A Measurement Study on Overhead Distribution of Value-Added Internet Services ⋆

Mehmet H. Gunes *, Sevcan Bilir, Kamil Sarac

Dept of Computer Science, University of Texas at Dallas, Richardson, TX 75083

Turgay Korkmaz

Dept of Computer Science, University of Texas at San Antonio, San Antonio, TX 78240

Abstract

During the last two decades, several value-added services (e.g., IP multicast, IP traceback, etc.) have been proposed to extend the functional capabilities of the Internet. Due to the increasing role of these services, there is a need to better understand their impact on the network. In this paper, we present an experimental study on the intersection characteristics of end-to-end Internet paths and trees. We analyze these characteristics to understand the scale and the distribution of "*state overhead*" that is incurred on the routers by various value-added network services. For the reliability of our analysis, a representative, end-to-end router-level Internet map is essential. Although several maps are available, they are at best insufficient for our analysis. Therefore, in the first part of our work, we exert a measurement study and present an alternative approach to obtain an end-to-end router-level map conforming to our constraints. In the second part, we conduct various experiments using our map and shed some light on the scale and distribution of the state overhead of value-added Internet services in both unicast and multicast environments.

Key words: Value-added Internet services, State overhead, Multicast state elimination, Load distribution, Router-level maps, Topology mapping
PACS: 30.000, 50.100, 50.110

⋆ An abridged version of this paper was presented at the 13th IEEE International Conference on Network Protocols (ICNP '05), Nov 6-9, 2005, Boston, MA, USA.

* Corresponding author.
Email addresses: mgunes@utdallas.edu (Mehmet H. Gunes), sevcan@utdallas.edu (Sevcan Bilir), ksarac@utdallas.edu (Kamil Sarac), korkmaz@cs.utsa.edu (Turgay Korkmaz).

1 Introduction

There is a continuous demand for extending the functional capabilities of the Internet through supporting several *value-added* services (e.g., IP multicast [1]; QoS support [2,3]; IP traceback [4–7]; receiver controlled communication service [8–10]; etc.). Value-added services are intended to improve the scalability and efficiency of end user applications, and enhance the reliability and security of the network infrastructure. However, introducing new services without breaking or negatively affecting the existing ones requires to overcome several challenges. One of the key challenges is the lack of understanding the topological and functional characteristics of the Internet. This understanding is vital in developing appropriate mechanisms and provisioning resources so that the Internet can support value-added services in an efficient and scalable manner.

In this paper, we present an experimental study to analyze the scale and distribution of *state overhead* that is introduced by various value-added services on the routers. As an effort to reach representative conclusions, we use a realistic Internet topology map that we construct from a set of path traces that we collect from the Internet. We then use our topology map to conduct several experiments to study the intersection characteristics of end-to-end paths and trees. The results of this study help us better understand various issues for the deployment, operation, and management of value-added services.

This paper mainly consists of two related parts. In the first part, we present the details of a measurement study that we conducted to obtain a representative router-level topology map for our analysis. In this part, we emphasize the need for a characteristic-oriented topology sampling and present our methodology to do so. In the second part, we use our topology map to study the router-level intersection characteristics of end-to-end Internet paths and trees. We use these characteristics to understand the scale and distribution of state overhead incurred by value-added services on the routers.

It is a well-known fact that the accuracy of a network map has an important effect on the validity of the results obtained in a measurement study. Therefore, depending on the goals of the study, one should be careful in collecting topology data and constructing the corresponding topology map. Several maps have been collected as part of other research efforts considering various characteristics of the Internet (see Section 2.1). However, as we discuss later in detail, these maps are at best insufficient for analyzing intersection characteristics of the Internet paths and trees as they are not either end-to-end or router-level. Moreover, the existing maps are collected without being concerned about the specific topological characteristics that we are interested in. Therefore, one of the major tasks in our study is to collect and process topology data, and then verify the representativeness of the resulting network map for the topological

characteristic that we study.

At this end, we use the traceroute utility [11] to collect topology data and process it to build a network map. The main distinguishing characteristic of our topology is that it includes *end-to-end* path traces among a relatively large number of vantage points (a total of 153 nodes) that are carefully selected from the *periphery* of the Internet in North America (see Section 3.2 for more details). After collecting the topology data, the main task is to process this data to build an *accurate* network map that corresponds to the Internet topology among the vantage points. This task involves identifying the set of IP addresses belonging to the same router (i.e., an operation referred to as *IP alias resolution* [12]) and handling unresponsive routers that are represented by a '*' in traceroute output. After pointing out the limitations of the existing IP alias resolution approaches, we propose a more effective mechanism for alias resolution. Later on, in Section 5, we demonstrate the impact of imperfect alias resolution on the results presented in this paper. In addition to alias resolution, we address the issue of *unresponsive routers* that do not respond to traceroute queries. We present a methodology to identify individual unresponsive routers across different traceroute outputs and represent each one with a single address in the resulting network map. Finally, we present an experimental study to demonstrate the representativeness of our topology map for the topological characteristic that we are interested in.

Using our topology map, we perform various experiments to study the router-level intersection characteristics of end-to-end Internet paths (for unicast applications) and trees (for multicast applications). We use these characteristics to better understand the scale and the place of "*state overhead*" that is incurred on the routers by value-added network services, e.g., IP multicast [1], IntServ [2], DiffServ [3], IP traceback [4–7], and recently proposed receiver controlled communication service primitives [8–10]. More specifically, we are seeking answers to various questions such as "Does the state overhead follow any known distribution?", "How is the overhead distributed at the backbone?", "Is there any relation between the overhead incurred on and the location (e.g., edge, border, backbone, etc.) of the routers in the network?" Answering these questions is essential to developing appropriate mechanisms and provisioning resources so that the Internet can support aforementioned value-added services in an efficient and scalable manner.

Our findings in this direction shed some light on various issues related to deployment, operation, management, and performance of value-added services. For example, we observe that the load distribution on routers follow a *heavy tailed distribution* where a small number of routers experience heavy load while a large number of routers experience lighter load. For unicast applications, even though the most heavily loaded routers are backbone routers, the average load on border and exchange point routers can get as high as

backbone routers. In sparse mode multicast applications, most of the load accumulates at backbone routers. As multicast groups get denser, the overhead on exchange point routers reaches to that of backbone routers. Finally, our experiments show that the previously proposed approaches on multicast state reduction are not effective in reducing the number of forwarding states at heavily loaded branching routers, which, most of the time, correspond to border and exchange point routers. As a result, an important conclusion from the second part of our study reveals the fact that the techniques that mainly focus on reducing the load at the core of the network, e.g., DiffServ [3] and Aggregated Multicast [13], may not always be sufficient in improving the efficiency and scalability of value-added services. Such efforts should also help reduce the load incurred on border and exchange point routers by the value-added services. We present the details of our analysis in Section 4.

In summary, the contributions of this paper are threefold: (1) a study that points out the need for characteristic-oriented topology sampling, (2) a detailed analysis of the key steps in obtaining a topology map for Internet measurement studies, and (3) an experimental study on the intersection characteristics of end-to-end Internet paths and trees. After presenting related work in Section 2, we discuss the details of our two contributions in Sections 3 and 4, respectively. Section 5 presents additional results on the impact of imperfect IP alias resolution on the results presented in this paper. Finally, Section 6 concludes the paper.

2 Related Work

Our work has two major steps: (1) constructing a router-level Internet map reflecting the overhead distribution characteristic that we are interested in, and (2) characterizing the state overhead (simply **load**) incurred by value-added Internet services on the routers. Hence, we discuss related work in two parts.

2.1 Router-Level Internet Measurements

There has been a large body of work related to Internet topology measurements. Earlier work examined routing and end-to-end path characteristics (including loss and jitter characteristics) of the Internet [14–18]. More recently, researchers have studied the connectivity characteristics of the Internet topology. One interesting recent finding was that the degree distribution of the nodes in the Internet follows a heavy tailed distribution. In their landmark work [19], Faloutsos et al. used Autonomous Systems (AS) and router

level Internet topologies to show that power laws can be used to characterize the degree distribution of the nodes in the Internet. Later on, Broido and Claffy [20] used around 220M traceroute data (collected by the Skitter tool that we discuss in the next section) to construct a router-level Internet map and used that map to study the connectivity characteristics of the Internet. They showed that Weibull distribution can be used to approximate the out-degree distribution of the routers.

The observations in [19] have generated a significant debate on whether the node degree distribution can be modeled by power laws or not. During this debate, researchers questioned many aspects of the methodology that is currently used in Internet measurements studies: while some pointed out the marginal utility of using additional vantage points in topology collection [21], several others stressed the necessity of increasing the number of the vantage points [22–24]; some discussed the difficulties of inferring the topological attributes from the collected data [25,26]; some pointed out the potential of sampling biases in topology collection [27]; and some questioned the validity of using degree distribution as the only (or the main) metric to characterize the Internet topology [28]. Specifically, in [21], Barford et al. studied the utility of adding new vantage points in traceroute-based topology discovery measurements. They pointed out that adding new vantage points as traceroute sources have diminishing return, suggesting that it is more important to add destinations than sources. But, later on Lakhina et al. [27] identified that using (k,m)-traceroutes (i.e., traceroutes from k sources to m destinations where $k \ll m$) introduces sampling bias in topology measurements. They experimentally showed that using (k,m)-traceroute probes result in network topologies with degree distributions following the power laws whereas the degree distributions in the underlying original graphs do not necessarily follow the power laws. More recently, Li et al. [28] showed that the degree distribution by itself is not sufficient to properly represent the Internet topology.

Our work is related to the above studies in the sense that we use similar techniques in data collection and processing steps. One difference in terms of topology collection between our work and the previous work is that most topology collection studies aim at collecting very large topology maps without considering the topological characteristic that is examined in the measurement study. In our work, we collect a topology data specifically for studying the intersection characteristic of end-to-end paths and trees. Another difference, in terms of topology verification, between our work and the previous work [27] is that the latter develops a sampling bias test based on the degree distribution characteristic. In our work, we are interested in another topological characteristic (i.e., load distribution characteristic) and therefore develop an alternative verification approach that is more suitable for our study. Section 3.2 provides more discussion about our topology collection and representativeness verification efforts.

Starting from early 1990s, several value-added services have been proposed or introduced into the Internet. They include IP multicast [1]; IP-based QoS support such as IntServ [2] and DiffServ [3]; packet marking and/or logging for IP traceback [4–7]; and recent proposals on receiver-controlled communication services such as p2cast [8], SIFF [9], and TVA [10]. We now discuss these services under two classes, namely, IP multicast and unicast.

IP Multicast

IP multicast [1] is one of the first value-added network services that is developed and partially deployed in the Internet. In IP multicast, source data propagates on a multicast distribution tree toward the receivers. Each router on a multicast tree maintains group specific *forwarding state*. As the number of multicast groups increase, the state overhead in the network increases. To reduce this overhead, researchers have proposed several state-reduction approaches that can be divided into two groups: (1) state aggregation and (2) tunneling approaches.

The main idea in *state aggregation* is to combine multiple multicast forwarding state entries into one single entry. This is possible when multiple entries have adjacent group addresses, same incoming, and same (in the case of perfect aggregation) or similar (in the case of leaky aggregation) outgoing interfaces at a router [29,30]. *Tunneling* proposals focus on reducing the number of multicast states by using unicast- or multicast-based tunnels [13,31–33]. As an example, Aggregated Multicast [13] uses domain local multicast tunnels to carry data from multiple multicast groups in a transit domain. This way, the number of states kept at internal routers reduces from the number of global multicast trees to the number of local (or tunnel) multicast trees.

Although much work has been done on state-reduction techniques, little has been done on understanding the characteristics of *multicast state overhead*. In this end, Wong and Katz conducted an experimental study to analyze the nature of the multicast state scalability problem in [34]. By using several AS level Internet maps, a snapshot of Multicast Backbone (MBone) overlay network, and a set of synthetic router level topologies, they provided a comprehensive analysis of state scalability problem. They also studied the effect of non-branching state elimination on the state scalability problem and showed that non-branching state elimination helps reduce the state load considerably.

In Section 4.1, we focus on multicast *state scalability* problem and extensively investigate the effectiveness of the existing state-reduction techniques using our router-level Internet map. We find that state scalability at the AS level is different than that at the router level. To better reflect the router-level state

distribution or state accumulation characteristics, it is essential to use representative router-level maps, as done in our analysis. AS-level maps or synthetic topologies are of limited use in understanding router-level state distribution characteristics of multicast services in the Internet.

Unicast

In addition to IP multicast, several value-added network services have been proposed for unicast environments. As discussed in the forthcoming paragraphs, researchers have informally discussed the scalability problems associated with the unicast value-added services. However, to the best of our knowledge, there is no work that experimentally evaluates the load incurred by value-added unicast services. In this end, our analysis in Section 4.2 is the first.

As part of the defense with the recent surge in network-based security attacks, several approaches have been proposed for tracing attack packets back to their origins. *Traceback* approaches are divided into two main groups: packet marking and packet logging. In packet marking approaches [4,6,7,35,36], routers mark packets in transit. In logging-based approaches [5], routers log a signature of the forwarded packets in a reserved local storage space. Both traceback approaches above introduce additional overhead (e.g., processing or storage overhead) on the routers. Depending on the underlying network topology, routing decisions, and traceback deployment strategies, the distribution of this overhead may show different characteristics. In the best case, the overhead introduced into the network is equally shared by all the routers. This avoids potential hot spots or bottlenecks. In the worst case, however, there may be a significant load imbalance among the routers deteriorating the overall performance of the network.

Similarly, as part of the defense against cyber attacks in the Internet, several researchers have proposed *receiver controlled communication* as an alternative communication model. These include p2cast [8], SIFF [9], and TVA [10]. The first two approaches require routers to maintain additional state information for each end-to-end flow and the last one requires routers to perform additional computation to generate a signature and store it on the packets. As a result, these services incur additional overhead on the routers and therefore cause similar concerns as in the above mentioned IP traceback approaches.

As part of providing QoS support to emerging real-time and multimedia applications (e.g., VoIP, VoD, IPTV, Teleconference), researchers have proposed new services, including the Integrated Services (IntServ) [2] and the Differentiated Services (DiffServ) [3]. In general, *IntServ* requires establishing a connection that can meet the given QoS requirements. It is considered to be an unscalable solution due to excessive amount of state overhead that might be

incurred particularly on the core routers. In order to reduce the overhead at the core, *DiffServ* is proposed as a more scalable solution as it does not require to maintain state information in the core. However, it still needs to maintain some state information at the edge routers [37]. Edge routers need to consider individual flows through traffic classification and traffic conditioning so that individual flows will enter the network in a controlled manner.

3 Building Representative Internet Maps

In this section, we first demonstrate the need for a representative topology collection approach for the measurement study at hand. Then we describe our methodology in collecting and processing a new router-level Internet map that we use in studying the overhead distribution characteristic in Section 4. Finally, we present an experimental evaluation on the appropriateness of the methodology that we use in our data collection procedure.

3.1 Characteristic-Oriented Topology Sampling

One fundamental goal in topology measurement studies is to understand the topological characteristics of the network in question. Clearly, one can directly analyze the whole network topology and make definitive conclusions about its characteristics *provided that* the network topology is known or easy to collect. However, in the case of the Internet, it will be too costly to consider the whole topology or impossible to attain the underlying network completely. Due to this practical limitation, we use *sampling* to collect a sample topology map and analyze the characteristics of this sample map to make *statistical inferences* about the characteristics of the Internet topology.

A network is a set of nodes that are interconnected based on some structural and/or functional relations. When sampled, many of these relations may get broken or modified. This results in a sample topology with different topological characteristics, particularly when the sample size is small. In other words, the sample topology may *not* necessarily *resemble* the original topology in *every* aspect. This observation suggests that the topology sampling technique should be selected in a way that the resulting sample topology should *resemble* the original topology with respect to the characteristic in question. Within the context of Internet measurement studies, we can think of several topology sampling techniques:

- **Node sampling**: choosing a number of nodes randomly from the network to collect information on node specific characteristics, e.g., node degree.

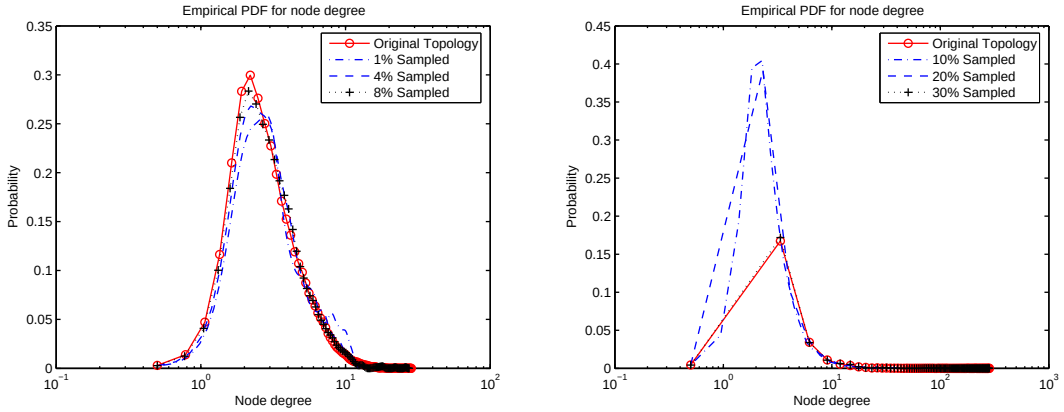
- **Edge sampling**: choosing a number of edges randomly from the network to collect information on edge specific characteristics, e.g., capacity.
- **Path sampling**: choosing a number of paths randomly from the network to collect information on path specific characteristics, e.g., path length.

Most measurement studies use sample Internet topologies that are collected by employing path sampling technique (i.e., traceroute based topology sampling) as it is the only router level sampling technique available to us in the Internet. However, the above discussion gives us a new angle to question, for instance, the relevance of using a path sampling-based Internet map when studying, say, the degree distribution characteristics of the nodes in the Internet.

To illustrate our point, we conduct an experiment where we examine the impact of the topology sampling methods in studying the degree distribution characteristics of a given synthetic graph. Even though the degree distribution characteristic is not the main characteristic studied in this paper, it will serve as a good example to demonstrate the need for characteristic-oriented topology sampling. Later on, in Section 3.4, we present an experimental study to demonstrate the representativeness of our topology collection approach for the analysis of intersection characteristic.

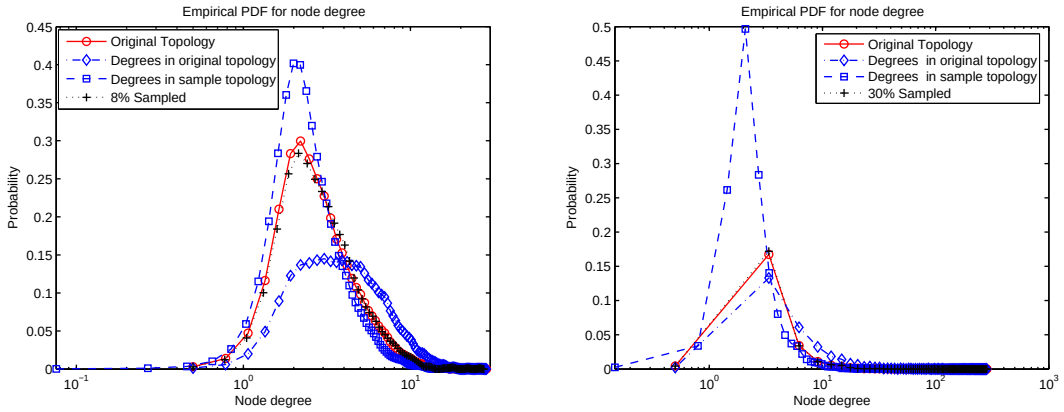
First, we use BRITE topology generator [38] to generate two synthetic graphs of 10K nodes: one *Waxman* graph and the other *Barabasi-Albert* power-law graph. We use two different topology sampling methods to collect sample topologies from these networks. They are (1) random node sampling where we randomly choose a number of nodes and collect their degree information and (2) traceroute-based path sampling where we randomly choose a number of node pairs and collect the minimum hop path between each node pairs. For node sampling, we use different sample sizes and collect degree information from each sampled node. Figure 1 illustrates the degree distribution characteristics of the original and the sample graphs. According to the results, in the case of the Waxman graph, a small size sample topology (8% or less) can give fairly accurate information about the degree distribution of the original graph. However, in the case of the power-law graph, since the variation of degree distribution in power-law graphs is higher than that in the Waxman graph, the required sample size must be increased to have a close approximation to the original degree distribution. In our experiments, we see that sampling 20% to 30% of nodes provides a good fit in power-law graphs.

We now use (`k`,`m`)-traceroute queries to collect path traces from `k` source nodes to `m` destination nodes in the original network. Then we compare the degree distribution of the nodes in the original topology and the ones in the (`k`,`m`)-traceroute sample topology. Figure 2 shows the results for (50,100)-traceroute sample topology for the Waxman and power-law topologies. In the case of the Waxman graph, traceroute-based sample topology includes over 30% of



(a) 10K Waxman Graph (b) 10K Power-law Graph

Fig. 1. Node degree distribution: Original graph vs. Random node sampled graph - using PDFs.



(a) 10K Waxman Graph (b) 10K Power-law Graph

Fig. 2. Node degree distribution: Original graph vs. (50,100) traceroute sample graph.

the nodes in the original topology. Still, the degree distribution of the sample topology is significantly different from that of the original one. Similar results are observed for the power-law topology. The figure also includes results for 8% (for Waxman) or 30% (for power-law) random node sampling cases for comparison purposes.

Note that, in Figure 2, we consider two types of node degrees for traceroute-based sample topologies. The one labeled "Degrees in sample topology" refers to the degree of the nodes within the sample topology only. This is what we get in a real network environment (assuming no IP alias resolution or anonymous router resolution problems - see Section 3). On the other hand, the one labeled "Degrees in original topology" refers to the actual degree of the sampled nodes in the original topology. Note that, in practice, this is not possible to obtain

in the Internet. However, since we use synthetic graphs, we are able to include it for comparison purposes.

In summary, we observe that maps constructed using path-sampling technique may not form representative samples to reflect the degree distribution characteristics of the original topology even when the sample topology includes up to 30% of the nodes in the original graph. In contrast, when random node sampling is used, we can obtain almost the same degree distribution as that of the original graph with much smaller sample sizes. Our conclusions from this experimental study is that the sampling technique does affect the outcomes of the measurement study. Therefore, one should consider the nature of the measurement task to identify the best possible sampling methodology in collecting the topology samples to be used in the study.

3.2 Topology Collection

In this section, we present the methodology that we use in sample topology collection. As mentioned previously, we need a topology map to study the overhead distribution of value-added Internet services. Based on the discussions presented in the previous subsection and considering the practical limitations of the existing topologies, we adapt the following methodology for topology collection.

In order to study the router-level state and processing overhead distribution of value-added services, it is deemed necessary to use a realistic Internet map that should have the following properties:

- Vantage points should be end points (or close to end points) in the Internet. This is so that the analysis represents the *end-to-end* load distribution characteristics.
- The map should include the path information among all vantage points as much as possible. In other words, the topology should represent the Internet topology among the vantage points (as dictated by the underlying routing protocol) corresponding to a (`n`,`n`) topology rather than a (`k`,`m`) topology. Topology maps are often obtained based on traces from a few vantage point to a large number of subnet prefixes. Naturally, the resultant maps become a tree-oriented topology. Such a topology is not sufficient for our analysis because it excludes significant amount of path information (path traces) among all the end points (leaves of the underlying trees).
- The map should include path traces in both directions between two end points and should not use path symmetry assumption which may not hold all the time in the Internet [39].
- The vantage points should be carefully selected so as to avoid any topolog-

ical imbalance that may cause bias during our experiments. As an example, having a single vantage point in, say, Japan or Australia along with a large number of vantage points in North America may result in heavy load accumulation on the routers toward this remote vantage point. This may introduce potential bias for our experiment results, and, hence, should be avoided.

Having stated the properties that we want to have in our Internet map, we can now look at the **available Internet maps** and discuss why they are insufficient for our analysis. There have been several topologies collected and used in other measurement studies. For example, Pansiot and Grad collected end-to-end routes in order to construct representative multicast tree topologies [14]. Their data set includes 11 Internet-wide tree topologies which are collected by running traceroutes from these 11 vantage points to over 5000 subnets on the Internet. Due to its tree nature, this data set is not suitable for our analysis. Paxson collected end-to-end path information among 37 vantage points and used this data set to study end-to-end routing behavior in the Internet [39]. Paxson's data set is suitable for our study but is of limited size. Finally, Spring et al. used traceroute queries from 750 publicly available traceroute vantage points [12] most of which are not end points. These traces aim at discovering internal topologies of ISP networks and are not necessarily complete end-to-end traces. Therefore, they are of limited use for our purposes.

In addition to the above studies, there are three well-known collaborative efforts that provide Internet measurements support in large scale. They are Skitter [18] project of CAIDA, PlanetLab [40], and Active Measurement Project (AMP) [17] of National Lab for Advanced Network Research (NLANR). Skitter has 30 publicly available monitors that provide traffic measurement services for researchers. PlanetLab is a Internet-wide measurement test bed that has around 170 monitors at 70 different locations world wide at the time of our data collection (late 2004). AMP has 150 measurement monitors most of which are deployed in the United States. AMP monitors provide the infrastructure to take site-to-site measurements on high-speed research networks for monitoring and/or debugging purposes for the networking community.

Our Data Set

To obtain a representative end-to-end router-level Internet map having the aforementioned properties, we significantly benefit from the resources of the above projects. Specifically, the vantage points that we use in our work include 120 measurement sites used by AMP project of NLANR and 33 traceroute servers that are listed at `www.traceroute.org` web site. We observed that most of our vantage points are located at universities or research institutions in North America and most of them are connected to Internet2. We also observed a significant overlap between our vantage points and the active measurement

sites of the PlanetLab.

While choosing the vantage points, we paid attention to choose the sites located at the periphery of the network. For this, we first used *ipas* tool [41] to map IP addresses of the vantage points to their AS numbers. Then, by consulting an AS level Internet map from [34], we classified these ASes as stub ASes and others. Most of the resulting candidate vantage points were located in North America and there were several others from Europe and Far East. Given the significant difference in the number of vantage points in North America and other parts of the world, we decided to use the vantage points located in North America only. Using a small number of vantage points at the other parts of the Internet would introduce an artificially large amount of load onto the routers around these nodes and would therefore result in a bias in the experimental results that we present in this paper.

At the end of this process, we were left with 153 vantage points that are located at stub ASes in North America. Finally, we used traceroute tool and collected end-to-end paths (153*152 traces) between all vantage points. After eliminating incomplete path traces and paths with loops, we had 19,739 path traces in our data set, based on which our topology is formed as discussed next. Note that, in this study, we tried to maximally utilize relevant vantage points to increase the number of sources.

3.3 Data Processing

After collecting the end-to-end paths, the next step is to process the data set to build an Internet map. This task involves two steps: (1) alias resolution for the routers that have multiple IP addresses and (2) resolving the identities of the unresponsive routers, i.e., routers causing traceroute program to print a '*' during the trace.

Alias Resolution

The first step in data processing is to resolve IP aliases of the routers. A router may respond to different traceroute queries with different interface IP addresses. This results in a situation where traceroute returns a list of interface IP addresses but does not group these interfaces into routers. Alias resolution refers to the process of checking if two (or more) given IP addresses belong to the same router.

There are two well-known IP alias resolution tools: *mercator* [42] and *ally* [43]. *Mercator* resolves aliases by using source IP addresses of ICMP PORT UNREACHABLE responses. *Mercator* sends an ICMP probe to each of the two IP addresses that are in question. If two probes with two different IP ad-

dresses result in ICMP responses with the same source IP address, then these two probed IP addresses are assumed to be aliases for the same router. *Ally* extends *mercator* by including a second step where it checks the IP identification field values in the IP protocol header of the returning ICMP response messages. The intuition in this approach is that even if the ICMP responses for two alias probes have different source IP addresses, they can still belong to the same router if they have close IP identification values. Given two IP addresses, *ally* tool returns three possible answers: "alias", "not alias", or "unknown" followed with an explanation. "Unknown" is returned when at least one of the probes does not result in a response. This can happen as some ISPs configure their routers to ignore probes directed to themselves.

Since *ally* is an improvement over *mercator*, we used *ally* to resolve IP aliases in our data set. Using *ally*, we detected 1536 IP alias pairs corresponding to 435 unique routers. The maximum number of aliases that a router has in our data set is 23. On the other hand, 79% of IP addresses (around 5900 addresses) did not have any alias. At first look, this result suggests that 5900 IP addresses represent 5900 different routers in our data set. But, after carefully studying the output of *ally* probes, we noticed that *ally* probes to 3122 (out of 7073) IP addresses did not return any response. This suggests that some of the IP addresses among 5900 addresses may correspond to the same router.

In a recent study [44], we presented a new alias resolution approach called *AAR*. In AAR, we mainly use the collected path traces to infer IP aliases within the collected data. The main observation that we use is as follows.

Internet consists of a number of ISP networks. In general, an ISP network consists of high speed routers that are connected to each other by high capacity links. Some of these connections use point-to-point links and others may use high speed multiple access links. The IP address assignment mechanism adheres to the guidelines presented in the Internet Registry IP Allocation Guidelines (RFC-2050). For point-to-point links between two router interfaces, a /30 subnet address is defined and used for IP address assignment to the interfaces. Recently, the use of /31 is made possible for point-to-point links (RFC-3021). For instance, the subnet of the point-to-point link in Figure 3 can be either /30 or /31. For both cases, the IP address assignment for interfaces of routers A and B are shown in the Figure 3 where x's denote the network address.

The common IP address assignment practice for point-to-point links results in a relation between the two IP addresses, say IP_A and IP_B, such that

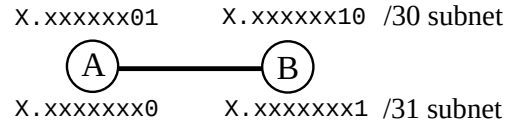


Fig. 3. IP address assignment for a point-to-point link.

No.	UWash to UConn (direct path)	UConn to UWash (reverse path)
1	140.142.153.10	140.143.16.229
2	140.142.153.23	140.142.153.10
3	198.107.151.12	198.107.151.51
4	**198.107.144.2**	**198.107.144.1**
5	**198.32.8.50**	**198.32.8.49**
6	**198.32.8.14**	**198.32.8.13**
7	**198.32.8.80**	**198.32.8.81**
8	**198.32.8.76**	**198.32.8.77**
9	**198.32.8.83**	**198.32.8.82**
10	**192.5.89.9**	**192.5.89.10**
11	**192.5.89.74**	**192.5.89.73**
12	159.247.232.130	159.247.232.132
13	137.99.22.42	137.99.22.41
14	137.99.23.139	137.99.23.189

Table 1
Tracroute results between Univ. Washington and Univ. Connecticut.

($\text{IP}_A = \text{IP}_B \pm 1$). Consecutiveness of IP addresses on a point-to-point link can be used to track potential path symmetry in path traces between two end points. That is, given path traces between two end points X and Y, one can look for a pattern in the form of ($\text{IP}_A = \text{IP}_B \pm 1$) where IP_A is an IP address in the path trace from X-to-Y and IP_B is an IP address in the path trace from Y-to-X. Once such a match is observed, IP aliases can be inferred from the proper alignment of the path traces.

We use the above procedure to infer IP aliases on a pair of path traces between University of Washington (UWash) and University of Connecticut (UConn). Table 1 displays the traceroute output from UWash to UConn on the left column and shows the *reverse* of the traceroute output from UConn to UWash on the right column. By closely examining these two traces, we can observe correlation between IP addresses in the 4^{nd} row till the 11^{th} row. Assuming point-to-point links with /30 or /31 subnets, we can construct the path segment corresponding to the traces as in Figure 4. This arrangement then can be used to detect IP aliases: 198.32.8.50 and 198.32.8.13 are IP aliases representing router b, 198.32.8.14 and 198.32.8.81 are IP aliases representing router c, etc.

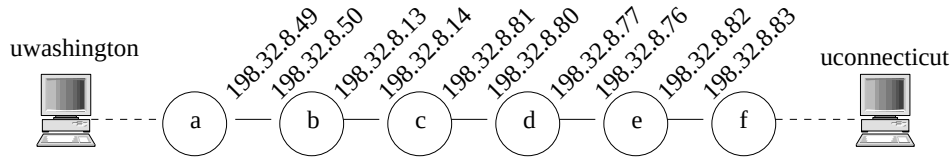


Fig. 4. Detecting IP aliases on a symmetric path segment.

The collected path pairs between two end points in our data set are used to detect IP aliases. Even though path asymmetry is a commonly observed property in end-to-end Internet paths, we can check for symmetric path *segments* and use this symmetry to resolve IP aliases. If the data set does not contain symmetric path segments then there may be only few aliases to be resolved.

Overall, we had 5652 unique nodes to represent 7073 IP addresses at the end of alias resolution process. Our alias resolution approach especially helped detecting IP aliases of routers in the backbone network in our data set. In Section 5, we report on the effects of the improvements in alias resolution on a set of experiment results that we present in this paper.

Resolving Unresponsive Routers

The second step in data processing is to identify unresponsive routers causing traceroute to display '*'s during the trace. This task is important because more than half of the traces contain at least one '*' corresponding to an unresponsive router. Routers that are configured not to respond TTL expiration event cause traceroute to display '*' in its output. Note that the simplistic approach that assigns a unique IP address to each unresponsive router would not be suitable as some of these unresponsive routers may in fact be the same router.

To resolve unresponsive routers in different traces, we compare all path pairs (say p_1 and p_2) with unresponsive routers and give them the same IP address as follows:

- Suppose p_1 and p_2 contain one '*' between two known routers. If the corresponding '*' entries have the same upstream router and the same downstream router while both p_1 and p_2 have the same final destination, then we consider such unresponsive routers as the same router and assign a unique name, e.g., *ur.1*, to it. This case is illustrated in Figure 5(a). Since the A-to-D and B-to-D traces include an unresponsive router which has the same upstream router x, the same downstream router y, and the same destination D, we assign *ur.1* to it.
- Suppose p_1 and p_2 contain two consecutive '*'s between two known routers. Similar to previous procedure, we first cluster these routers and give the same name to routers in the same cluster if the cluster has the same upstream, the same downstream routers, and the same destination, as illustrated in Figure 5(b).

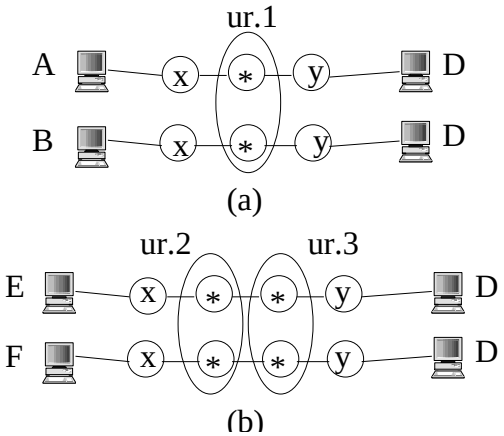


Fig. 5. Resolving unresponsive routers.

- Discard traces having more than two consecutive '*'s.

This way, we mapped 2748 unresponsive routers (i.e., 2748 occurrences of '*'s) to 406 distinct routers. Finally, we obtained our router-level map with 6,058 unique nodes and 13,873 links using 19,739 paths. Compared to topologies that have been collected by using (k,m)-traceroute queries (k≪m), we expect our (n,n)-traceroute topology to be more suitable for studying the end-to-end path intersection characteristics.

3.4 Representativeness of Our Data Set

One question at this point is how representative our topology map is. Given the scale and the complexity of the Internet and the lack of necessary resources, it would be too difficult (if not impossible) to obtain a highly realistic map of the Internet. Moreover, due to the lack of knowledge about the actual characteristics of the Internet, it is not possible to be sure about the representativeness of any sample map. Besides, at router-level, current Internet necessitates path sampling, which may not be suitable for all purposes but fits our study when (n,n)-sampling is performed.

Qualitatively speaking, it is clear that as the sample size increases, the collected data (our map) will be more and more representative for the sample space (the Internet). With this in mind, we conducted our measurement study. We believe that we maximally utilized the resources publicly available to us and obtained a large size end-to-end router-level map conforming to our constraints as outlined in Section 3.2. Note that, the data set may seem to be small compared to (k,m)-traceroute data sets, but it is a large one considering (n,n)-traceroute based data sets. A (k,m)-traceroute based map would be a larger one as we can have a large m but then the nature of the topology would not be suitable for our study. In our study, we maximize relevant vantage

points to increase the number of sources, i.e., n.

Having said that, we want to look for some quantitative evidence to demonstrate the representativeness of our data set. During the recent years, researchers have used the *power-law conformance* [19] and *sampling bias* [27] tests to evaluate their data sets. The main idea in these tests is to check the degree distribution characteristic of the data set and compare it with that of the available data sets. The observed similarities between the new data set and the existing ones are used to justify the representativeness of the collected data set at hand.

Considering the arguments presented in Section 3.1, we believe that the above mentioned tests are not suitable to test the representativeness of our data set. That is, our topology collection is geared toward collecting a data set for studying the load distribution characteristic of value-added Internet services. On the other hand, the above mentioned tests are designed to test the degree distribution characteristic of a topology map. Since the two characteristics are different from each other, using a procedure designed for the one to test the other would not probably produce a reliable result [1].

Based on this observation, we decided to employ an alternative approach to demonstrate the representativeness of our topology collection approach. Our goal is to demonstrate that a (n,n)-traceroute topology sampling approach results in a more representative sample topology as compared to a (k,m)-traceroute topology sampling approach for studying the load distribution characteristic. We believe this test is more suitable to our study than aforementioned tests since it considers the load characteristic that we study.

Comparison of (n,n)- vs (k,m)-Traceroute based Topology Collection

Our load definition resembles betweenness centrality measure which identifies potential traffic load on a node by counting the number of shortest paths that traverse the node [45]. In the classical definition of the betweenness, however, there is no distinction between nodes as routers or end-hosts. In studying the end-to-end load distribution characteristic, we consider the end-to-end paths only and do not consider the paths between the routers. Therefore, in our context, **load** on a router is defined as *the ratio of the number of end-to-end paths traversing the router to the total number of traces taken in the network.* That is,

$$L(v) = \sum_{s \neq t \neq v \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

[1] Despite the arguments presented here, we used the power-law conformance and sampling bias tests on our data set. According to test results, our data set complies with the power-law conformance tests but presents sampling bias.

where σ_{st} is the number of shortest paths from s to t, $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through vertex v, and s and t has a degree of 1 (i.e., s and t are end-nodes).

For our comparisons, we use a 10K node *Barabasi-Albert* power law based synthetic network that we generated using BRITE [38]. We then identified a number of nodes (approximately half of them) among small degree nodes (degrees being less than the median degree which is 4) to represent the periphery nodes of the network (corresponding to edge routers of a computer network). Next, we linked a total of 10K new nodes to the periphery of the network to represent the end-systems in the network. The end-system nodes are added in a way that they can appear only as a source or a destination of a path trace in the network. Finally, this network represents our original network topology from which we collect both (`k`,`m`) and (`n`,`n`)-traceroute sample topologies. We select the `k`, `m`, and `n` values such that the resulting sample topologies are formed by using the same number of path traces. We then compare the load characteristics of the original topology and the sample topologies.

In our comparisons, we use the average load on a number of highly loaded routers as our comparison metric. Our expectation is that the load on the highly loaded routers in the case of (`n`,`n`)-traceroute topology samples better resemble (as compared to that in the case of (`k`,`m`)-traceroute topology samples) the load value of the highly loaded routers in the original topology. Note that the load of highly loaded routers is more critical to network operation than the load of lightly loaded ones. Hence, we analyze the representativeness of samples considering the highest loaded routers.

Table 2 presents the average results of 20 runs of different sample topologies formed by collecting 2450, 6320, and 9900 traces from the original network respectively. The first column in the table shows the approach used to collect the sample topologies. The first entry (i.e., (`10K`,`10K`) entry) corresponds to traces among all end points. The following entries are for different (`k`,`m`) and (`n`,`n`) values. The second column shows the number of routers in the sample topologies. The third to the last columns show the average load on a number of highly loaded routers for different sample topologies collected from the original topology. More specifically, the third column shows the load of the highest loaded router, the fourth column shows the average load among the 0.1% highest loaded routers, the fifth, the sixth and the seventh columns show the average load among the 0.2%, 1% and 2% highest loaded routers, respectively. Finally, the values for the first row (i.e., (`10K`,`10K`) row) indicates the values for the original topology for comparison purposes.

We now carefully analyze the table to see whether (`k`,`m`) or (`n`,`n`) samples better resemble the original topology's load characteristics. We observe that as the disparity of `k` and `m` reduces the average load for 0.1% highest loaded routers

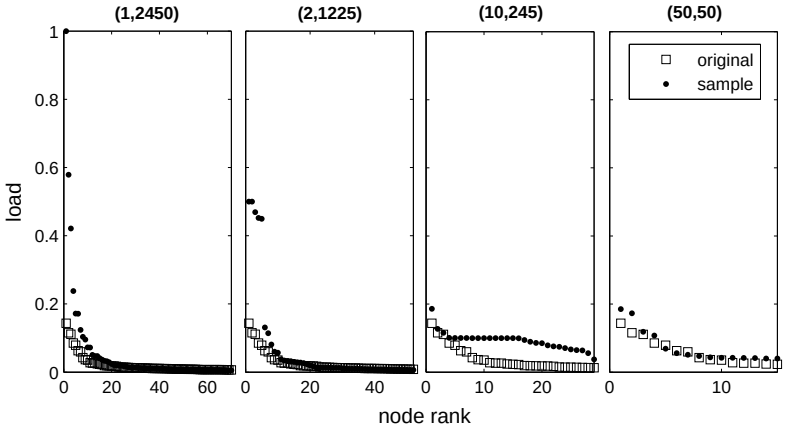
	avr. number of routers	average load				
		max	max 0.1%	max 0.2%	max 1%	max 2%
(10K, 10K)	10000	0.143	0.077	0.050	0.017	0.010
(1, 2450)	3523.9	1.000	0.651	0.419	0.106	0.057
(2, 1225)	2565.2	0.503	0.485	0.425	0.132	0.072
(10, 245)	1428.0	0.195	0.194	0.154	0.112	0.090
(50, 50)	769.5	0.189	0.189	0.177	0.107	0.076
(2, 3160)	4709.5	0.500	0.412	0.295	0.080	0.043
(4, 1580)	3558.4	0.289	0.260	0.241	0.095	0.053
(16, 395)	2319.5	0.176	0.154	0.123	0.076	0.060
(80, 80)	1316.1	0.182	0.182	0.146	0.075	0.052
(3, 3300)	5149.7	0.356	0.319	0.244	0.072	0.039
(5, 1980)	4276.8	0.260	0.216	0.199	0.080	0.044
(20, 495)	2812.5	0.190	0.144	0.114	0.064	0.050
(100, 100)	1676.4	0.179	0.155	0.132	0.064	0.044

Table 2
Average load of highest loaded nodes for (k,m)- and (n,n)-traceroute sampling.

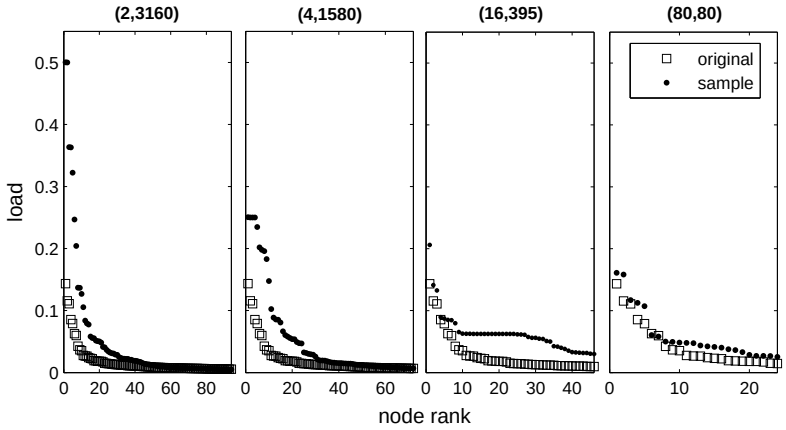
becomes closer to the original network's load for 2450 trace samples. (k,m) samples even yield better results than (n,n) samples for 6320 and 9900 traces. On the other hand, looking to the average load of 2% highest loaded routers, it seems the average load is closer to the original networks' load when the disparity is high. Overall, the load values for the (n,n)-traceroute topology samples seem to be closer to that of the original topology as compared to (k,m)-traceroute topology samples.

Figure 6 shows the load distribution of 2% highest loaded routers for (k,m) and (n,n)-traceroute based sample topologies of a single run. This run is the closest run to the average of all runs presented in Table 2. According to the figures, the load values on the highly loaded individual routers in the (n,n)-traceroute sample topologies are closer to that of the original topologies when compared to the (k,m)-traceroute sample topologies. (k,m)-traceroute samples seem to resemble the average load of the topology when the disparity between k and m is high. However, this is also when the top ranked nodes' load is highly overestimated. When the disparity is reduced, (k,m)-samples resemble the top rank better but then disturb the 2% highest loaded nodes' load.

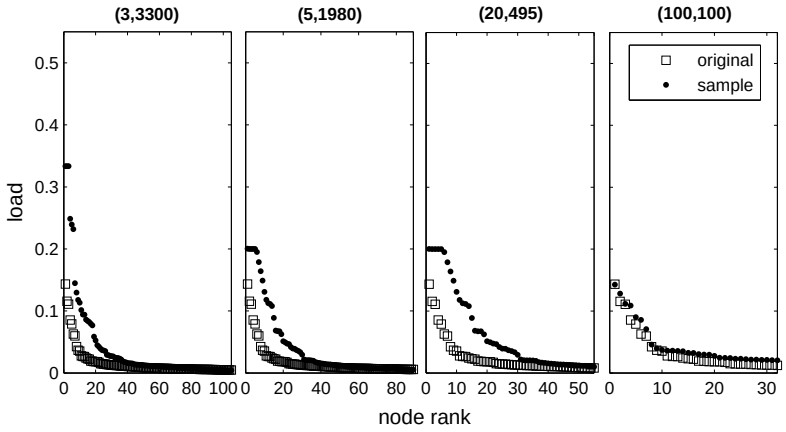
In addition to load distribution, we compare the load of routers in the sample network with that of the original network in Figure 7. (These values are from



(a) 2450 trace sample topologies

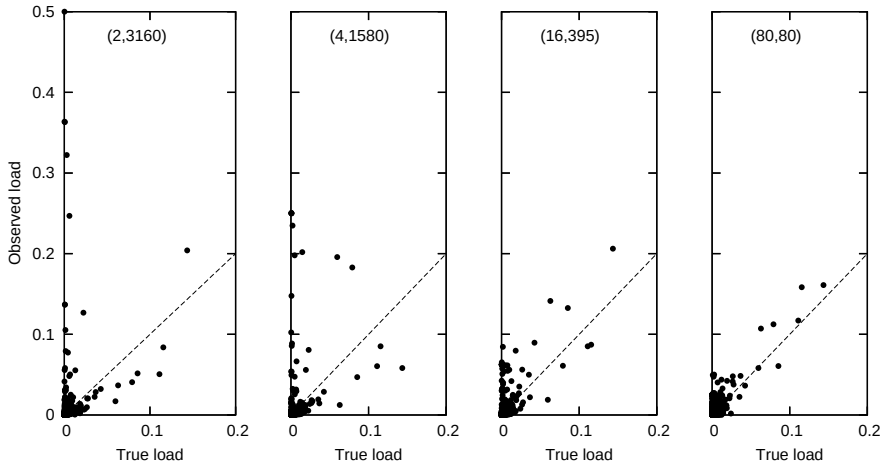


(b) 6320 trace sample topologies

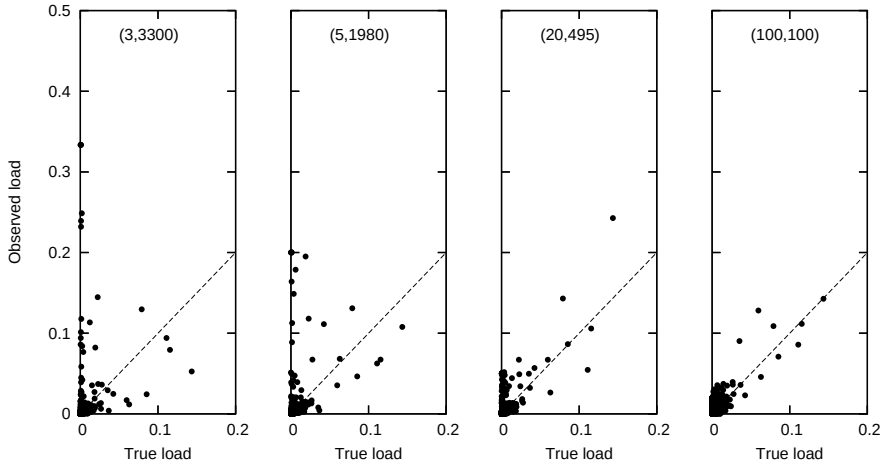


(c) 9900 trace sample topologies

Fig. 6. Load distribution for the 2% highest loaded nodes.



(a) 6320 trace sample topologies



(b) 9900 trace sample topologies

Fig. 7. Load Comparison

the network used in Figure 6.) In these figures, 'Observed Load' indicates the load of the routers in the sample topology and 'True Load' indicates the load in the original topology. Note that, each point in these figures may correspond to one or more routers in the sample topology with the same 'Observed' and the same 'True' load. In all cases, (`k`,`m`)-traceroute based samples show greater deviation from x=y line, indicating the load values of routers in samples are further away from the original network's. For instance, in (2,3160) sample topology, two routers have a load of 0.5 while in the original network their load is less than 0.001. This indicates, nodes of very low significance in the original topology became the most significant nodes in the sample topology. Overall, these figures also indicate that (`n`,`n`)-traceroute based sample topologies better resemble the original topology considering the load characteristic since its values are closest to x=y line.

We have also conducted experiments with different type (e.g., Waxman) graphs and different size synthetic topologies (3K, 5K, and 25K topologies) and observed similar results. These results suggest that (n,n)-traceroute topology sampling results in more realistic sample topologies with respect to the load characteristic of the routers.

4 Load Distribution on the Internet

One common characteristic of value-added network services is that they incur state and/or processing overhead that we briefly call "*load*" on the routers in the underlying paths or trees. For example, IP multicast, IntServ, and p2cast require to establish connections along the underlying end-to-end paths, resulting in state overhead on the routers in these paths. In this context, multiple simultaneous connections between the same end systems can be reduced to one by using end-to-end tunnels [13]. This simply allows us to use the number of end-to-end paths (or trees) crossing over a router as the state (*load*) on that router. For DiffServ, we can consider the number of paths intersecting at the edge routers as DiffServ requires to maintain state information at the edge. For IP traceback approaches, packets destined to each traceback enabled destination introduce state and/or processing overhead on the routers. We can deal with state overhead again by analyzing the number of intersecting paths. However, it is difficult to characterize processing overhead due to the lack of actual data about the traffic between end points. If we assume that the amounts of traffic between end points are random variables and that these random variables are identically distributed, then it is easy to see that the number of end-to-end paths crossing over a router will also be a good indication for the level of processing overhead on that router. From the foregoing discussion, we mainly consider the number of end-to-end paths intersecting at a router as the "*load*" on that router, and thus analyze it under various cases.

4.1 Load Distribution in Multicast Context

In this section, we present our analysis on multicast state scalability problem at the router level. We first investigate the effects of two important parameters, namely *usage rate* and *session density*, on multicast state distribution in the network. Usage rate refers to the number of multicast groups in the network and session density refers to the number of receivers in a multicast group. By considering scenarios with different usage rates and different session density values, we examine state distribution characteristics under various cases. We also examine state distribution at backbone and exchange point routers as they constitute potential scalability bottleneck points. Finally, we revisit the

effectiveness of multicast state elimination approaches that focus on improving multicast state scalability.

Effect of Usage Rate and Session Density

In this set of experiments, we use different combinations of session density (trees with 2, 15, and 50 receivers) and usage rate (2, 5, 10, 15, and 50 trees) levels. In each experiment, we form multicast trees by randomly choosing the sources and the receivers among the vantage points according to usage rate and session density values respectively. Then, for each experiment, we count the number of states at backbone and exchange point routers. We use DNS information and the topology map of the Abilene Network to identify the backbone and exchange point routers in our data set [46,47]. We run 10 experiments for each session density and usage rate case. We observe somehow similar results in the individual runs and therefore use only these cases for the evaluations. The results of the experiments are shown in Tables 3 and 4 as the average overhead for each experiment.

According to the first rows (2 receiver tree case) in Tables 3 and 4, at low session densities, the backbone routers have relatively more load than the exchange point routers especially at high usage rates. On the other hand, as session density increases, the load at exchange point routers get closer to the load at the backbone routers (see the third rows in Tables 3 and 4).

We believe that this behavior is expected. That is, at low session densities, the first receiver will incur state on at most two exchange point routers. This

	Usage Rate (Num Trees)				
Session Density	**2**	**5**	**10**	**15**	**50**
2 Receiver Trees	0.36	1.27	2.28	3.18	10.32
15 Receiver Trees	1.27	2.90	6.68	8.90	27.86
50 Receiver Trees	1.64	3.82	8.41	11.90	35.73

Table 3
Average load at backbone routers (w.r.t. usage rate)

	Usage Rate (Num Trees)				
Session Density	**2**	**5**	**10**	**15**	**50**
2 Receiver Trees	0.33	1.00	0.92	1.83	6.58
15 Receiver Trees	1.00	1.83	4.34	5.83	15.00
50 Receiver Trees	1.50	3.33	7.33	11.16	32.75

Table 4
Average load at exchange point routers (w.r.t. usage rate)

happens when the path from receiver to sender crosses over the backbone. The additional receivers will then incur state on at most one exchange point router (assuming a single backbone domain). On the other hand, these receivers may incur state overhead on more than one backbone router. Therefore, at low session densities, the backbone routers are likely to get more states than the exchange point routers. While we increase the session density, the probability that each exchange point leading toward a receiver (or a multicast sender) will increase, and, therefore, most of the exchange points will incur state overhead for many multicast trees. On the other hand, since it is a low probability for a backbone router to be on *all* end-to-end paths, the load on backbone routers will be limited. As we increase the usage rate and session density levels to 50%, the average load on exchange point routers exceeds the average load on backbone routers (results not shown).

Another observation from the analysis is that multicast usage rate seems to be a more effective parameter for state scalability at backbone and exchange point routers. According to Table 3, as we increase the usage rate from 2 trees to 50 trees at a session density of 2-receiver trees (the first row of the table), the average state overhead at backbone routers increases from 0.36 to 10.32. On the other hand, if we fix the usage rate at 2 trees and increase session density from 2 receiver trees to 50 receiver trees (the first column of the table), the average state overhead at backbone routers increases to 1.64. Since the increase in the first case is more, we conclude that usage rate is a more effective parameter for state scalability at backbone routers. A similar conclusion can be reached for exchange point routers in the same way by analyzing Table 4.

Multicast State Elimination - Revisited

We now consider the effectiveness of multicast state elimination approaches. These approaches aim at reducing the state overhead by eliminating unnecessary multicast states on tree routers (see Section 2.2 for more details). We divide the studies in multicast state elimination into two categories. In the first category, the main idea is to remove unnecessary states at non-branching routers in a multicast tree. In the second one, an aggregated multicast approach is used to build domain-local multicast tunnels to reduce the number of states within a transit domain. In this section, we look at these two approaches separately.

In the first category, researchers evaluate the effectiveness of state elimination approaches by looking at the number/ratio of non-branching states that are eliminated from the network. In [34], authors study state elimination on per-node resolution on AS level Internet maps and conclude that except for a negligible number of nodes, state elimination techniques are effective in reducing the number of states by removing the non-branching states from the

nodes at the AS level.

In our work, we look at the effectiveness of state elimination approaches on our router-level Internet map. For our evaluations, we consider several scenarios by choosing different number of multicast usage rates (i.e., 10, 25, and 50 trees) and session densities (i.e., 10, 25, and 50 receivers per tree). After constructing multicast trees, we count both the total number of states (both branching and

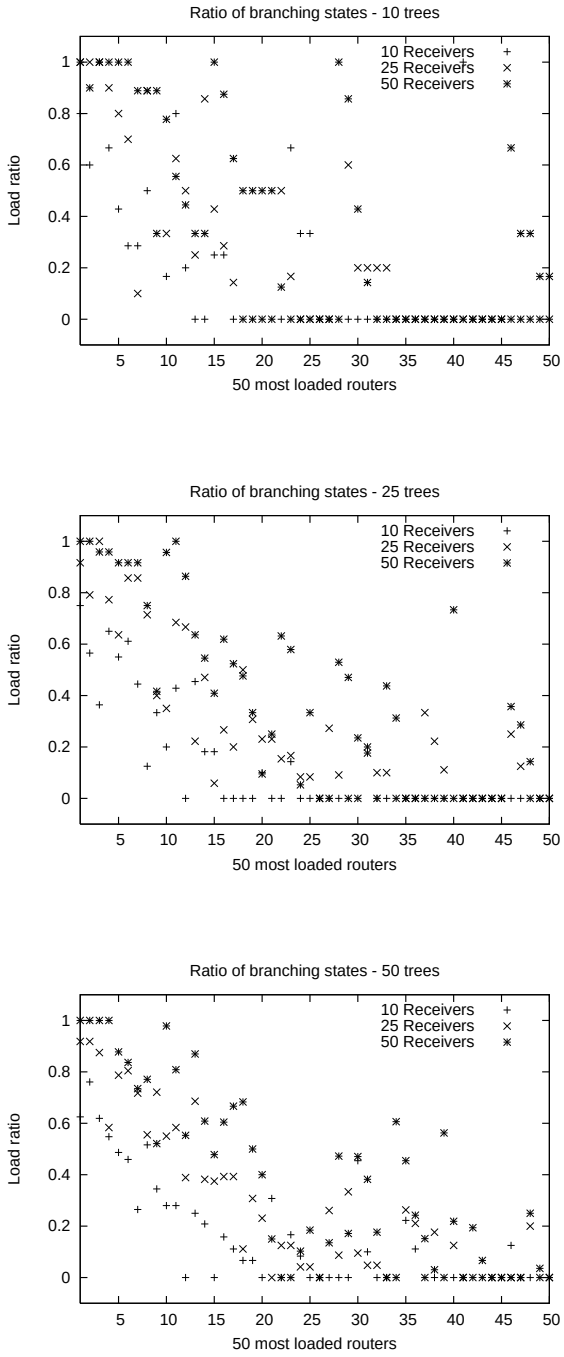


Fig. 8. Nonbranching state ratio on routers.

non-branching states) and the number of branching states on each router in the network. Figure 8 presents the ratio of branching states on 50 most loaded routers (most loaded with respect to total number of states).

According to the figure, we see that in each experimental scenario there are a number of routers whose branching ratio is significantly high. In other words, most of the states these routers are maintaining are branching states and non-branching state elimination techniques cannot help reduce the state overhead on these routers much. After a close look at the results, we observe that most of these routers are backbone or exchange point routers in our data set. This observation suggests that, contrary to the conclusions in [34], state elimination approaches are not necessarily effective in removing multicast states at bottleneck routers (e.g., backbone and exchange point routers). Since such bottleneck routers are potential performance choke points, eliminating non-branching states at other routers will not likely provide a practical solution to the state scalability problem.

In the second category, an aggregated multicast approach [13] is used to achieve state reduction through inter-group tree sharing within a transit domain. In this approach, multiple multicast trees that share the same ingress border router into and share the same (or a similar) set of egress border routers out of the transit domain use a single multicast tree. This tree is formed locally between the ingress router and the egress routers within the domain. Clearly, this approach helps reduce the forwarding states in the internal routers of a deploying transit network at the expense of performing additional processing at the ingress point for encapsulating and tunneling and at the egress points for decapsulating multicast data. According to the experimental evaluations presented in [13], the approach can provide up to 75% savings on multicast states at internal routers of the deploying domains. The provided savings is significant and the approach can help alleviate the state overhead in the internal routers. But the approach cannot help reduce the state overhead at the border routers of the domains.

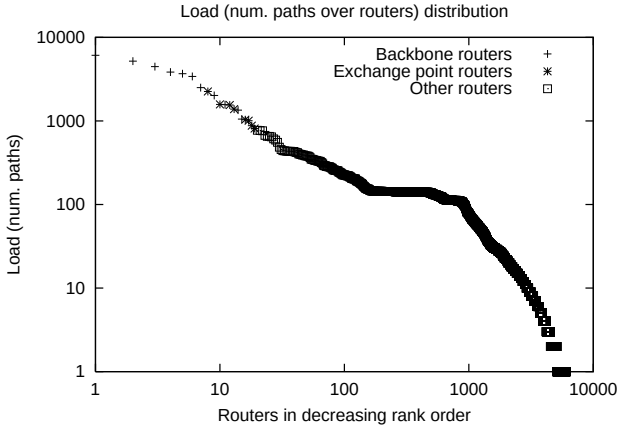
4.2 Load Distribution in Unicast Context

In this section, we study the worst-case load distribution characteristics of value added unicast services. Due to the lack of a representative model for large scale Internet user behavior, average case analysis requiring to consider different load patterns is left as a future work.

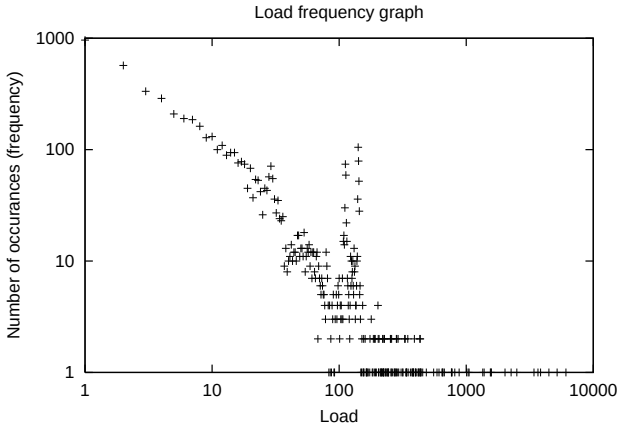
For the worst-case analysis, we consider all unicast paths (19,739 paths) among end points in our data set, and count the number of end-to-end paths passing over a router as the load (i.e., state overhead) incurred on that router.

Figure 9-a presents the worst-case load distribution in a decreasing order (i.e., rank distribution) in log-log scale. In addition, Figure 9-b presents the frequency distribution of the load. These figures show that a small number of routers have large numbers of paths passing over them and remaining significant majority of the routers appear on smaller number of paths. More specifically, there are 7 routers in the range [6095-2503], i.e., the router that is loaded highest appears on 6095 paths and 7^{th} highest loaded router appears on 2503 paths. According to our data set, all of these routers belong to Abilene backbone which corresponds to a significant portion of the backbone network in our data set. Then, we list 10 routers in [2503-879] range and these routers are mostly exchange point routers. Finally, 6,000+ routers fall in the range [879-1].

Next we check for a potential correlation between load and degree distribution. Figure 10 depicts the relation between the degree of the routers and their load. Here, we use the term degree to refer to the number of neighbors rather than



(a) Load distribution



(b) Load frequency distribution

Fig. 9. Load distribution in the worst-case scenario.

the number of interfaces of a router. Let D be a random variable denoting the degree of routers and L be a random variable denoting the load on routers. The *correlation* of D and L is defined by

$$\rho_{L,D} = \frac{E[DL] - E[D]E[L]}{\sqrt{\mathrm{VAR}(D)\mathrm{VAR}(L)}}.$$

Given that the part of the variables have heavy-tailed distribution, we take the logarithms of D and L and then compute the *correlation coefficient* as 0.4, showing that there is a positive but not a strong correlation.

Specifically, the highest degree for a router in our data set is 55. However, the load on that router is not as high as that on others. In addition, the degree of the router which has the highest load in the worst case is 40. And, the average degree for the *backbone* routers is 17. This observation makes perfect sense when we consider the hierarchical structure of the Internet topology. That is, in the Internet, it is the exchange point and border routers that have a large number of peers. Core routers, on the other hand, bear a large load (i.e., appears on a lot of end-to-end paths) but do not peer with a large number of other nodes. Note that previous studies that use AS level topology maps cannot reflect this observation.

In summary, in unicast environments, load accumulation at the core is significantly higher than at other parts of the network. Most of the highly loaded individual routers are backbone and exchange point routers. In addition, the results indicate that there is a positive but not so significant correlation between the degree of a router and its load.

From value-added services point of view, these results indicate that the potential overhead of such services would be significant at the backbone and exchange point routers. As an example, from IP traceback point of view, the routers at the core of the network would be overloaded more than the routers

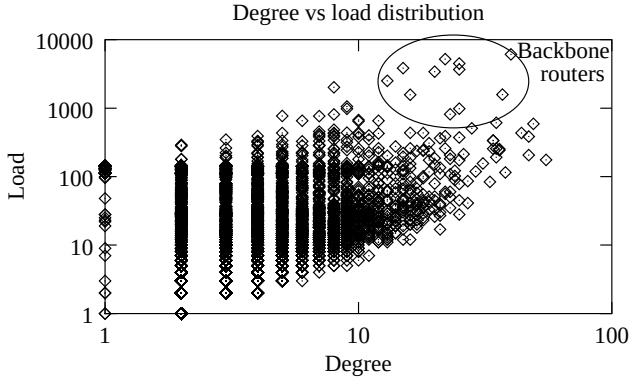


Fig. 10. Degree vs load distribution.

at the regional ISP networks and the local ISP networks. Therefore for the deployment and practical usage purposes, there is a need for an IP traceback approach that can provide the traceback service without incurring overhead on all the routers or that can provide the service with partial deployment (i.e., should work even if it is not deployed at the backbone ISP networks). In a recent study on IP traceback, we use our observations in this paper to build an autonomous system (AS) level IP traceback service to achieve this objective [48].

5 Effect of Alias Resolution on Measurement Results

In this section, we study the effect of incomplete alias resolution on our measurement results. For this, we use two versions of topology data: one generated by using *ally* to resolve IP aliases, called T_{ally}, and the other that uses both *AAR* and *ally*, called $T_{combined}$. First, we look at the effect of alias resolution on the load distribution in unicast case. Here, we consider the worst-case scenario and count the number of end-to-end unicast paths (load) passing over a number of backbone (e.g., Abilene backbone) routers. Table 5 shows the results.

Each row in the table represents a backbone router in $T_{combined}$. The second column gives the number of aliases that we identified for the routers and the

Router No.	**Num. Alias**	**Total Load**	**Min. Load**	**Max. Load**	**Avg. Load**
1	5	6095	313	2259	1219
2	8	5193	311	2014	649
3	5	4476	221	2142	895
4	5	3847	151	1418	769
5	5	3672	111	2005	734
6	6	3414	223	913	569
7	4	2503	483	798	626
8	3	2018	433	1007	673
9	3	1566	420	620	522
10	3	1054	110	599	351
11	3	769	118	337	256

Table 5
Effect of alias resolution on unicast load distribution at the backbone routers.

third column gives the total load on the routers in $T_{combined}$. Due to imperfect IP alias resolution, each of these routers appears as different routers in T_{ally}. In other words, due to the fact that none of these routers respond to probes directed to themselves, *ally* cannot detect any IP aliases for these routers. In our work, we first count the load on each router in T_{ally}. Then by using the AAR IP alias information, we group these routers into their alias groups. Then, for each alias group (an alias group in T_{ally} corresponds to the same router in $T_{combined}$), we include the minimum, the maximum, and the average of the load values in the fourth, fifth, and sixth columns, respectively. Average load is the average of the loads on the routers in the same alias group and the minimum and the maximum loads indicate the minimum and the maximum of the loads in an alias group.

According to the table, the 11 routers in $T_{combined}$ are represented as 50 different routers in T_{ally} causing over 4.5 fold artificial increase in the backbone network size. In addition, each backbone router in T_{ally} bears a unicast load that is significantly smaller than the actual load accumulated at backbone routers (column 3 in the table). This difference initially made us believe that our backbone network consisted of a larger number of nodes each one having smaller overhead during our experiments on unicast load distribution (see the differences between the values presented in columns 3 to 6 in Table 5). In addition, during our comparisons on multicast state distribution characteristics, the use of T_{ally} have indicated a trend where the state overhead at exchange point routers tend to be more than that of backbone routers for usage rate and session densities of 33% and more (as compared to 50% obtained by using $T_{combined}$ as reported in Section 4.1).

In summary, our experience in using T_{ally} and $T_{combined}$ topologies clearly demonstrates that the success of IP alias resolution significantly affects the shape of the resulting topology. This in turn significantly affects the nature of the results obtained from the measurement study that uses the topology.

6 Conclusions

In this paper, we have focused on two related issues. In the first part, we have conducted a measurement study to build a router-level Internet map. We have presented a detailed analysis of the challenges in building a representative network map from a set of collected path traces. Motivated with the limited success of existing IP alias resolution tools, we have developed and presented a new approach to improve the current state-of-the-art in alias resolution. We have also presented a procedure to resolve routers presented with a '*' in traceroute outputs.

In the second part, we have used our network map to analyze the distribution of state overhead incurred by value-added services in both multicast and unicast environments. Specifically, we have shown that usage rate (i.e., number of trees) of multicast services is more important than session density in increasing the overall state overhead in the network. This suggests that tunneling mechanisms (e.g., aggregated multicast) that combine multiple multicast trees into one single tree are effective in reducing the overall state overhead in the backbone. On the other hand, we have observed that such an approach is not always effective in reducing the state overhead at some of the heavily loaded border and exchange point routers. In the context of unicast services, we have shown that the backbone and the exchange point routers bear a heavy load. Therefore, it is deemed necessary to develop mechanisms that can reduce the state overhead not only at the core of the network (as done in DiffServ) but also at the border and exchange point routers.

References

[1] K. Almeroth, The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment, IEEE Netw. 14 (1-2) (2000) 10-20.

[2] R. Braden, D. Clark, and S. Shenker, Integrated services in the Internet architecture: An overview, Internet Engineering Task Force (IETF), RFC 1633, June 1994.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An architecture for differentiated services, Internet Engineering Task Force (IETF), RFC 2475, December 1998.

[4] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, Practical network support for IP traceback, ACM SIGCOMM, Stockholm, SWEDEN, August 2000.

[5] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, S. Kent, and W. Strayer, Single-packet IP traceback, IEEE/ACM Transactions on Networking, 10 (12) (2002) 721–734.

[6] A. Yaar, A. Perrig, and D. Song, Pi: A path identification mechanism to defend against DDoS attacks, IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 2003.

[7] A. Yaar, A. Perrig, and D. Song, FIT: Fast Internet traceback, IEEE INFOCOM, Miami, FL, USA, March 2005.

[8] K. Sarac, SSM-based receiver-controlled communication in the Internet, South Central Information Security Symposium, Denton, TX, USA, April 2003.

[9] A. Yaar, A. Perrig, and D. Song, SIFF: A stateless Internet flow filter to mitigate DDoS flooding attacks, IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 2004.

[10] X. Yang, D. Wetherall, and T. Anderson, A DoS-limiting network architecture, ACM SIGCOMM, Philadelphia, PA, USA, August 2005.

[11] V. Jacobson, *Traceroute tool*. Lawrence Berkeley Laboratory (LBL), February 1989. Available from ftp://ee.lbl.gov/traceroute.tar.Z.

[12] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, Measuring ISP topologies using Rocketfuel, IEEE/ACM Transactions on Networking 12 (2) (2004) 2–16.

[13] J. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla, Aggregated multicast – A comparative study, in Special issue of Cluster Computing: The Journal of Networks, Software and Applications 8 (2) (2005) 15–26.

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

[15] R. Caceres, N. Duffield, H. J., and D. Towsley, Multicast-based inference of network-internal Loss characteristics, IEEE Transactions on Information Theory 45 (11) (1999) 2462–2480.

[16] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, An architecture for large-scale Internet measurement, IEEE Communications, 36 (8) (1998) 48–54.

[17] A. McGregor, H.-W. Braun, and J. Brown, The NLANR network analysis infrastructure, IEEE Communications Magazine 38 (5) (2000) 122–128.

[18] D. McRobb, K. Claffy, and T. Monk, *Skitter*: CAIDA's macroscopic Internet topology discovery and tracking tool, 1999. Available from http://www.caida.org/tools/skitter/.

[19] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, Power-laws and the AS-level Internet topology, IEEE/ACM Transactions on Networking 11 (8) (2003) 514–524.

[20] A. Broido and k. claffy, Internet topology: Connectivity of IP graphs, SPIE ITCom Conference, Denver, CO, USA, August 2001.

[21] P. Barford, A. Bestavros, J. Byers, and M. Crovella, On the marginal utility of network topology measurements, ACM Internet Measurements Workshop, San Francisco, CA, USA, November 2001.

[22] B. Donnet, P. Raoult, T. Friedman, and M. Crovella, Efficient algorithms for large-scale topology discovery, ACM SIGMETRICS, Banff, Alberta, CANADA, June 2005.

[23] *DIMES Project*, Tel-Aviv University. Available from http://www.netdimes.org.

[24] J.-L. Guillaume and M. Lapaty, Relevance of massively distributed explorations of the Internet topology: Simulation results, IEEE INFOCOM, Miami, FL, USA, March 2005.

[25] L. Amini, A. Shaikh, and H. Schulzrinne, Issues with inferring Internet topological attributes, SPIE ITCom, Boston, MA, USA, July/August 2002.

[26] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, Topology inference in the presence of anonymous routers, IEEE INFOCOM, San Francisco, CA, USA, March/April 2003.

[27] A. Lakhina, J. Byers, M. Crovella, and P. Xie, Sampling biases in IP topology measurements, IEEE INFOCOM, San Francisco, CA, USA, March/April 2003.

[28] L. Li, D. Alderson, W. Willinger, and J. Doyle, A first-principles approach to understanding the Internet's router-level topology, ACM SIGCOMM, Protland, OR, USA, August 2004.

[29] D. Thaler and M. Handley, On the aggregatability of multicast forwarding state, IEEE INFOCOM, Tel-Aviv, ISRAEL, March 2000.

[30] P. Radoslavov, D. Estrin, and R. Govindan, Exploiting the bandwidth-memory tradeoff in multicast state aggregation, Technical report, University of Southern California, July 1999.

[31] L. Blazevic and J. Boudec, Distribured core multicast (DCM): A multicast routing protocol for many groups with few receivers, Networked Group Communication Workshop, Pisa, ITALY, November 1999.

[32] I. Stoica, T. Ng, and H. Zhang, Reunite: A recursive unicast approach to multicast, IEEE INFOCOM, Tel-Aviv, ISRAEL, March 2000.

[33] J. Tian and G. Neufeld, Forwarding state reduction for sparse mode multicast communications, IEEE INFOCOM, San Francisco, CA, USA, March 1998.

[34] T. Wong and R. Katz, An analysis of multicast forwarding state scalability, IEEE International Conference on Network Protocols (ICNP), Osaka, JAPAN, October 2000.

[35] D. Song and A. Perrig, Advanced and authenticated marking schemes for IP traceback, IEEE INFOCOM, Anchorage, AK, USA, April 2001.

[36] D. Dean, M. Franklin, and A. Stubblefield, An algebraic approach to IP traceback, Network and Distributed System Security Symposium, San Diego, CA, USA, February 2001.

[37] Z. Wang, Internet QoS: Architectures and Mechanisms for Quality of Service, Morgan Kaufman Publishers, 2001.

[38] *BRITE*: Boston university representative internet topology generator. Available from www.cs.bu.edu/brite.

[39] V. Paxson, End-to-end routing behavior in the Internet, ACM SIGCOMM, Stanford, CA, USA, August, 1996.

[40] *PlanetLab*. Available from http://www.planet-lab.org.

[41] *IPAS Tool*. Available from http://mna.nlanr.net/Software/IPAS/docs/index.html.

[42] R. Govindan and H. Tangmunarunkit, Heuristics for Internet map discovery, IEEE INFOCOM, Tel-Aviv, ISRAEL, March 2000.

[43] *Rocketfuel Project*. Available from http://www.cs.washington.edu/research/networking/rocketfuel/.

[44] H. Gunes and K. Sarac, Analytical IP alias resolution, IEEE International Conference on Communication (ICC), Istanbul, TURKEY, June 2006.

[45] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, kc claffy, and A. Vahdat, The Internet AS-level topology: Three data sources and one definitive metric, ACM SIGCOMM Computer Communication Review 36, (1) (2006) 17–26.

[46] *Reverse DNS Lookup*. Available from http://remote.12dt.com/rns/.

[47] *Abilene Internet2 Backbone*. Available from http://abilene.internet2.edu/.

[48] T. Korkmaz, C. Gong, K. Sarac, and S. Dykes, Single Packet IP Traceback in AS-level Partial Deployment Scenario, International Journal on Security and Networks 2 (1/2) (2007) 95–108.