Topology Inference in the Presence of Anonymous Routers

Bin Yao, Ramesh Viswanathan, Fangzhe Chang, Daniel Waddington
Bell Laboratories, Holmdel, NJ

Abstract— **Many topology discovery systems rely on traceroute to discover path information in public networks. However, for some routers, traceroute detects their existence but not their address; we term such routers *anonymous routers*. This paper considers the problem of inferring the network topology in the presence of anonymous routers. We illustrate how obvious approaches to handle anonymous routers lead to incomplete, inflated, or inaccurate topologies. We formalize the topology inference problem and show that producing both exact and approximate solutions is intractable. Two heuristics are proposed and evaluated through simulation. These heuristics have been used to infer the topology of the 6Bone, and could be incorporated into existing tools to infer more comprehensive and accurate topologies.**

I. Introduction

Understanding network topologies is important to protocol evaluation, performance optimization, and network management [1], [2], [3], [4], [5]. To probe the topology of large scale public networks, many topology discovery systems rely on traceroute [6]. Traceroute sends a series of hop-limited UDP packets to a known destination and uses ICMP responses generated by intermediate routers to obtain the path between the source and the destination. In order to obtain comprehensive topology information, most existing systems run traceroute from multiple probing hosts [7], [8], [9], [10], [11] or use source-routed traceroutes [12], [13]. Two routers appearing consecutively on a path implies the existence of a link between them; this *induces* a *natural construction* of the underlying topology from the traceroute probe results. We have developed an IPv6 topology discovery tool, Atlas [13], which exploits the widespread support of source-routing in IPv6 routers to probe paths between all pairs of discovered routers.

¿From our experience with Atlas, and in results from other topology discovery works that probe the IPv4 Internet, we have observed the following phenomena: some routers do not send out ICMP responses while others use the destination addresses of traceroute packets instead of their own addresses as source addresses for outgoing ICMPv6 packets. In both cases, the presence of a router, but not its address, can be detected by traceroute. We call such routers *anonymous routers*, and routers that return their addresses *known routers*. IPv4 and IPv6 routers with ICMP disabled behave anonymously. We have additionally observed that IPv6 routers not configured with global addresses also appear as anonymous. Such router configurations exist for numerous reasons. For example, not assigning routers IPv6 global addresses reduces administrative overhead. Anonymous routers are also less likely to be the target of malicious attacks and allow ISPs to keep their network topologies opaque. Convenience, security, and privacy concerns are therefore some reasons for the continued presence of anonymous routers.

Deriving the actual topology from a traceroute probe result becomes significantly more complicated in the presence of anonymous routers. In the probe result, each router appears multiple times, once in each probed path that traverses the router. For a known router, these multiple occurrences can be resolved as belonging to the same router either because of a common interface address or because additional probing resolves its different interfaces as belonging to the same router [13], [12], [14]. On the other hand, for an anonymous router, there is no way to distinguish its occurrences from those of other anonymous routers. Therefore, each occurrence is treated as a potentially distinct router. Consequently, the natural construction of a topology from a probe result would be inaccurate. Consider, for example, the topology of Fig. 1, which has two anonymous routers 7 and 8 (represented by squares), with all others being known routers (represented by circles). The topology constructed from its probe result is that in Fig. 2, which contains twenty anonymous routers. Therefore, even the presence of a few anonymous routers can significantly distort the constructed topology.

At the very least, deriving an accurate topology requires identifying which occurrences of anonymous routers in the probe result correspond to the same router. This identification cannot be done using additional probing: the addresses of the anonymous routers are unknown and therefore cannot be used as the source or destination of any probe packets. Thus, the actual topology must be deduced purely through analysis.

We observe that solving this problem is non-trivial by examining the shortcomings of some straightforward approaches. Some existing topology discovery tools stop probing a path when an intermediate router fails to respond [7], thereby avoiding anonymous routers in the probe results. This approach, however, would result in a topology with reduced coverage and loss of connectivity information even among the discovered routers. For example, in the topology of Fig. 1, this approach would fail to capture that nodes 2 and 3 are connected to each other or to other known nodes. A topology constructed in which all anonymous occurrences in the probe results are treated as distinct routers (the topology naturally induced by the probe result) better reflects the connectivity information of the underlying topology. However, as in Fig. 2, the resulting topology can be greatly inflated.

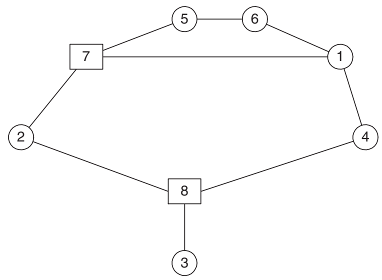

Fig. 1. Actual network topology

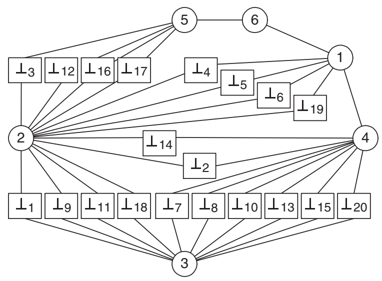

Fig. 2. Topology induced from probe result

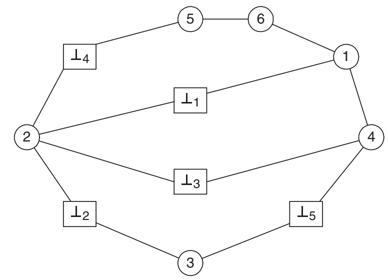

Fig. 3. Topology using bisimilarity

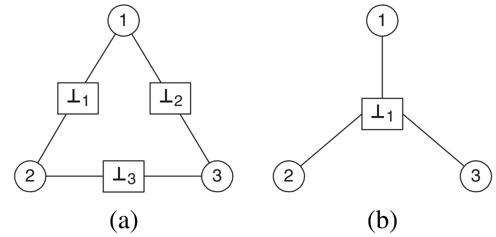

Fig. 4. Using raw path data for topology synthesis

A more sophisticated approach would be to infer that any two anonymous nodes that are adjacent to the same set of known routers can be identified as being the same. This idea could be generalized, using the notion of bisimilarity [15], to identify all anonymous routers that are bisimilar in the graph of the induced topology. Applying this to the topology of Fig. 2 would result in that of Fig. 3. This is still quite different from the actual topology (Fig. 1), both in the number of anonymous nodes and in its topological structure. This approach can be shown, in general, to always produce a topology in which every anonymous node would have exactly two neighbors, which would often be inaccurate. Other criteria, therefore, need to be developed for analyzing which anonymous routers in the induced topology should be identified as being the same.

In this paper, we explore a more systematic approach to inferring topology in the presence of anonymous routers. Since the topology inferred must be consistent with the traceroute results obtained by probing, we first formulate some properties based on the probe result, that any topology must satisfy to yield the probe result. This underlines the importance of a subtle distinction between probe results and their induced topology. Consider the following two probe results: (a) S_1, in which the traceroutes $1\perp_1 2$, $2\perp_2 3$, and $3\perp_3 1$ are observed, and (b) S_2 in which the traceroutes $1\perp_1 2\perp_2 3$ and $3\perp_3 1$ are observed, where the nodes $\perp_1, \perp_2, \perp_3$ denote anonymous nodes. For both these probe results, the induced topology is the same, namely, that given in Fig. 4(a). However, the topology of Fig. 4(b) is consistent with the first set of observed paths but not the second, since it contains no simple path (*i.e.*, one without routing loops) whose trace could be $1\perp_1 2\perp_2 3$. Thus, the topology of Fig. 4(b) would be acceptable if the probe result had been S_1 but not if it had been S_2. This example shows that, in the presence of anonymous routers, the topology should be inferred directly from the raw path data rather than by analyzing the induced topology.

In this paper, we characterize the constraints on a topology for it to be consistent with some observed probe results, and formulate the topology inference problem as an optimization problem subject to these constraints. We show that the structure of any consistent topology can be obtained from the induced topology by merging anonymous nodes that meet certain conditions. These conditions define the criteria under which anonymous nodes generated in the probe result can be identified as being the same actual router. We show that the topology inference optimization problem is NP-complete and approximating it within n^δ (where n is the size of the probe result, and δ a fixed constant) is NP-hard. These results imply that there are no polynomial-time algorithms that synthesize or approximate the actual topology within n^δ for all probe results (unless $P = NP$). We then develop heuristics that aim to synthesize the actual topology for most probe results. We present and establish the correctness of a pruning algorithm for reducing the size of the probe results, which is critical to inferring the topology of large networks. We evaluate our heuristics through simulations and experiments and show that they perform well for topologies with characteristics similar to what we observe on the 6Bone.

The rest of the paper is organized as follows. In Section II we formally define the topology inference problem. Section III establishes the technical connection between topologies consistent with probe results and their induced topologies. This result is used to establish the intractability and inapproximability of topology inference in Section IV, and to derive heuristics in Section V. Evaluation results of these heuristics are presented in Section VI and related work is discussed in Section VII.

We conclude in Section VIII.

II. The Topology Inference Problem

In this section, we formulate the topology inference problem. Section II-A formalizes probe results (the input to topology inference) and topologies (the output of topology inference), and Section II-B specifies requirements on the topology that should be produced for a given probe result.

A. Topologies and Probe Results

A topology is an undirected graph with nodes representing routers and edges representing links. Each node is annotated with a label representing the address of the router. For the remainder of the paper, let I be a set of possible addresses, not containing the special symbol $\perp$.

Definition 2.1 [Network Topology]

- A *network topology*, $\mathcal{T} = \langle V, E, L\rangle$, where V is a set of nodes, E is a set of undirected edges of the form $\{u, v\}$ with $u, v \in V$, and $L : V \to I \cup \{\perp\}$ is a labeling function such that for any $u, v \in V$, we have that if $L(u) = L(v) \neq \perp$ then $u = v$. The set of known nodes of the topology, $K(\mathcal{T}) = \{v \mid L(v) \neq \perp\}$, and its set of anonymous nodes, $A(\mathcal{T}) = \{v \mid L(v) = \perp\}$.
- A *route* in the topology $\mathcal{T}$ is a simple path in the graph $\langle V, E\rangle$, *i.e.*, a node sequence $v_1 \dots v_n$ with $n \geq 2$, $v_1 \neq \cdots \neq v_n$, and $\{v_i, v_{i+1}\} \in E$ for $1 \leq i < n$.
- The *trace* of a route $v_0 \dots v_n$ is the sequence $L(v_0) \dots L(v_n)$.
- The set *TraceRoute*$(\mathcal{T})$ of traces admissible by the topology $\mathcal{T}$ consists of all sequences $L(v_0) \dots L(v_n)$ such that $v_0 \dots v_n$ is a route in $\mathcal{T}$ and $v_0, v_n \in K(\mathcal{T})$.

A topology is, thus, simply an undirected graph equipped with a labeling function on nodes. The label, $L(v)$, of a node, v, indicates how the node advertises itself in the trace of any route that goes through that node. Known nodes advertise their addresses and anonymous nodes are labeled with the special symbol $\perp$. Taking L to be a function of nodes reflects the assumption that if a router is anonymous then it behaves anonymously in every route. The additional condition imposed on L captures the requirement that distinct routers cannot share the same unicast address. Finally, the definition of *TraceRoute*$(\mathcal{T})$ takes into account the fact that traceroutes can only be obtained between known nodes.

Strictly speaking, a router has one address for each of its interfaces and the label of a node should therefore be a collection of addresses. However, the probing phase can identify which addresses belong to the same router [12], [14], [13]; we can therefore assume that the set of traces in the probe result use a common representative address for each router. This assumption serves to simplify our presentation of the topology inference problem without affecting the applicability of its analysis.

The result of probing the network is a set of observed traces. Any anonymous node observed may be potentially distinct from any other anonymous node and is tagged uniquely in the probe result. We formally represent any anonymous node in the probe result as $\perp_k$ with k a natural number. A probe result therefore consists of sequences $u_1 \dots u_n$, where each $u_i \in I$ or of the form $\perp_k$, with the source, u_1, and destination, u_n, having to be known addresses (*i.e.*, $u_1, u_n \in I$).

Definition 2.2 [Probe Result] A *probe result*, $S \subseteq I(I \cup \{\perp_i \mid i \in N\})^* I$, such that:

- If $u_1 \dots u_n \in S$ then $u_1 \neq \cdots \neq u_n$
- For any k, there is at most one $p \in S$ such that $p = \sigma \perp_k \tau$ for some σ, τ.

The first condition in Definition 2.2 imposes the requirement that each trace observed does not include any routing loops, and the second condition ensures that every anonymous node in the probe result is tagged uniquely.

For any path $p = u_1 \dots u_n \in I(I \cup \{\perp_i \mid i \in N\})^* I$, define the known nodes appearing in p, $K(p) = \{u_1, \dots, u_n\} \cap I$, and the anonymous nodes $A(p) = \{u_1, \dots, u_n\} \cap \{\perp_i \mid i \in N\}$. We define the known nodes in the probe result, $K(S) = \cup_{p \in S} K(p)$, and the anonymous nodes, $A(S) = \cup_{p \in S} A(p)$.

The natural topology constructed from a probe result (*e.g.*, Fig. 2) is called its induced topology. The induced topology has edges between any pair of nodes that appear consecutively in the trace of some route. Its labeling function, *Erase*, makes every generated node of the form $\perp_k$ anonymous and every discovered interface a known node whose label is itself.

Definition 2.3 [Induced Topology] For any probe result S, define the topology *induced by* S, $T(S) = \langle V(S), E(S), \mathit{Erase}\rangle$, where $V(S) = K(S) \cup A(S)$, E(S) = $\{\{u, u'\} \mid \sigma u u' \tau \in S \text{ for some } \sigma, \tau\}$ and $\mathit{Erase}(u) = u$ if $u \in I$, and $\mathit{Erase}(u) = \perp$ otherwise.

B. Admissible Topologies

The goal of topology inference is to produce the topology of the underlying network on which exhaustive traceroutes yielded the observed probe result. We will write $\mathcal{T} \models \mathcal{S}$ to denote that probing the network topology $\mathcal{T}$ can yield the probe result S. Alternatively, $\mathcal{T} \models \mathcal{S}$ can be read as: if S is the probe result obtained then $\mathcal{T}$ is an admissible topology. In the presence of anonymous nodes, there may be more than one admissible topology for a given probe result S. For example, the topologies of Fig. 2, Fig. 3, and Fig. 1 can all yield the results obtained by probing Fig. 1. Among the admissible topologies, it seems reasonable to expect the actual topology to be closest to the one that has the fewest number of routers. Since the set of known nodes, for a probe result S, is fixed to be $K(S)$, the size of the topology is determined by the number of anonymous nodes in it.

Definition 2.4 [Minimum Topology] A *minimum topology* for a probe result S, with respect to an admissibility relation $\models$, is a topology $\mathcal{T}$ such that $\mathcal{T} \models \mathcal{S}$, and for any topology $\mathcal{T}' \models \mathcal{S}$, we have that $|A(T)| \leq |A(T')|$.

We now proceed to define the admissibility relation. The guiding principle is to reflect enough of the characteristics of exhaustive traceroute probing so that a minimum topology with respect to this relation would with high likelihood be very close to the real underlying topology. The first obvious requirement is that each trace observed in the probe result should be the trace of a route in the topology. The function, *Erase*, defined in Definition 2.3, is extended to paths by $\textit{Erase}(u_0 \ldots u_n) = \textit{Erase}(u_0) \ldots \textit{Erase}(u_n)$ and to probe results by $\textit{Erase}(S) = \{\textit{Erase}(p) \mid p \in S\}$. Essentially, the function "erases" the arbitrary tags generated for the anonymous nodes in the probe result.

Definition 2.5 [Trace Preservation] A topology $\mathcal{T}$ preserves the traces in a probe result S, written $\mathcal{T} \models^t S$ iff $\textit{Erase}(S) \subseteq \textit{TraceRoute}(\mathcal{T})$.

We do not insist that every trace of the topology should be in the probe result since some routes may not have been explored during the probing phase. Also, note that trace preservation is defined with respect to the probe result S rather than the induced topology $T(S)$, as discussed in Section I and illustrated by the topologies in Fig. 4.

The minimum topology with respect to the trace preservation requirement can be trivially obtained as follows. Let S be the probe result. Let $n = max_{p \in S}|A(p)|$ be the largest number of anonymous nodes appearing in any path in S. Then a minimum topology for S, with respect to $\models^t$, is the topology $T = \langle V, E, L \rangle$, where $V = K(S) \cup \{\perp_1, \ldots, \perp_n\}$, $E = \{\{u, v\} \mid u \neq v\}$, and $L(u) = u$ if $u \in K(S)$ and $L(u) = \perp$ otherwise. In other words, the topology T is a full-mesh on the known nodes $K(S)$ together with n anonymous nodes. Any trace that has at most n anonymous nodes is observable in this topology and therefore $T \models^t S$. It is minimum because there is a trace with n anonymous nodes in S and therefore any topology $\models^t S$ would need to have n distinct anonymous nodes for this to be a route (simple path).

This full-mesh topology, which is a minimum topology with respect to $\models^t$, is clearly unlikely to be close to the underlying topology. We, therefore, further refine our definition of an admissible topology. One intuition for why the full mesh topology is unacceptable is that nodes are too close together — every node is one-hop away from the other. Therefore, the additional restriction we place is that the shortest distance between known routers is no less than can be inferred from the probe result.

Distance preservation is formalized as follows. For a topology $\mathcal{T} = \langle V, E, L \rangle$, define $d_{\mathcal{T}}(v_1, v_2)$ for $v_1, v_2 \in V$ to denote the shortest distance from node v_1 to node v_2 in graph $\langle V, E \rangle$ with each edge counting as cost 1. Note that by our condition on the labeling function, for any interface $u \in I$, there is at most one node v with $L(v) = u$. For addresses $u_1, u_2 \in I$, we therefore define $d_T(u_1, u_2) = d_T(v_1, v_2)$ where $L(v_1) = u_1, L(v_2) = u_2$. We define the distance observed in the probe result, $d_S(u, u')$ to be $d_{T(S)}(u, u')$, where $T(S)$ is the induced topology (Definition 2.3). Note that $d_S(u, u')$ is not the length of an explicit path of the form $u \sigma u' \in S$. Rather, it is the shortest length of a path from u to u' that can be constructed by concatenating subpaths of any paths observed in S. Thus, the distance preservation criterion does not assume that every route traced during probing was the shortest one but only reflects the weaker assumption that a shortest route between any two known nodes can be constructed from all the links that were discovered in S.

Definition 2.6 [Admissibility] Let S be a probe result and $\mathcal{T}$ be a topology. Then $\mathcal{T} \models \mathcal{S}$ iff

- Trace Preservation: $T \models^t S$, and
- Distance Preservation: For any $u, v \in K(S)$, we have that $d_{\mathcal{T}}(u, v) \geq d_S(u, v)$.

Definitions 2.4 and 2.6 are used to define the topology inference problem.

Definition 2.7 *TOP-INF*: Given a probe result S, produce a topology $\mathcal{T} = \langle V, E, L \rangle$ that is a minimum topology for S with respect to $\models$. The objective function is the number of anonymous nodes $|A(\mathcal{T})|$. We denote the minimum of this objective function, for a probe result S, by $\alpha(S)$.

Our first observation is that even deciding admissibility is NP-complete.

Lemma 2.8: Given a topology $\mathcal{T}$ and a set S, deciding whether $\mathcal{T} \models \mathcal{S}$ is NP-complete.

Proof: The problem is in NP because for every trace $\sigma \in S$, we guess a route τ verifying that it is a simple path in $\mathcal{T}$ and that $L(\tau) = \textit{Erase}(\sigma)$. The minimum distance preservation criteria can be checked in polynomial time by computing all-pairs shortest paths in $\mathcal{T}(S)$ and $\mathcal{T}$.

The problem is proven to be NP-hard by reducing from the Hamiltonian path problem [17] defined as: given an undirected graph $G = (V, E)$ and vertices $v_1, v_2 \in V$, determine whether there is a Hamiltonian path from v_1 to v_2 in G. For a graph $G = (V, E)$ and vertices v_1, v_2, we produce the following instance of the admissibility problem: the topology $\mathcal{T} = \langle V, E', L \rangle$ where $E' = E \cup \{\{v_1, v_2\}\}$, and $L(u) = u$ if $u = v_1$ or $u = v_2$, and $L(v) = \perp$ for all $v \neq v_1, v_2$; the trace set $S = \{v_1 \perp_1 \cdots, \perp_{|V|-2} v_2, v_1 v_2\}$. In other words, S contains a trace of length $|V| - 1$ from v_1 to v_2 going only through anonymous nodes and a trace of length 1. It can then be shown that $\mathcal{T} \models \mathcal{S}$ iff there is a Hamiltonian path from v_1 to v_2 in graph G. ∎

III. Merging Anonymous Nodes

The topology inference problem, as specified, requires searching for an arbitrary topology that is minimum and admissible; by Lemma 2.8, even checking the latter condition is intractable. To permit feasible analysis of this problem, the search space of all topologies needs to be restricted. In this section, we show that it suffices to consider topologies of a special form: those that arise by partitioning the discovered anonymous routers and merging each partition in the topology induced by the probe result.

Partitions are most easily expressed using equivalence relations. Recall that an equivalence relation R is one that is reflexive, symmetric, and transitive. The equivalence class of an element x, denoted $[x]_R$, is defined to be the set

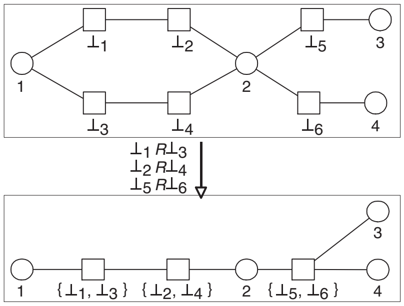

Fig. 5. Quotient Topology

$\{y \mid xRy\}$. The equivalence classes provide a partition of the underlying set. Let $\mathcal{T} = \langle V, E, L\rangle$ be a topology. We say that an equivalence relation on V is *label-respecting* if $\forall x, y,\ xRy \Rightarrow L(x) = L(y)$. Because of the conditions on the labeling function (Definition 2.1), if R does not relate distinct known nodes or a known node and an anonymous node, then it is label-respecting. The result of merging nodes of some topology can be formalized as the "quotient" topology with respect to the equivalence relation (the merging relation).

Definition 3.1 [Quotient Topology] Let $\mathcal{T} = \langle V, E, L\rangle$ be a topology and R a label-respecting equivalence relation on V. The quotient topology $\mathcal{T}/R = \langle V/R, E/R, L/R\rangle$ where $V/R = \{[u]_R \mid u \in V\}$, $E/R = \{\{[u]_R, [v]_R\} \mid \{u, v\} \in E\}$, and $L/R([u]_R) = L(u)$.

The quotient topology T/R is obtained from T by merging nodes that are related by R into a single node whose neighbor-set is the union of the neighbor-sets of all the nodes that have been merged. The label-respecting condition ensures that L/R is well-defined, *i.e.*, if $[u]_R = [v]_R$ then $L/R([u]_R) = L/R([v]_R)$. Fig. 5 gives an example of the quotient construction.

We are interested in quotients of the induced topology $T(S)$ (Definition 2.3) that are admissible for the probe result S. Admissibility requires trace preservation and distance preservation, which are reflected as corresponding conditions on the merging relation.

Definition 3.2 [Admissible Equivalence Relation] Let S be a probe result. A label-respecting equivalence relation R on $\mathcal{T}(S)$ is *admissible* (with respect to S) iff

- Trace Preservation: If uRv then there is no path $\sigma_1 u \sigma_2 v \sigma_3 \in S$
- Distance Preservation: If uRv then for any $x, y \in K(S)$, we have that $d_S(x, u) + d_S(v, y) \geq d_S(x, y)$

We next develop the key result that for a probe result S, any admissible topology contains a fragment (subtopology) that is structurally identical (isomorphic) to a quotient topology $T(S)/R$ with R being an admissible equivalence relation. To state this result, we need some definitions. A *homomorphism* $h : \langle V_1, E_1, L_1\rangle \rightarrow \langle V_2, E_2, L_2\rangle$ is a function $h : V_1 \rightarrow V_2$ such that for any $u \in V_1$, $L_2(h(u)) = L_1(u)$ and for any $\{u, v\} \in E_1$ we have that $\{h(u), h(v)\} \in E_2$. If h is a bijective function, and for every edge $\{u_2, v_2\} \in E_2$ there exists an edge $\{u_1, v_1\} \in E_1$ with $h(u_1) = u_2, h(v_1) = v_2$, then h is an *isomorphism* and two topologies are *isomorphic* if there exists an isomorphism between them. A *subtopology* of a topology $\langle V, E, L\rangle$ is a topology $\langle V', E', L'\rangle$ such that $V' \subseteq V, E' \subseteq E$ and $L'(u) = L(u)$ for any $u \in V'$.

We first prove the following lemma which shows that there is a homomorphism from the induced topology to any admissible topology, with the homomorphism satisfying additional conditions.

Lemma 3.3: Suppose that a topology $\mathcal{T} \models \mathcal{S}$. Then there exists a homomorphism $h : \mathcal{T}(S) \rightarrow \mathcal{T}$ such that:

1) For any path $\sigma_1 u \sigma_2 v \sigma_3 \in S$, we have that $h(u) \neq h(v)$
2) For any $u, v \in V(S)$, we have that $d_S(u, v) \geq d_{\mathcal{T}}(h(u), h(v))$.

Proof: Let $T = \langle V, E, L\rangle$. Consider any path $u_1 \ldots u_n \in S$. Since $T \models^t S$, there is a route $v_1 \ldots v_n$ in T whose trace $L(v_1) \ldots L(v_n) = \mathit{Erase}(p)$. We take $h(u_i) = v_i$ for $1 \leq i \leq n$. The function h is well-defined on $A(S)$ because each $\bot_k \in A(S)$ occurs in exactly one path $p \in S$ (Definition 2.2). The function h is well-defined on any $u \in K(S)$ because $L(h(u)) = \mathit{Erase}(u) = u \neq \bot$ which, by the condition on the labeling functions of topologies (Definition 2.1), means that $h(u)$ is unique. We next show that h is a homomorphism. By definition of h, $\mathit{Erase}(u) = L(u)$. Consider any edge $\{u, v\} \in E(S)$. Then there is a path $\sigma_1 u v \sigma_2 \in S$. By definition of h, the sequence $h(\sigma_1)h(u)h(v)h(\sigma_2)$ is a path in T which gives us that $\{h(u), h(v)\} \in E$. To establish Condition 1., consider any path $\sigma_1 u \sigma_2 v \sigma_3 \in S$. Then $h(\sigma_1)h(u)h(\sigma_2)h(v)h(\sigma_3)$ is a route, *i.e.*, a simple path, and hence $h(u) \neq h(v)$. Finally, Condition 2. follows from h being a homomorphism. ■

Using the special properties of the homomorphism constructed in Lemma 3.3, we can construct an admissible merging relation (the classical algebraic kernel of the homomorphism) which quotients the induced topology into an isomorphic image of a subtopology.

Corollary 3.4: Suppose that a topology $\mathcal{T} \models \mathcal{S}$. Then T has a subtopology T' such that $T' \models S$ and T' is isomorphic to a topology $T(S)/R$ for some equivalence relation R that is admissible with respect to S.

Proof: Let $h : T(S) \rightarrow T$ be the homomorphism provided by Lemma 3.3. Define the equivalence relation R on $V(S)$ by uRv iff $h(u) = h(v)$. Since h is a homomorphism, R is label-preserving. By Condition 1. of Lemma 3.3, R meets the trace preservation requirement of admissibility. Now, consider any u, v such that uRv and any $x, y \in K(S)$. By Condition 2., $d_S(x, u) \geq d_{\mathcal{T}}(h(x), h(u))$, *i.e.*, $d_S(x, u) \geq d_{\mathcal{T}}(x, h(u))$ (since $L(h(x)) = x$. Similarly, $d_S(v, y) \geq d_{\mathcal{T}}(h(v), y)$. Since $h(u) = h(v)$ (by uRv), $d_{\mathcal{T}}(x, h(u)) + d_{\mathcal{T}}(h(v), y) \geq d_{\mathcal{T}}(x, y) \geq d_S(x, y)$ (with the last inequality because $\mathcal{T} \models \mathcal{S}$). We thus obtain that R meets the distance preservation condition. Finally, take the subtopology $\mathcal{T}' = \langle V', E', L'\rangle$ defined by $V' = \{h(u) \mid u \in V(S)\}$ and $E' = \{\{h(u), h(v)\} \mid \{u, v\} \in E(S)\}$. Then $h^* : V(S)/R \rightarrow V'$ defined by $h^*([x]_R) = h(x)$ gives an isomorphism from $\mathcal{T}(S)/R$ to the subtopology $\mathcal{T}'$ of $\mathcal{T}$. ■

Corollary 3.4 provides a characterization of admissible topologies. It shows that for a topology to be the underlying network topology, it must contain a subtopology (intuitively, the probed fragment) with structure identical to $T(S)/R$ for some admissible equivalence relation R. Thus, the admissibility of an equivalence relation provides the formal systematic criterion for when two anonymous nodes can be identified as being the same router.

The following lemma, required in Sections IV and V, establishes sufficient conditions for $T(S)/R \models S$. The first part states that if R satisfies the trace preservation condition (of Definition 3.2) then $T(S)/R$ preserves the observed traces. The second part states that if the probe result does not include any paths with two consecutive anonymous nodes then any admissible merging relation R yields an admissible topology. The example of Section V-B shows that the condition on the probe result is necessary.

Lemma 3.5: 1) Let S be a probe result. For any label-preserving equivalence relation R on $T(S)$ that satisfies the trace preservation condition, we have that $T(S)/R \models^t S$.

2) Suppose that S is a probe result such that for any $\bot_k \in V(S)$ if $\sigma u \bot_k v \tau \in S$ then $Erase(u) = u, Erase(v) = v$. Then for any label-preserving equivalence relation R that is admissible, we have that $\mathcal{T}(S)/R \models S$.

IV. Intractability of Topology Inference

We first show that the problem *TOP-INF* is NP-complete. This requires us to consider the decision version of the optimization problem *TOP-INF*.

Definition 4.1 *TOP-INF-DEC*: Given a probe result S and a number k, is there a topology $T \models S$ with $|A(T)| \leq k$?

Theorem 4.2: The problem *TOP-INF-DEC* is NP-complete.

Proof: The problem is in NP because given a probe result S, we guess a relation R on $A(S)$ which requires a polynomial number of guesses (for each pair $u, v \in A(S)$ guess whether to relate them or not.) and extend R to be the identity relation on $K(S)$. We can then check in polynomial-time that: (a) R is an equivalence relation with at most k equivalence classes on $A(S)$, (b) R satisfies the path preservation condition, which by Lemma 3.5 ensures that $T(S)/R \models^t S$, and (c) for any $u, v \in K(S)$, $d_{T(S)/R}(u, v) \leq d_{T(S)}(u, v)$.

The problem is shown to be NP-hard by reducing the coloring problem to it. An instance of the coloring problem is an undirected graph $G = \langle V, E\rangle$ and an integer k with the problem being to determine whether there exists a k-coloring of G. For a graph $G = \langle V, E\rangle$, we define the probe result $f(G)$ as

$$f(G) \;=\; \{o \bot_u u \mid u \in V\} \cup \{u \; k_{\{u,v\}} \; v \mid \{u, v\} \notin E\}$$

where $o, u, v, k_{\{u,v\}}$ are known addresses. Given an instance G, k of the coloring problem, we produce the instance $f(G), k$. We now show that G has a k-coloring iff $f(G)$ admits a topology with at most k anonymous nodes. Assume that $f(G)$ admits a topology with at most k anonymous nodes. By Corollary 3.4, there exists an admissible equivalence relation R on the set $\{\bot_u \mid u \in V\}$ with at most k distinct equivalence classes (each of the non-anonymous nodes forms a singleton equivalence class in a label-preserving equivalence relation). Consider the coloring which assigns nodes u, v the same color iff $\bot_u R \bot_v$. This clearly uses at most k colors; all that remains to be shown is that this is a proper coloring. Note that $d_{f(G)}(\bot_u, u) = 1$, and $d_{f(G)}(u, v) \leq 2$ iff $\{u, v\} \notin E$. Since R is admissible, we have that if $\bot_u R \bot_v$ then $d_{f(G)}(u, v) \leq d_{f(G)}(u, \bot_u) + d_{f(G)}(\bot_v, v) = 2$ which in turn implies that $\{u, v\} \notin E$. This proves that any two nodes that are colored the same are not adjacent in the graph G and that this is a valid coloring.

For the converse direction, assume that G has a k-coloring given by the function $c : V \rightarrow \{1, \ldots, k\}$ such that for any $\{u, v\} \in E$ we have that $c(u) \neq c(v)$. Define the equivalence relation R on the topology $\mathcal{T}(f(G))$ by $\bot_u R \bot_v$ iff $c(u) = c(v)$, and xRy iff $x = y$ for $x, y \in K(f(G))$. Then R has at most k equivalence classes on the anonymous nodes. R trivially meets the path-preservation condition and meets the distance preservation condition because if $\bot_u R \bot_v$ then $\{u, v\} \notin E$ (because c is a proper coloring) and thus $d_{f(G)}(u, v) = 2$. Hence R is admissible, and by Lemma 3.5, $\mathcal{T}(f(G))/R \models f(G)$ and $\mathcal{T}(f(G))/R$ has at most k anonymous nodes. ∎

Given that the decision problem *TOP-INF-DEC* is NP-complete, the topology inference problem *TOP-INF* is intractable. Instead of looking for a topology that is the exact minimum, the best we can hope for is to infer a topology that is close to the minimum, or in other words, an approximation algorithm for *TOP-INF*. In the approximation version of the problem, given a probe result S, we try to find a topology $\mathcal{T}$ such that the ratio $|A(T)|/\alpha(S)$ is within some approximation bound ϵ that is as small as possible. Unfortunately, we can show that achieving an approximation bound of $\epsilon = n^{\delta}$, for some fixed constant δ, is also NP-hard. In other words, *TOP-INF* falls into the hardest class of NP-complete problems (*c.f.* [18]) with respect to approximability.

Theorem 4.3: There exists a polynomial-time reduction τ from the satisfiability problem, SAT, to *TOP-INF* that for some fixed $\delta > 0$ ensures that for all instances I of SAT

$$\begin{array}{rcl} I \in SAT & \Longrightarrow & \alpha(\tau(I)) \leq K(|I|) \\ I \notin SAT & \Longrightarrow & \alpha(\tau(I)) > K(|I|)n^{\delta} \end{array}$$

where $K(|I|)$ is a polynomial-time (in $|I|$) computable function, and n is the size of $\tau(I)$. Therefore, there exists a fixed $\delta > 0$ such that approximating *TOP-INF* to within a factor n^{δ} is NP-hard, where n is the size of the input probe result.

Proof: Consider the function f defined in the reduction from coloring to *TOP-INF-DEC*. As proven in Theorem 4.2, we have that for any graph G, $\alpha(f(G)) = \chi(G)$, where $\chi(G)$, the chromatic number of G, is the minimum number of colors required to color G. We thus have that for any values c, ρ, and for any graph G,

$$\begin{array}{rcl} \chi(G) \leq c & \Longrightarrow & \alpha(f(G)) \leq c \\ \chi(G) > c\rho & \Longrightarrow & \alpha(f(G)) > c\rho \end{array}$$

By the inapproximability result of coloring [18], we have a polynomial-time reduction σ from SAT to coloring that for some fixed $\epsilon > 0$ ensures that for all instances I,

$$\begin{array}{lll} I \in SAT & \Longrightarrow & \chi(\sigma(I)) \leq K'(|I|) \\ I \notin SAT & \Longrightarrow & \chi(\sigma(I)) > K'(|I|)n^{\epsilon} \end{array}$$

where $K'(|I|)$ is a polynomial-time (in $|I|$) computable function, and n is the size of $\sigma(I)$. We take the desired reduction τ of the theorem to be the composition $f \circ \sigma$. Note that $f(G)$ for any graph G has at most $O(|G|^2)$ paths each of length 2 and thus $|f(G)| = O(|G|^2)$. Thus, taking $\delta = \epsilon/2$, and $K = K'$, we get the statement of the theorem. ■

V. Pruning and Heuristics

The results of Section IV show that both exact solutions and approximate solutions, with worst case error bounds, are unlikely to be achieved with polynomial-time algorithms. We therefore look for heuristics that, on many input instances, produce topologies that are close to the actual topology. Corollary 3.4 shows that the topology inference problem can be solved by deriving an admissible equivalence relation R with fewest equivalence classes such that $T(S)/R \models S$. Our heuristics are based on this idea.

The heuristics use time and space that is polynomial in the number of anonymous nodes in $T(S)$, which could still be large when applied to probe results obtained from large networks. To keep computation times and memory footprints feasible, it becomes critical to reduce the input size. Our implementation therefore includes an initial pruning phase where the number of anonymous nodes in $T(S)$ is significantly reduced before the heuristics are applied. Section V-A describes the pruning algorithm and establishes its correctness. Section V-B develops the heuristics.

A. Pruning

One simple strategy for pruning is that if a path p in the probe result is a subpath of some other path, then p can be safely removed. The anonymous nodes appearing in p then no longer have to be considered for merging. Our pruning method is a generalization of this idea, where instead of requiring a path to be an exact subpath (for it to be removed), it only requires that all the purely anonymous segments, without considering their direction, can be recovered in another path.

Call a trace $p \in I\{\perp_i \mid i \in N\}^+ I$, *purely anonymous*, *i.e.*, one in which source and destination are known interfaces, all intermediate nodes are anonymous, and there is at least one intermediate anonymous node. For a trace $p \in I(I \cup \{\perp_i \mid i \in N\})^* I$ in the probe result, define its purely anonymous segment traces, *ATrace*(p), as the set:

$$\mathit{Erase}(\{p' \in I\{\perp_i \mid i \in N\}^+ I \mid \quad p = \sigma p' \tau \text{ for some } \sigma, \tau\})$$

For example, *ATrace*$(1\perp_1 2\perp_2\perp_3 34\perp_4 56)$ is the set $\{1\perp 2, 2\perp\perp 3, 4\perp 5\}$. For a sequence σ, define its reversal σ^R, by $(u_1 \ldots u_n)^R = u_n \ldots u_1$, and for a set of sequences, $S^R = \{p^R \mid p \in S\}$. We generalize the notion of subpaths,

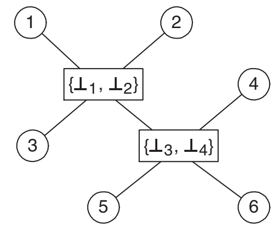

Fig. 7. Inadmissibility due to simultaneous merging

and define a path p to be *subsumed* by a path p', written $p \lessdot p'$ iff *ATrace*$(p) \subseteq$ *ATrace*$(p') \cup ($*ATrace*$(p'))^R$. For example, the path $2\perp_1 345\perp_2 6 \lessdot 13\perp_3 26\perp_4 54$. For a probe result S, define the result of its pruning $S^\sharp$, to be a minimal subset of S such that for every $p \in S$, there exists a $p^\sharp \in S^\sharp$ such that $p \lessdot p^\sharp$. For a topology $T = \langle V, E, L\rangle$, define the subtopology induced by a set $V' \subseteq V$, as the topology $T_{|V'} = \langle V', E', L'\rangle$, where $E' = \{\{u, v\} \in E \mid u, v \in V'\}$ and $L'(v) = L(v)$ for any $v \in V'$. For any probe result S, the *pruned topology* $T(S)^\sharp = T(S)_{|K(S)\cup A(S^\sharp)}$. Note that the pruned topology does not include any anonymous nodes in traces that are subsumed, but does include any links between known interfaces that are present in them.

We can then establish the following lemma, whose proof follows similar ideas as Corollary 3.4.

Lemma 5.1: Suppose that a topology $\mathcal{T} \models \mathcal{S}$. Then T has a subtopology T' such that $T' \models S$ and T' is isomorphic to a topology $T(S)^\sharp/R$ for some equivalence relation R on $T(S)^\sharp$ that is admissible with respect to S.

Lemma 5.1 shows that for a probe result S, it suffices to consider quotients of the pruned topology $T(S)^\sharp$ (as opposed to the full induced topology $T(S)$). Our heuristics are therefore applied to the pruned path set $S^\sharp$ and pruned topology $T(S)^\sharp$. Pruning is a generalization of bisimilarity; for the probe results of the network of Fig. 1, the pruned topology is that of Fig. 3. In our implementation, the pruned path set $S^\sharp$ is obtained by transforming each path to eliminate consecutively occurring known interfaces, re-ordering path segments so that the source and destination interfaces are in increasing order and then eliminating any paths that are subpaths of existing paths after the transformation.

B. Heuristics

For a given probe result S, we try to infer a quotient topology $T(S)/R$ for a merging relation R having the fewest number of equivalence classes. As Corollary 3.4 shows, the relation R has to be admissible with respect to S — one obvious methodology would therefore be to look for a merging relation satisfying the conditions given by Definition 3.2. Unfortunately the following example shows that not every admissible relation results in an admissible quotient topology.

Example: Let S be a probe result that includes the paths $1\perp_1 2, 2\perp_2\perp_3 4, 5\perp_4 6$, and additional paths consisting only of known nodes such that the shortest distances between the

```
Input:  A probe result S
Output: An equivalence relation R such that T(S)/R ⊨ S
Steps:
R_0 := {(x,x) | x ∈ V(S)};  /* no nodes are initially merged */
i := 0;
UP := A(S);  /* set of unpartitioned anonymous nodes */
while UP ≠ ∅ do
   P_i := ∅;  /* next equivalence class */
   C := UP;  /* candidate nodes to be included in equivalence class */
   while C ≠ ∅ do
      Pick u ∈ C; P_i := P_i ∪ {u};  /* add a candidate node */
      C := C − {u} − {v | u U(R_i) v};  /* remove nodes unmergeable with u */
   end while
   R_{i+1} := R_i ◁ P_i;  /* add new equivalence class to R_i */
   UP := UP - P_i; i := i + 1;
end while
R := R_i;
```

Fig. 6.   Algorithm for topology inference

nodes are: $d_S(1,3) = d_S(4,6) = 2, d_S(1,4) = d_S(2,4) = d_S(3,6) = d_S(3,5) = 3, d_S(1,6) = 4$. Then an equivalence relation R such that $\bot_1 R \bot_2$ and $\bot_3 R \bot_4$ (with other related elements implied by reflexivity and symmetry) is admissible with respect to S. However, consider $T(S)/R$, part of which is shown in Fig. 7. Although R is admissible, we do not have that $T(S)/R \models S$ because $d_{T(S)/R}(1,6) = 3 < d_S(1,6)$ which violates the distance preservation criterion.

The reason for this example is that merging $\bot_1, \bot_2$ did not violate the distance preservation criteria only in the absence of any other anonymous nodes being merged, and similarly for $\bot_3, \bot_4$. Once both the pairs were merged simultaneously, the resultant topology was not admissible.

In general, merging many pairs of anonymous nodes simultaneously may produce an inadmissible topology. However, we show that more than one pair of nodes can be merged provided they belong to the same equivalence class. Our heuristics are therefore based on producing the merging relation incrementally, extending by one equivalence class at a time. The construction of each equivalence class takes into account nodes that have already been merged to ensure that the distance preservation requirement is met. (Lemma 3.5 ensures that path preservation is automatically ensured by any admissible merging relation.) We next make these intuitions more precise.

Let R be a label-respecting equivalence relation on $T(S)$ for some probe result S. Define the set $UP(R)$ (the nodes that are not partitioned by R) as the set $\{x \in A(S) \mid \ |[x]_R| = 1\}$. For a non-empty set $P \subseteq UP(R)$, we define the relation $R \lhd P$ (the extension of R with a new equivalence class P) as $x R \lhd P y$ iff $xRy \vee (x \in P \wedge y \in P)$, which can be seen to be a label-respecting equivalence relation. Note that for any $x \in P$, we have that $[x]_{R \lhd P} = P$. We can now define the conditions for extending a merging relation.

Definition 5.2 Let S be a probe result, and R a label-respecting equivalence relation on $\mathcal{T}(S)$. Then an equivalence relation R' on $UP(R)$ is an *admissible extension of* R (with respect to S) iff

- Trace Preservation: If $R'(u,v)$ then there is no path $\sigma_1 u \sigma_2 v \sigma_3 \in S$
- Distance Preservation: If $R'(u,v)$ then for any $x, y \in K(S)$, we have that $d_{T(S)/R}(x,u) + d_{T(S)/R}(v,y) \geq d_{T(S)/R}(x,y)$

Note that the main difference from Definition 3.2 is that distance preservation is computed with respect to the quotient topology $T(S)/R$ rather than the full topology $T(S)$, to take into account nodes that have already been merged (by R).

Lemma 5.3: Let S be a probe result and R a label-respecting equivalence relation such that $T(S)/R \models S$. Then for any equivalence relation R' that is an admissible extension of R, we have that $T(S)/(R \lhd [x]_{R'}) \models S$ for any $x \in UP(R)$.

Note that Lemma 5.3 only shows that an extension of the equivalence relation R with *one* equivalence class produces an admissible topology. As should be expected from our previous example, we do not have, in general, that $T(S)/(R \cup R') \models S$.

For presenting our topology inference algorithm, it is convenient to recast Definition 5.2 using an unmergeability relation. Let R be a label-respecting equivalence relation on the topology $T(S)$. We define the unmergeability relation (with respect to R), $U(R) \subseteq UP(R) \times UP(R)$, as $(u,v) \in U(R)$ iff either (a) there exists a path $\sigma_1 u \sigma_2 v \sigma_3 \in S$, or (b) there exists $x, y \in K(S)$ such that $d_{T(S)/R}(x,u) + d_{T(S)/R}(v,y) < d_{T(S)/R}(x,y)$. Note that R' is an admissible extension of R iff $R' \cap U(R) = \emptyset$, and that a set $P \subseteq UP(R)$ is an equivalence class of an admissible extension iff $\forall u, v \in P. (u,v) \notin U(R)$.

The general algorithm for topology inference is given in Fig. 6. For each set P_i constructed, we have that $(u,v) \notin U(R_i)$ for any $u, v \in P_i$, and P_i is therefore an equivalence class of an admissible extension of R_i. Using Lemma 5.3, we can show by induction that for each i, $T(S)/R_i \models S$, and that the equivalence relation produced results in an admissible quotient. Furthermore, note that each set P_i produced is maximal in that for any node $u \notin P_i$, the set $P_i \cup \{u\}$ is not an equivalence class of any admissible extension of R_i. Together with the fact that $U(R_i) \subseteq U(R_{i+1})$ this implies that the equivalence relation produced is a maximal one, *i.e.*, no additional anonymous nodes can be merged.

Theorem 5.4: For the equivalence relation R produced by the algorithm of Fig. 6, $T(S)/R \models S$ and for any $R' \supseteq R$

such that $T(S)/R' \models S$, we have that $R' = R$.

Different heuristics can be obtained from the basic algorithm of Fig. 6 by fixing a particular choice of $u \in C$ that is added to the currently generated equivalence class P_i (in Line 8 of the algorithm); we consider two possibilities. One choice is to pick a u such that $|\{v \in C \mid u\, U(R_i)\, v\}|$ is minimum among all nodes in C; we call the resulting algorithm the "Min" heuristic. The motivation for this heuristic is to make each equivalence class P_i as large as possible, thereby ensuring that the equivalence relation R has a small number of equivalence classes. It can be shown that the size of P_i is at least $\lfloor \log_k n \rfloor$, where $n = |UP|$ is the number of remaining unpartitioned nodes and k is the minimum number of equivalence classes in any admissible extension of R_i (The proof of this is similar to that in [19].) However, the Min heuristic may perform poorly, if for some intermediate merging relation R_i, the number of equivalence classes in any admissible extension of R_i is large, *i.e.*, the new unmergeability relation $U(R_i)$ is such that many previously mergeable pairs of nodes become unmergeable.

A second heuristic tries to ensure that the equivalence classes constructed do not significantly alter the mergeability of the remaining nodes. This heuristic, referred to as "Max", chooses a u such that $|\{v \in C \mid u\, U(R_i)\, v\}|$ is maximum among all nodes in C. The intuition is that such nodes are the most constrained as to which nodes they can be merged with and therefore choosing them earlier ensures flexibility in extending the merging relation. However, the size of each equivalence class produced would be small, which could increase the number of equivalence classes.

VI. Evaluation

The two heuristics, Min and Max, were evaluated on data collected via simulations and results from probing an IPv6 testbed. For simulation, ns-2 [20] was enhanced to support both source routing and the Atlas probe engine [13]. A set of connected topologies were randomly created according to specified numbers of known routers, anonymous routers and links. There was no route changes during the simulations. Probe results collected from these topologies by ns-2 were then processed by the heuristics. When both Min and Max were applied to the probe result obtained from the topology in Fig 1, the actual topology was correctly inferred.

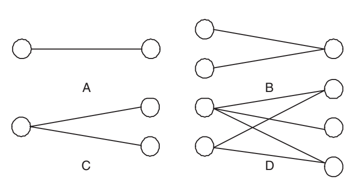

Fig. 8. Mapping from actual to inferred anonymous routers

A. Evaluation Method

Directly comparing the number of anonymous routers in actual and inferred topologies does not necessarily reflect the minimality of the inferred topology nor its similarity with the actual topology. One anonymous router in an actual topology may give rise to multiple anonymous routers in its inferred topology and vice-versa. As a result, the two topologies may have the same number of anonymous routers even though they differ significantly from each other.

We therefore use an approach based on mappings from anonymous routers in the probe result (*generated anonymous routers*) to actual and inferred anonymous routers. For each simulated topology, the mapping from generated to actual anonymous routers is produced by ns-2, whereas the mapping from generated onto inferred anonymous routers is produced by each heuristic. A graph $G = (X, Y, E)$ is produced based on the two mappings: X represents the set of actual anonymous routers, Y represents the set of inferred anonymous routers, and $\{x, y\} \in E$ iff there is at least one generated anonymous router that maps to both x and y, *i.e.*, it is obtained from probing x and got merged into y. It can be easily shown that G is bipartite with node partitions X and Y. There are only four possible cases for bipartite connected components of G, as illustrated in Fig. 8.

Case A, the one-to-one mapping, indicates all generated anonymous routers resulting from an actual router x are merged into a single inferred router y. As a result, all observed connectivity information for x is preserved by y. This case is therefore called the *perfect-merge*. Case B, the many-to-one mapping, means a set $M \subseteq X$ of actual anonymous routers are merged into a single inferred anonymous router y. In addition, y has the aggregated connectivity of all routers in M. This case, denoted as the *over-merge* case, indicates that the actual topology is not minimum. Case C, the one-to-many mapping, indicates that a single actual router and its connectivity is splitted into multiple inferred routers and therefore called the *under-merge* case. Case D, the many-to-many mapping, means generated anonymous routers, *i.e.*, connectivity information, belonging to multiple actual anonymous routers are intermingled into multiple inferred anonymous routers. In this case, if the component is $G' = (X', Y', E')$ and $|Y'| > |X'|$, then it is called the *mixed-non-optimal* case. Otherwise, it is called the *mixed-optimal* case.

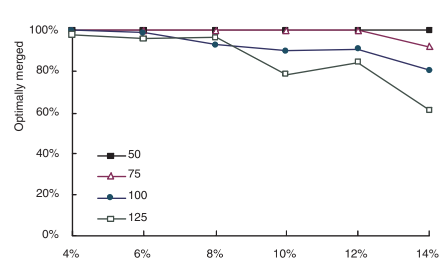

Fig. 9. Percentage of routers optimally merged by Min

B. Simulation Results

Topologies with 50 routers and 50, 75, 100, and 125 links were studied. Changing the number of nodes and links in proportion does not change the results significantly. Each link

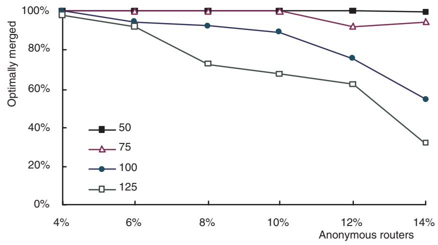

Fig. 10. Percentage of routers optimally merged by Max

was assigned the same cost and shortest path routing was used. Choosing link cost randomly between 1 and 10, thereby simulating the non-shortest path routing case, does not change the result significantly. According to our observation of the 6bone, it is likely that only a small percentage of routers are anonymous. Therefore, numbers of anonymous routers were selected to be from 2 to 7, corresponding to 4% to 14%. If there was only one anonymous router, both heuristics always achieved perfect merge and is therefore not listed here. For all the figures in this section, the x-axis represents the percentage of actual routers that are anonymous. Each data point on the graph is the average obtained from running the simulation on about 50 randomly generated topologies.

The probability that an actual anonymous router is merged optimally was first studied. Its upper bound can be calculated by first summing up the number of actual anonymous routers in perfect-merge, over-merge and mixed-optimal cases, and then dividing it by the total number of actual anonymous routers. The results obtained by Min and Max heuristics are shown in Fig. 9 and Fig. 10 respectively. In both figures, y-axis represents percentage of anonymous routers merged optimally, and different lines represent topologies with different number of links: 50, 75, 100 and 125. It can be observed that Min optimally merged more than 90% anonymous routers in most cases while Max did not do as well. By calculating the difference between numbers of anonymous routers in the actual and inferred topologies, we observed that less than 25% more anonymous routers were introduced by both heuristics in all cases considered in Fig. 9 and Fig. 10.

Since the inferred topology is not necessary the actual topology, it is also interesting to see how similar they are. This can be measured by the percentage of actual anonymous routers in perfect-merge and over-merge cases, where connectivity of actual anonymous routers are correctly reflected in the inferred topology. The results are listed in Fig. 11 and Fig. 12. Min achieved more than 80% node similarity when no more than 8% routers were anonymous, and Max performed much worse.

The performance of both Min and Max improved as average node degree and percentage of anonymous routers decreased. When there are few actual anonymous routers, our distance preservation criterion makes it unlikely for generated anonymous routers from different actual routers to be mixed up. Under such condition, our heuristics would produce a topology that is both close to minimum and similar to the actual topology. The same argument also applies when average node degree is small.

We observed that Max typically had worse performance than Min. The two heuristics differs in their strategies in selecting equivalent classes: Min tries to build a larger equivalence class in each round while Max tries to leave more flexibility for future rounds. It turned out, from our results, the former had bigger impact.

C. Experiment Results

Both heuristics were further tested on probe results collected from a small IPv6 testbed in our lab. Two of the routers were configured as anonymous. When the heuristics were applied to the probe result, the inferred topology is the same as the actual topology. Applying the Min heuristic on the probe result of the 6Bone produced 34 anonymous routers while the total number of routers was 1351, which means about 2.5% routers were anonymous [13]. According to simulation results, the inferred topology is highly similar to the actual topology and very close to the minimum under such conditions. Therefore, one may conjecture that the inferred result is similar to the actual topology of the 6Bone.

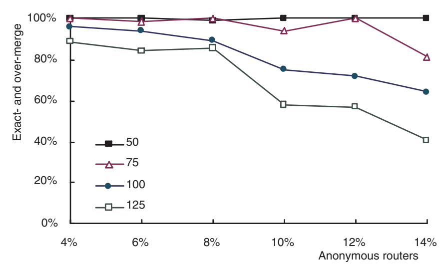

Fig. 11. Similarity between inferred and actual routers for Min

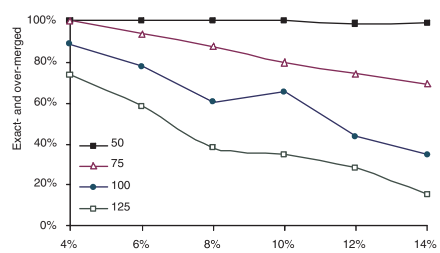

Fig. 12. Similarity inferred and actual routers for Max

VII. Related Work

Source routed traceroute had been used in IPv4 topology discovery tools to obtain router connectivity and to determine whether two interfaces belong to the same router [12]. By sending source routed traceroute to a known destination, Govindan and Tangmunarunkit [12] could identify whether a router was source routing capable. They recognized some intermediate IPv4 routers did not send back ICMP responses, which indicated that they might be anonymous. However, since the authors were interested only in router adjacencies, rather than paths, such routers were ignored. Pasiot and Grad [21] used source routing capable routers to analyze the topologies of multicast trees on the Internet. Any path that contained non-responsive routers were discarded from their analysis.

Some topology discovery tools use multiple hosts to send traceroute packets in order to detect comprehensive network topology. For example, the Skitter project [9], [16] sent traceroute packets from more than 20 hosts distributed around the world. The authors reported that close to one third of probed paths contain anonymous, private, or invalid addresses, *i.e.*, anonymous routers. To handle anonymous routers, arc graph, place holder graph, and shortcut place holder graph were introduced. But the focus was on extracting the topology of the Internet's core, rather than inferring actual anonymous routers in the topology. The Rocketfuel project [14] used results from 294 public traceroute servers to construct router level ISP topologies. Techniques that can reduce the amount of probing and resolve address aliasing were presented. But anonymous routers were not discussed. Cheswick et. al. [7] proposed using IP-in-IP tunnels to distribute probe packets so that probes could originate from multiple sources. In their tool, probing of a path was stopped if an intermediate router failed to respond; the anonymous router problem was not addressed. Paxson [8] ran traceroute among 37 different hosts in order to study end-to-end routing behavior in the Internet. The IDMaps [4] project studied the placement of tracers for distance estimation on the Internet. The focus of these two projects was on end-to-end performance rather than accurate topology, therefore, they did not address the anonymous router problem.

The only previous work to consider the anonymous router problem, due to Broido and Claffy [16], uses shortcut placeholder graphs in which adjacent anonymous nodes between two known nodes are discarded if there already exists a path of equal or shorter length between the two known nodes. This results in a topology that is essentially equivalent to that based on merging bisimilar nodes whose shortcomings have already been discussed in Section I. Compared to the bisimilarity approach, shortcut graphs may contain fewer anonymous nodes but this reduction also results in loss of path information from the probe result.

Our topology discovery tool, Atlas, effectively sends traceroute from every known routers by exploiting widespread support for source routing in IPv6 routers. It also continues to probe a path after determining an intermediate routers is anonymous. Therefore, a more comprehensive topology can be derived. As a side effect, more occurrences of anonymous routers are included in the probe result. Consequently, it is crucial for us to solve the anonymous routers problem.

VIII. Conclusions

This paper has identified the anonymous routers (those whose addresses could not be determined by traceroute) phenomenon and defined the topology inference problem in the presence of anonymous routers. We have shown that the problem is intractable, established lower bounds on its approximability, and developed heuristics and evaluated them through simulation and experimentation. Our simulation results show that our heuristics are able to produce topologies close to the actual topology under conditions similar to the 6Bone.

As part of future work, we would like to obtain an approximation algorithm that provably achieves this worst case bound. This paper made the simplifying assumption that a router cannot have both a known and anonymous interface address. While this assumption is consistent with our experience in the context of the 6Bone, it would be interesting to solve the more general problem of inferring topologies in which actual routers may behave anonymously in some paths and but not in others.

Although the impact of anonymous routers on topology inference was made evident by the Atlas project, the problem is also relevant in IPv4. Furthermore, we believe that the techniques developed in this paper can also be incorporated into existing tools to discover more comprehensive and accurate network topologies.

References

[1] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The Impact of Internet Policy and Topology on Delayed Routing Convergence," in *Proc. IEEE INFOCOM*, 2001.

[2] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 3, 1997.

[3] T.S. Ng and H. Zhang, "Predicting Internet Network Distance with Coordinates-based Approaches," in *Proc. IEEE INFOCOM*, 2002.

[4] P. Francis et. al., "IDMaps: A Global Internet Host Distance Estimation Service," *IEEE/ACM Transactions on Networking*, vol. 9, no. 5, pp. 525–540, Oct. 2001.

[5] H. Tangmunarunkit and R. Govindan and S. Shenker, "The Impact of Policy Routing on Internet Paths," in *Proc. of IEEE INFOCOM*, March 2001, pp. 736–742.

[6] V. Jacobson, "Traceroute Software," Lawrence Berkeley Laboratories, Dec. 1998.

[7] B. Cheswick, H. Burch, and S. Branigan, "Mapping and Visualizing the Internet," in *Proc. USENIX*, 2000.

[8] V. Paxson, "End-to-end Routing Behavior in the Internet," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, 1997.

[9] "Skitter," http://www.caida.org/tools/measurement/skitter/.

[10] P. Barford, A. Bestavros, J. Byers, and M. Crovella, "On the Marginal Utility of Deploying Measurement Infrastructure," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[11] W. Willinger H. Chang, S. Jamin, "Inferring AS-Level Internet Topology from Router-Level Path Traces ," in *Proceedings of SPIE International Symposium on Convergence of IT and Communication*, August 2001.

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

[13] D. Waddington, F. Chang, R. Viswanathan, and B. Yao, "Atlas: Discovering IPv6 Network Topology," submitted for publication.

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

[15] Robin Milner, *Communication and Concurrency*, Prentice Hall International Series in Computer Science. Prentice Hall, 1989.

[16] A. Broido and K. Claffy, "Internet Topology: Connectivity of IP graphs," in *Proceedings of SPIE International Symposium on Convergence of IT and Communication*, August 2001.

[17] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, McGraw Hill, 1994.

[18] S. Arora and C. Lund, "Hardness of Approximations," in *Approximation Algorithms for NP-Hard Problems*, D. S. Hochbaum, Ed. PWS, 1995.

[19] A. Wigderson, "Improving the Performance for Approximate Graph Coloring," *Journal of the ACM*, vol. 30, no. 4, 1983.

[20] "Network Simulator NS-2," http://www.isi.edu/nsnam/ns/.

[21] J. Pansiot and D. Grad, "On Routers and Multicast Trees in the Internet," *ACM Computer Communication Review*, Jan. 1998.