Identity Trail: Covert Surveillance Using DNS

Saikat Guha, Paul Francis
Cornell University
{saikat, francis}@cs.cornell.edu

Abstract

The Domain Name System (DNS) was originally designed with the assumption that the DNS will return the same answer to any given query regardless of who may have issued the query, and that all data in the DNS is thus visible. Such an assumption can no longer be justified for private Internet hosts, particularly mobile laptops and PDAs. IP addresses in the DNS reveal a host's geographic location and corporate affiliation to anyone that is interested without the host's knowledge or consent. This paper identifies an attack that allows anyone on the Internet to covertly monitor mobile devices to construct detailed profiles including user identity, daily commute patterns, and travel itineraries. We identify a growing number of users vulnerable to this attack (two million and climbing), and covertly monitor over one hundred thousand of them. We demonstrate the feasibility and severity of such an attack in today's Internet. We further propose short-term and long-term defenses for the attack.

1 Introduction

The Domain Name System (DNS) is a core Internet infrastructure that maps user-friendly mnemonics to non-user-friendly IP addresses. The DNS resolves IP addresses for both *public services*[1] like Google, as well as *private services*[2] such as Alice's personal laptop. The DNS does not distinguish between the scope of the services it resolves.

As stated in RFC 4033 [1], the DNS was originally designed with the assumption that the DNS will return the same answer to any given query regardless of who may have issued the query, and that all data in the DNS is thus public. The DNS does not provide any authorization mechanism or other means of differentiating between inquirers. Indeed, DNS nameservers don't even know the IP address of the querying host. Network DoS attackers exploit this shortcoming to learn the IP address of the victim and overwhelm the victim's link to the Internet. This paper identifies an attack whereby merely learning the IP address of the victim can result in a breach of privacy as defined in [2].

An IP address encodes a wealth of information about the host. The address identifies the host's ISP (university, corporation, residential broadband ISP etc.), and the geographical location of the host to within a nearby city [3]. This information is available in public WHOIS databases, reverse DNS entries [4], and commercial databases [5].

Private services that wish to use user-friendly DNS names are consequently forced to make their existence and location visible to everyone on the Internet. Alice, for instance, may wish to run a private FTP server on her personal laptop that she can use to transfer files to and from her laptop whether she is at work or at home. Since the laptop acquires a different IP address each time she connects, Alice is forced to relearn her address each time. Alternatively, Alice can configure her laptop to update the DNS with its latest address such that she can use a stable user-friendly DNS name in her FTP client. By putting her address in the DNS for her own personal use, however, Alice reveals her geographic location to everyone on the Internet.

DynDNS [6], No-IP [7], TZO [8], and many other online services [9, 10, 11] cater to individuals that wish to register DNS names and dynamically update their IP address mapping. The most popular amongst these services has a growing user-base of two million users. As we report later, a majority of these DNS names are for mobile hosts. Common services running on these hosts include private

[1] Services available to everyone e.g. www.google.com

[2] Services available to a small group of people e.g. alice.dyndns.org

FTP and web servers, BitTorrent trackers, and webcams. Even though each of the services may enforce access control policy at the service level, simply allowing authorized users to use the DNS to resolve the network level IP address leaks private information to anyone who cares to learn it. This privacy issue was not considered one way or another in the design of the DNS. In a largely static Internet, the IP address does not divulge much private information and thus the issue has not been important in the past. In the present Internet, however, with an increasing number of mobile private hosts, the lack of confidentiality of private IP addresses published into the DNS is no longer justified.

Overall this paper makes three contributions. (1) We identify an attack that allows an attacker to covertly monitor a victim's location at any given moment, and over time build a detailed profile of the victim including the victim's identity, daily commute patterns, and trip itineraries. (2) We demonstrate the feasibility of such an attack on the Internet today by performing surveillance on over a hundred thousand users without raising any alarms and report the depth of private information gleaned. (3) We propose short-term fixes to the DNS that can be deployed today to mitigate this attack, and discuss a long-term solution for secure name resolution for private services on the Internet.

The rest of the paper is organized as follows. Section 2 presents an overview on the DNS, dynamic DNS services and related work in DNS security. Section 3 presents our attack. Section 4 reports on a covert surveillance experiment of over a hundred thousand Internet users. Section 5 discusses short-term and long-term measures to defend against such an attack. Section 6 concludes the paper and outlines future work.

2 DNS Overview and Related Work

The DNS describes the architecture and infrastructure for name resolution on the Internet [12]. The namespace is hierarchical for user-friendliness and ease of administration. Each domain or subtree of the namespace is managed by designated nameservers called the *authoritative nameservers* for that domain. An authoritative nameserver responds with the IP address (or other information in the DNS) for a DNS name in its domain when queried. If the DNS name lies outside the nameserver's domain, the server can forward the query to another nameserver that may be able to answer authoritatively (called *recursive querying*), and forward the response back to the original inquirer. Alternatively, the server can simply return the address of the next server to query (called *non-recursive querying*). A server that performs recursive querying may cache the results of any queries that it forwards and respond to subsequent queries for the same DNS name from its cache thereby improving performance and reducing the load on the authoritative nameservers. Clients OS network stacks typically implement the minimum functionality necessary to send the query to a recursive nameserver; ISPs and public services provide recursive nameservers that clients can use for all queries.

The DNS does not enforce any access control policy before name resolution. In particular, the authoritative nameserver typically does not learn the identity of the host performing the name resolution. This is because a recursive DNS query contains the address of only the nameserver performing the recursive lookup, and not the address of the host on whose behalf the query is being performed. Furthermore, when a response is cached and a query answered from the cache, the authoritative nameserver does not learn that a lookup took place. Any host on the Internet can query for the IP address of another host through a recursive nameserver without revealing the original inquirer's identity to the authoritative nameserver.

Dynamic DNS services such as DynDNS administer multiple zones within which a user can create a DNS name (e.g. alice.dynalias.org, or bob.homeip.net). The user configures a background DynDNS client application to run on their laptop and update the laptop's IP address with the service every time the IP address changes. DNS name creation and dynamic updates of the associated IP address are performed over HTTP and protected with HTTP-based authentication methods. The nameservers for these dynamic DNS services, however, are typical DNS nameservers that cannot authenticate the source of a DNS query.

In addition to resolving the DNS name to an IP address, the DNS provides inverse resolution that maps an IP address to a canonical DNS name for

that host. The canonical name for an IP address is assigned by the host's ISP. For example, the DNS name alice.dyndns.org may resolve to 192.0.2.1 in the address block allocated to Acme Inc. The reverse DNS resolution for the same IP, however, may return host-342.acme.org as the canonical name for the host. Anyone can find the ISP-assigned canonical name for the host that a DNS name points to.

Past work in securing DNS can be classified into two categories. The first category deals with protecting the DNS from outages and DoS attacks. CoDNS [13] and CoDoNS [14] use a peer-to-peer substrate for DNS queries in order to improve resiliency against failures and shield authoritative nameservers from flash crowds and DoS attacks, but otherwise allow anyone to resolve the DNS name for any host. The second category of past work deals with the integrity of DNS responses. DNSSEC [1] provides data and origin authentication of DNS data. ConfiDNS [15] provides better integrity of non-DNSSEC responses in CoDNS. Gabrilovich et al. [16] identify homograph attacks against the DNS where a user can be tricked to resolve a look-alike DNS name instead of the intended DNS name, and offer some potential solutions. None of these systems authenticate the source of the DNS query, and consequently cannot defend against the attack identified in this paper.

Proposed replacements for the DNS allow an individual to learn the victim's routing and addressing information without the victim's explicit consent. DOA [17] unconditionally resolves an endpoint identifier to a stack of addresses revealing the route to the destination. UIA [18] allows endpoint addresses to be resolved as long as the endpoint can be named; the ability to name another endpoint is transitive and is likely to be universal for global communication. Neither of the proposed approaches are designed to protect the confidentiality of a private host's address.

3 Identity Trail Attack

The attack intends to track a victim's location covertly. The 9-line attack code is listed in Figure 1. The attack consists of logging a DNS lookup (line 6) and IP address to geolocation result (line 8) every hour. The geolocation uses a public service [5] that returns the city, province, and country of the host as well as the canonical hostname for the IP address. The attack assumes that the attacker knows the DNS name of the victim's laptop; the next section discusses how an attacker can learn the names of hundreds of thousands of potential victims. The attack uses only public services such as the DNS and IP geolocation. The attack does not require superuser privileges, nor does it need to send any packets to the host being monitored. Finally, thousands of hosts can be monitored in parallel. The extent of private information that can be extracted using this attack is explored in the next section.

```
1: #!/bin/bash
2: HOST=victim.dynalias.net
3: GEO='http://www.ippages.com/simple/'
4: FLD='hostname,ip,city,state,country'
5: while sleep 3600; do
6:     IP=$(host $HOST | awk '{print $4}')
7:     date
8:     curl -s "$GEO?get=$FLD&ip=$IP"
9: done
```

Figure 1: BASH shell script tracking the location of `victim.dynalias.net` every hour.

4 Attack Validation

In order to determine the feasibility and severity of launching the described attack over the Internet today, we monitored the mobility of over a hundred thousand Internet users. The attack involved discovering names of potential victims and monitoring their IP address for extended periods of time without being detected. Finally, we analyzed the mobility patterns in the trace to profile our victims' daily commute patterns, business and personal trip itineraries and, in some cases, even the identities of the victim.

4.1 Discovering DNS Names

We discovered 36,011 potential victims through a variety of methods. In our first experiment, we targeted users of the DynDNS service by first performing Google and Yahoo! searches for all 65 DynDNS-controlled domains under which users can register DNS names. We found 4351 DNS names (far fewer than we expected) mentioned in web pages, mailing list archives, usenet posts, and other publicly searchable forums. In our second experiment, we performed a dictionary scan of four of the most popular DynDNS domains. Our

Figure 2: Tracking a user's summer road-trip through the DNS

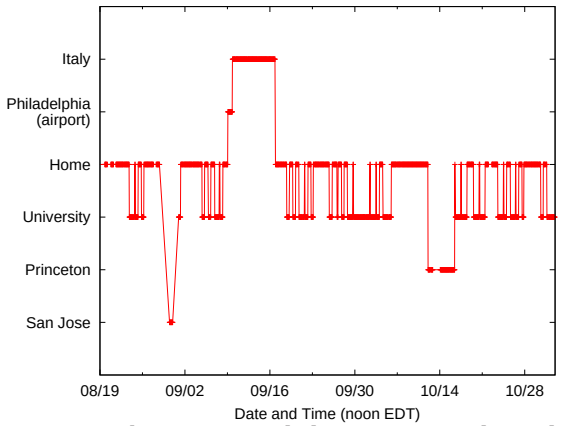

Figure 3: Tracking a user's daily commute and travel through the DNS

dictionary consisted of 24,289 combinations of common first and last names and initials. The scan successfully resolved 31,660 DNS names with a success rate of up to 39% for the dyndns.org domain; the high success rate suggest similarities with [19] where the authors find that registered userids, in many cases, match first or last names of the user. The scan was performed from 40 hosts in 5 hours and rate-limited to an aggregate of 5 packets per second to DynDNS to avoid detection. Only 9 of the hosts discovered in the dictionary scan were also returned by the online search engines suggesting almost all the names discovered by the dictionary scans are for private services. To verify this hypothesis, we performed a third experiment where we used Nmap [20] to scan a subset of the hosts discovered to determine the services provided. We discovered that 50% run HTTP servers, 21% run FTP servers, and 11% run the Windows File-and-Printer sharing service; the FTP and Windows services typically require authentication while the HTTP homepage for unauthorized users is typically devoid of content and hyperlinks.

Overall we make the following three observations. (1) The services discovered are intended for private use based on service authentication and the lack of advertisement on public Internet forums. (2) An attacker can covertly discover a large number of potential victims. (3) Poorly chosen DNS names registered by DynDNS users, in some cases, may leak the name of the user for a mobile host.

4.2 Monitoring Hosts

In our first surveillance experiment, we monitored 18,720 hosts from 7/20 to 8/8/06 and found evidence of deliberate user mobility. The monitoring load was rate-limited to 1 packet per second and did not raise any alarms at DynDNS or at the source. Figure 2 is a screenshot of a summer road-trip taken by user M as tracked by our application. M's name resolves to a Seattle IP on 7/20. It subsequently resolves to Port Angeles WA on 7/21, and continues down a southern route along the west-coast through Otis OR, Smith River CA, Garberville CA, Los Angeles CA, Los Alamos CA, and Garden Grove CA at 1–2 day intervals. M then resolves to Las Vegas, NV for 3 days starting the night of 8/2. Finally, M appears to drive north through Montana back home to Saskatoon, Canada on 8/8, which is where he was resolved to on 11/7 as well. Based on reverse DNS lookups, M logs in through local broadband ISPs except on 7/29 when M logs in through a dial-up ISP whose proxy servers are located in Reston, Virginia; fundamental limitations of geolocalization pertaining to proxies are explained in [3]. M runs a local firewall configured to filter all inbound packets. Unfortunately, we were unable to disambiguate M's real identity enough to contact him for verification.

We verified the correctness of our application in our second surveillance experiment where we tracked the authors of this paper and compared their traces to real-world data. Figure 3 plots the mobility of one of the authors from 8/18 to 11/2/06. All

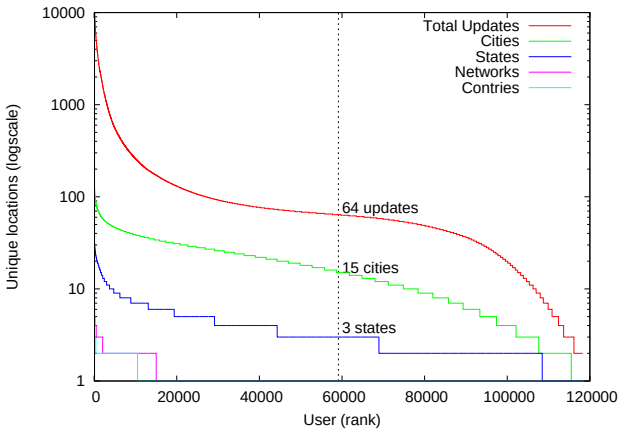

Figure 4: Tracking the mobility of 128,000 users

the information in the figure was gathered by performing geolocalization and reverse DNS lookups for the IP address. Geolocation within the United States was correct to within 100 mi, and in Italy was correct to within 250 mi. The daily commute patterns are usually correct to within one or two hours, but in some instances when the author did not turn the tracked laptop on for several hours after commuting, the trace is inaccurate. The trace, however, does accurately capture university holidays, work related trips and one airport layover. Based on the reverse DNS lookups suggesting the user's academic affiliation and trip to Cisco Systems Inc. in San Jose, and the overlap of the Italy trip with a popular data communications conference in Italy during the same period, there was enough information to narrow down the identity of the person tracked to within two people in the university CS department. Additional public information available on the department homepage yielded a unique match.

In our third surveillance experiment, we follow 128,000 DNS names for 77 days beginning 8/14/06. Anonymized update records were obtained from logs kept by DynDNS. We filter out updates where consecutive IP addresses for a user belong to the same /24 subnet, or belong to the same ISP and are geolocated to the same city in order to discount DHCP-related updates. Figure 4 plots the number of unique cities, provinces, countries and networks that mobile users were resolved to. As evident from the figure, the median number of updates across all users was 64; in the median case a user logged in from 15 cities, and 3 states over the course of our measurement. The number of users connected to more than one ISP was 15,055, while that geolocated to more than one country was 10,539. Over 91% of the users in our DynDNS-log dataset appeared to move between at least two locations.

Overall we make the following four observations. (1) An overwhelming majority of users updating their IP address on DynDNS are mobile users. (2) The IP address can accurately track a user to within a few hundred miles. (3) The extent of private information leaked over time is potentially significant enough to reconstruct trip itineraries, daily commute patterns, and in some cases, narrow down the identity of the user. (4) This entire surveillance operation can be performed covertly by any attacker on the Internet.

5 Solutions

Fundamentally, the attack illustrated in the previous two sections exploits the lack of access control policy enforcement in the DNS for name resolution. At present, any attacker on the Internet can track the mobility of any DNS-named host at any given time. In the short-term, nameservers can use heuristics to limit the scope of the attack without requiring any client modifications. In the long-term, however, the core issue of policy for name resolution of private services must be tackled.

5.1 Short-Term Defense

The DNS does not provide any mechanism for a host to specify access control policies for its own name. One fix would be to encourage users to pick hard-to-guess DNS names and restrict their dissemination to trusted parties. Another complementary fix would be for the authoritative nameserver to restrict host lookup to a white-list of recursive nameservers authorized for that host. The white-list may be set statically by the user, dynamically by the update application, or heuristically by the nameserver based on query patterns. Deploying a white-list does not require changes to the DNS protocol or client applications. This approach relies on access control of ISPs' recursive nameservers. The approach protects against attackers on networks far from the victim, but not from attackers on the same network as the victim since both would have ac-

cess to a common ISP recursive nameserver, which would be on the white-list.

A flawed way to add access control to the DNS protocol itself is to encrypt responses so only authorized users may decrypt them. Users of the private service could configure shared secrets or public keys with the authoritative nameserver. The nameserver could then encrypt the IP address using symmetric or asymmetric encryption [21, 22] based on the attack model. Performing cryptographic operations on nameservers, however, is likely to be an unacceptable bottleneck, especially for services that provide resolution for a large number of private hosts.

5.2 Long-Term Defense

Modifying the name resolution process to identify the inquirer and to directly involve the access control policies of the private host provides secure and scalable name resolution in the long run. Consider a name resolution architecture where Alice's laptop registers its IP address with a *registrar*. The registrar plays the role similar to that played by an authoritative nameservers in the DNS. In order to communicate with Alice, Bob contacts the registrar. Instead of returning Alice's IP address to Bob, however, the registrar *proxies* Bob's query to Alice's IP address and proxies Alice's responses to Bob much like in SIP [23]. The registrar conceals Alice's IP address from Bob. Alice can conduct an identification protocol over this proxied path, and reveal her IP address to Bob if he is granted access. As an optimization, the registrar can itself enforce access control policy registered by Alice in order to reduce lookup latency. While this proxy-based end-to-end name resolution architecture doesn't allow for in-network caching, it has been shown that caching the heavy-tail of DNS queries [24] for private hosts is of little value in the first place [25].

Proxy-based end-to-end secure name resolution compliments emerging Internet architectures for private hosts. In [26], the authors propose an Internet architecture where endpoints negotiate protocol stacks, configuration parameters and coordinate the opening of NAT/firewall ports given a third party to proxy connection-setup messages; the registrar used for secure name resolution can provide this proxy service for private hosts.

More research is needed, however, to better understand the privacy-performance trade-off for the DNS as well as for end-to-end secure name-resolution in the context of public and private services. For public services, it may be the case that there is no need to replace the DNS with the more heavy-weight approach as policy usually allows anyone to learn the IP address. For private services, on the other hand, it may be the case that the need for security outweighs the performance hit.

6 Conclusions and Future Work

This paper presents an attack on mobile users that dynamically register their IP address in the DNS. The attack allows any attacker on the Internet to covertly glean private information about the user including daily commute patterns and itineraries for trips. The paper demonstrates the ease with which hundreds of thousands of vulnerable users can be monitored without their knowledge. This information could easily be logged for later use should the attacker later learn or infer the identity of the user. The root cause of the attack lies in the lack of access control for DNS name resolution combined with an increasing number of mobile Internet users. This paper suggests a short-term patch to existing DNS services that restricts the scope of the attacker, and a more long-term solution that involves end-to-end secure name resolution for private services on the Internet. The proposed solutions are preliminary in that we do not have a lot of field experience with them. One goal of this paper is to draw the attention of the community to attacks on the DNS that stem from design assumptions in the original Internet architecture that are no longer valid.

Because this attack exploits the public nature of IP addresses for private hosts, this paper additionally suggests that it is appropriate to ask whether the DNS with its focus on public services should be supplemented with a secure name resolution service for private hosts.

We certainly do not answer this question—indeed we have not fully implemented the end-to-end name resolution architecture, much less experimented with it on a broad scale and studied its security provisions. We do believe, however, that secure name resolution for private services deserves debate within the research community.

References

[1] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "RFC 4033: DNS Security Introduction and Requirements," Mar. 2005.

[2] H. Nissenbaum, "Privacy as contextual integrity," *Washington Law Review*, vol. 79, no. 1, pp. 119–158, Feb. 2004.

[3] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in *Proceedings of the SIGCOMM '01*, San Diego, CA, Aug. 2001.

[4] N. Spring, R. Mahajan, and T. Anderson, "Quantifying the causes of path inflation," in *Proceedings of the SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003.

[5] The Privacy Ecosystem, "IPPages – IP Address properties of your Internet Connection." [Online]. Available: http://www.ippages.com/

[6] Dynamic Network Services, Inc., "DynDNS – A free DNS service for those with dynamic IP addresses." [Online]. Available: http://www.dyndns.com/

[7] Vitalwerks Internet Solutions, LLC., "No-IP – Dynamic DNS, Static DNS for Your Dynamic IP." [Online]. Available: http://www.no-ip.com/

[8] Tzolkin Corporation, "TZO.com – Dynamic DNS Services for your Dynamic or Static IP Address." [Online]. Available: http://www.tzo.com/

[9] Deerfield dot com, "DNS2GO – Dynamic DNS Services for your IP Address." [Online]. Available: http://www.dns2go.com/

[10] CanWeb Internet Services Ltd., "DynIP – Dynamic DNS Service." [Online]. Available: http://www.dynip.com/

[11] GravityFree, "DtDNS – Your Complete DNS Solution." [Online]. Available: http://www.dtdns.com/

[12] P. Mockapetris and K. Dunlap, "Development of the domain name system," in *Proceedings of the SIGCOMM '88*, Stanford, CA, Aug. 1988.

[13] K. Park, V. S. Pai, L. Peterson, and Z. Wang, "CoDNS: Improving DNS performance and reliability via cooperative lookups," in *Proceedings of the Sixth Symposium on Operating Systems Design and Implementation (OSDI 2004)*, San Francisco, CA, December 2004.

[14] Venugopalan Ramasubramanian and Emin Gün Sirer, "CoDoNS: The Design and Implementation of a Next Generation Name Service for the Internet," in *Proceedings of SIGCOMM'04*, Portland, OR, August 2004.

[15] Lindsey Poole and Vivek S. Pai, "ConfiDNS: Leveraging Scale and History to Improve DNS Security," in *Proceedings of WORLDS'06*, Seattle, WA, November 2006.

[16] E. Gabrilovich and A. Gontmakher, "The homograph attack," *Communications of the ACM*, vol. 45, no. 2, p. 128, Feb. 2002.

[17] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris, , and S. Shenker, "Middleboxes no longer considered harmful," in *Proceedings of the OSDI '04*, San Francisco, CA, Dec. 2004.

[18] B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea, F. Kaashoek, and R. Morris, "Persistent personal names for globally connected mobile devices," in *Proceedings of the OSDI '06*, Seattle, WA, Nov. 2004.

[19] M. Perkowitz, R. B. Doorenbos, O. Etzioni, and D. S. Weld, "Learning to understand information on the internet: An example-based approach," *Journal of Intelligent Information Systems*, vol. 8, no. 2, pp. 133–153, 2004.

[20] Gordon Lyon, "Nmap Security Scanner." [Online]. Available: http://insecure.org/nmap/

[21] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," *FOCS*, vol. 00, p. 394, 1997.

[22] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proceedings of the CRYPTO '05*, Santa Barbara, CA, Aug. 2005.

[23] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261: SIP Session Initiation Protocol," June 2002.

[24] Jaeyeon Jung and Emil Sit and Hari Balakrishnan and Robert Morris, "DNS Performance and Effectiveness of Caching," in *Proceedings of SIGCOMM Internet Measurement Workshop*, San Francisco, CA, November 2001.

[25] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications," in *Proceedings of INFOCOM'99*, New York, NY, Mar. 1999, pp. 126–134.

[26] S. Guha, Y. Takeda, and P. Francis, "NUTSS: A SIP-based Approach to UDP and TCP Network Connectivity," in *Proceedings of SIGCOMM'04 Workshops*, Portland, OR, Aug. 2004, pp. 43–48.