Using DNS as an Access Protocol for Mapping Identifiers to Locators

Oleg Ponomarev, Andrei Gurtov

Helsinki Institute for Information Technology, Finland
Helsiniki University of Technology and University of Helsinki

Abstract

Currently IP addresses serve both as end host identifiers and routing locators. There are various efforts to split these roles. We suggest using Domain Name System (DNS) as an access protocol for identifier to locator mapping.

1 Introduction

Hosts change their Internet locations more and more often due to the increasing number of mobile devices. Therefore the current practice of using IP addresses both as an identifier and a locator becomes inconvenient [Nik07].

One approach is to perform address translation at network routers as in Locator/Identifier Separation Protocol (LISP) [FFOM08]. There are four different variants of LISP and one of them is even DNS-based, but it requires tunnels between the routers and thus cooperation from the network providers.

We may employ, for example, the Host Identity Protocol (HIP) [MN06] to get permanent end-host identifiers (and other benefits), but we still need a scalable service and an infrastructure for tracking the current locations (IP addresses) of the hosts. It was proposed [Ahr07] to store the location information in the OpenDHT service [RGK$^+$05], but there are many operational issues with this approach [RCKS05].

Let us consider utilizing Domain Name System (DNS) [Moc87, RT97] for this purpose instead. We try to achieve the required scalability in a simple way, so that this could be deployed in practice.

2 DNS Model for Identifier/Locator Mapping

In this section we consider a way to store identifier to locator mappings in the DNS. We need to deal with flat global identifiers and we do not have administrative division like in usual domain names for hierarchical structure. Therefore a single organization should serve all identifiers and we would like to reduce its workload. At the same time it is desirable to avoid long time-to-live (TTL) values allowing caching to make the location data dynamic enough.

We try to solve this contradiction by introducing two levels in the system design. The first level would be served by an independent organization and would perform mapping of an identifier to a location tracking provider selected by the user (similar to root-servers or top-level domain servers). We may consider data at this level as more or less static and return long TTLs with the data. The second level is managed by the networks or some public services tracking the actual location of the end-host (the Internet connection may be provided by a different network at the moment). The records at the second level need small or zero TTL to reflect the dynamic nature of the data.

The link between these two levels can be done with CNAME resource records, so it would be understood by recursive resolvers and they would request the actual information from the second level. This would allow us to utilize existing DNS servers without any modifications to perform complete lookups and cache static information requested by the clients earlier. An example of such records for mapping 2001:0075:6099:97fa:1b0c:4322:fb26:7ea1 to 193.167.187.1 is shown in Figure 1. The same two-level design may be employed for resolving Host Identity Tag (HIT) to hostname.

Let us estimate the amount of data at the first level. If 6×10^9 end-hosts suddenly start to use our service, we have to store their random part of HIT (100 bits) and just an index of their location tracking provider (e.g., 28 bits), which gives us about 90 GB of data (or even less when compressed). This amount can be stored in random access memory of a few servers and replicated multiple times around the world similar to the current root-servers architecture [Ass07] using IP Anycast [Har02] technology. Further, even if each end-host changes its location tracking provider every hour, we will receive less than two million updates per second that could be handled by a limited pool of servers.

3 Updating Data

Since DNS allows dynamic modifications [RT97], we may allow them to the users. We just need to verify private keys possessed by HIP-aware end-hosts to check their identity. This can be achieved, for example, if we insert HIT in the SOA record, so end-hosts will perform HIP base exchange before sending a DNS UPDATE packet.

Users should be able to update data at both levels: at the first level when they decide to change their location tracking provider and at the second when they change IP addresses on the network interfaces. We propose DNS only as an access protocol (or one of the protocols) and we do not insist on using the conventional DNS server software, since the usage patterns differ from other domains.

We developed an experimental version of BIND9 [BIN04] that allows dynamic updates from hosts able to prove their HIP identities. Our modified version makes updates received from a certain host equivalent to updates authenticated by a key $\langle hexadecimal-hit-of-that-host\rangle.hit$, which allows simple configuration of the update policies.

4 Discussion

We consider as an advantage that DNS is well-known to system administrators, there is much experience with DNS-servers operations under high load, and it is rarely filtered by the firewalls. Almost every host in the Internet has access to a recursive resolver, that would perform queries on behalf of a client.

We need to store mostly static and very limited (just a network's index) data at the first level and shift the burden of serving frequently updating location data to the networks. We do not have any rendezvous servers in our design eliminating an extra point of failure.

Users may easily switch their location tracking providers (like they change operators of their DNS zone now) and it should improve the quality of the service, but we still need an independent stable service similar to DNS root-servers that the Internet community would trust and start using this architecture.

Acknowledgments

Thanks to Miika Komu and Antti Louko for their useful comments.

```
; The first level
$TTL    86400 ; 1 day
hit-to-ip.arpa.  IN SOA  soa.hit-to-ip-servers.org. ...
                 IN NS  a.hit-to-ip-servers.org.
                 IN NS  b.hit-to-ip-servers.org.

1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.7.0.0.1.0.0.2 IN CNAME
   1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.7.0.0.1.0.0.2.hit-to-ip.infrahip.net.

; The second level
$TTL 1 ; 1 second
hit-to-ip.infrahip.net.  IN SOA  felwood-hit.infrahip.net. ...
                         IN NS ns1.infrahip.net.
                         IN NS ns2.infrahip.net.

1.a.e.7.6.2.b.f.2.2.3.4.c.0.b.1.a.f.7.9.9.9.0.6.5.7.0.0.1.0.0.2 IN A 193.167.187.1
```

Figure 1: Example of records for mapping 2001:0075:6099:97fa:1b0c:4322:fb26:7ea1 to 193.167.187.1.

References

[Ahr07] Jeff Ahrenholz. HIP DHT interface: draft-ahrenholz-hiprg-dht-01, February 2007. Work in progress.

[Ass07] Root Server Technical Operations Association. http://www.root-servers.org, 2007.

[BIN04] ISC BIND. http://www.isc.org/sw/bind/, 2004.

[FFOM08] D. Farinacci, V. Fuller, D. Oran, and D. Meyer. Locator/id separation protocol: draft-farinacci-lisp-04.txt, April 2008. Work in progress. Expires: April 10, 2008.

[Har02] T. Hardie. Distributing authoritative name servers via shared unicast addresses. RFC 3258, IETF, April 2002.

[MN06] Robert Moskowitz and Pekka Nikander. Host identity protocol architecture. RFC 4423, IETF, May 2006.

[Moc87] P. Mockapetris. Domain names - implementation and specification. RFC 1035, IETF, November 1987.

[Nik07] P. Nikander. Identifier / locator separation: Exploration of the design space (ilse), February 2007. Work in progress. Expires: August 30, 2007.

[RCKS05] Sean Rhea, Byung-Gon Chun, John Kubiatowicz, and Scott Shenker. Fixing the embarrassing slowness of opendht on planetlab. December 2005.

[RGK^{+}05] Sean Rhea, Brighten Godfrey, Brad Karp, John Kubiatowicz, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Harlan Yu. OpenDHT: A public DHT service and its uses. In *Proc. of ACM SIGCOMM'05*, Philadelphia, PA, USA, August 2005. ACM Press.

[RT97] Yakov Rekhter and Susan Thomson. Dynamic updates in the domain name system (DNS UPDATE). RFC 2136, IETF, April 1997.