques for Internet Topology Discovery, *IEEE Communications Surveys and Tutorials*, Vol. 17, No. 2, pp. 1044-1065, December, 2015.

[3] E. Gregori, L. Lenzini, V. Luconi, AS-Level Topology Discovery: Measurement Strategies Tailored for Crowdsourcing Systems, *Computer Communications*, Vol. 112, pp. 47-57, September, 2017.

[4] R. Motamedi, B. Yeganeh, B. Chandrasekaran, R. Rejaie, B.M. Maggs, W. Willinger, On Mapping the Interconnections in Today's Internet, *IEEE/ACM Transactions on Networking*, Vol. 27, No. 5, pp. 2056-2070, October, 2019.

[5] K. Vermeulen, J. P. Rohrer, R. Beverly, O. Fourmaux, T. Friedman, Diamond-Miner: Comprehensive Discovery of the Internet's Topology Diamonds, *USENIX Conference on Networked Systems Design and Implementations (NSDI)*, Santa Clara, USA, 2020, pp. 479-494.

[6] G.-X. Sun, S. Bin, Router-Level Internet Topology Evolution Model Based on Multi-Subnet Composited Complex Network Model, *Journal of Internet Technology*, Vol.18, No. 6, pp. 1275-1283, November, 2017.

[7] C.-F. Lai, H.-Y. Weng, H.-Y. Chou, Y.-M. Huang, A Novel NAT-based Approach for Resource Load Balancing in Fog Computing Architecture, *Journal of Internet Technology*, Vol. 22, No. 3, pp. 513-520, May, 2021.

[8] K. Bakhshaliyev, M. A. Canbaz, M. H. Gunes, Investigating Characteristics of Internet Paths, *ACM*

Transactions on Modeling and Performance Evaluation of Computing Systems, Vol. 4, No. 16, pp. 1-24, September, 2019.

[9] D. Feldman, Y. Shavitt, N. Zilberman, A Structural Approach for PoP Geo-Location, *Computer Networks*, Vol. 56, No. 3, pp.1029-1040, February, 2012.

[10] G. Nomikos, X. Dimitropoulos, traIXroute: Detecting IXPs in Traceroute Paths, *Proceedings of the 17th Passive and Active Measurements Conference (PAM)*, Heraklion, Greece, 2016, pp. 346-358.

[11] M. E. Tozal, K. Sarac, Estimating Network Layer Subnet Characteristics via Statistical Sampling, *Proceedings of International Conference on Research in Networking (IFIP Networking)*, Prague, Czech Republic, 2012, pp. 274-288.

[12] J. Grailet, B. Donnet, Towards a Renewed Alias Resolution with Space Search Reduction and IP Fingerprinting, *2017 Network Traffic Measurement and Analysis Conference (TMA)*, Dublin, Ireland, 2017, pp. 1-9.

[13] J. Mogul, J. Postel, Internet Standard Subnetting Procedure, *Internet Engineering Task Force*, RFC 950, August, 1985.

[14] M. Gunes, K. Sarac, Inferring Subnets in Router-Level Topology Collection Studies, *Proceedings of ACM/ USENIX Internet Measurement Conference (IMC)*, New York, NY, USA, 2007, pp. 203-208.

[15] M. E. Tozal, K. Sarac, TraceNET: an Internet Topology Data Collector, *Proceedings of the 10th ACM Internet Measurement Conference (IMC)*, New York, NY, USA, 2010, pp. 356-368.

[16] M. E. Tozal, K. Sarac, Subnet Level Network Topology Mapping, *Proceedings of the 30th IEEE International Performance Computing and Communications Conference (IPCCC)*, Orlando, FL, USA, 2011, pp. 1-8.

[17] J.-F. Grailet, F. Tarissan, B. Donnet, TreeNET: Discovering and Connecting Subnets, *Proceedings of the 8th Traffic and Monitoring Analysis Workshop (TMA)*, Louvain la Neuve, Belgium, 2016, pp. 1-8.

[18] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, J.-J. Pansiot, Topology Discovery at the Router Level: a New Hybrid Tool Targeting ISP Networks, *IEEE Journal on Selected Areas in Communication*, Vol. 29, No. 6, pp. 1776–1787, October, 2011.

[19] P. Mérindol, B. Donnet, O. Bonaventure, J.-J. Pansiot, On the Impact of Layer-2 on Node Degree Distribution, *Proceedings of the 10th ACM/USENIX Internet Measurement Conference (IMC)*, Melbourne, Australia, 2010, pp. 179-191.

[20] J.-F. Grailet, B. Donnet, Revisiting Subnet Inference WISE-ly, *2019 Network Traffic Measurement and Analysis Conference (TMA)*, Paris, France, 2019, pp.73-80.

[21] V. Jacobson, Traceroute, *UNIX*, man page, Feburary, 1989.

[22] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, J.-J. Pansiot, Quantifying and Mitigating IGMP Filtering in Topology Discovery, *2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, USA, 2012, pp. 1871-1876.

[23] M. Sun, J. Wang, Discovering Subnets in Router-Level Topology Studies, *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, Xi'an, China, 2011, pp. 68-72.

[24] H. Kardes, M. Gunes, T. Oz, Cheleby: A Subnet-Level Internet Topology Mapping System, *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS)*, Bangalore, India, 2012, pp. 1-10.

[25] A. Takada, N. Hayashi, M. Nakamura, T. Seki, K. Yamagoe, Topology Discovery Method Using Network Equipment Alarms, *International Conference on Network and Service Management(CNSM)*, Izmir, Turkey, 2020, pp. 1-5.

[26] K. Keys, Y. Hyun, M. Luckie, K. Claffy, Internet-Scale IPv4 Alias Resolution with MIDAR, *IEEE/ACM Transactions on Networking*, Vol. 21, No. 2, pp. 383–399, April, 2013.

[27] K. Keys, Internet-scale IP alias resolution techniques, *ACM SIGCOMM Computer Communication Review*, Vol. 40, No. 1, pp. 50-55, January, 2010.

[28] M. A. Canbaz, K. Bakhshaliyev, M. H. Gunes. Router-level Topologies of Autonomous Systems, *International Conference on Complex Networks*, Boston, USA, 2018, pp.243-257.

[29] E. Marechal, P. Mérindol, B. Donnet, ISP Probing Reduction with Anaximander, *International Conference on Passive and Active Network Measurement (PAM)*, Online, 2022: pp. 441-469.

[30] A. Y. Nur, Analysis of Autonomous System Level Internet Topology Graphs and Multigraphs, *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, Dubai, United Arab Emirates, 2021, pp.1-7.

[31] J.-F. Grailet, B. Donnet, Virtual Insanity: Linear Subnet Discovery, *IEEE Transactions on Network and Service Management*, Vol. 17, No. 2, pp. 1268-1281, February, 2020.

[32] J.-F. Grailet, B. Donnet, Travelling Without Moving: Discovering Neighborhood Adjacencies, *2021 Network Traffic Measurement and Analysis Conference (TMA)*, Virtual, 2021, pp. 1-8.

[33] K. Bakhshaliyev, M. H. Gunes, Generation of 2-mode Scale-free Graphs for Link-level Internet Topology Modeling, *PLOS ONE*, Vol. 15, No. 11, pp. 1-23, November, 2020.

[34] B. Donnet, P. Raoult, T. Friedman, M. Crovella, Deployment of an Algorithm for Large-Scale Topology Discovery, *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 12, pp. 2210-2220, December, 2006.

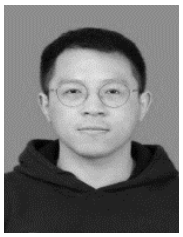
[35] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, PlanetLab: An Overlay Testbed for Broad-Coverage Services. *ACM/ SIGCOMM Computer Communication Review*, Vol. 33, No. 3, pp. 3-12, July, 2003.

[36] Geoff Huston, CIDR, 2021, https://www.cidr-report.org/as2.0/

[37] Hurricane Electric. http://bgp.he.net.

[38] The CAIDA UCSD, The AS relationship, 2013, http://data.caida.org/datasets/as-relationships/.

Biographies



Wei Yao received the M.S. degree in computer system architecture from Northeastern University, Shenyang, China, in 2018. He is currently pursuing the Ph.D. degree in computer science and technology from Northeastern University, Shenyang, China. His research interests include Internet of Things security, anomaly detection, e-healthcare, and network measurement.

Hai Zhao received the M.S. and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 1987 and 1995, respectively. He is the Director of the Liaoning Provincial Key Laboratory of Embedded Technology. His current research interests include Internet of Things security, deep learning, and wireless sensor networks.

Jing-Jing Chen is currently pursuing the M.S. degree in communications engineering from Northeastern University, Shenyang, China. Her research interests include cyber security and network measurement.