Internet Engineering Task Force
INTERNET-DRAFT
Oct 17, 2004
Expires Apr 18, 2005

A. Durand
Sun Microsystems, Inc.

Service Discovery using NAPTR records in DNS
<draft-durand-naptr-service-discovery-00.txt>

Status of this Memo

Abstract

This document describes a method to store and retrieve local configuration information from the DNS using NAPTR records in the reverse path DNS tree. It works for both IPv4 in-addr.arpa and IPv6 ip6.arpa.

1. Introduction

DHCP is the protocol of choice to pass configuration information and perform service discovery in well defined environment. However, defining and getting new options for DHCP is a slow process, as it requires not only standardization steps but also need to be implemented in all potential clients and server.

This memo proposes a new approach based on NAPTR records in the reverse path tree of the DNS. Defining new options for experimental purposes can be done with very little to no code change in the clients and none on the server.

This protocol can be deployed independently of DHCP and only requires the operation of a regular DNS server.

This protocol is also suitable when the administrative authority who manages the service is different from the administrative authority who manages DHCP. This is true in particular in deployment using NAT where the NAT box act as a local DHCP server and is usually not configured (or cannot be configured) to get the missing data from an upstream DHCP server.

Using the reverse path DNS tree instead of the forward path DNS tree has two major advantages:

- it does not require to reserve any label,
- it matches nicely the underlying topology.

2. NAPTR Record

2.1 NAPTR

NAPTR records are defined in RFC3403 [1]. The format of the NAPTR RR, whose DNS type code is 35, is:

```
NAPTR order        16 bits
      preference   16 bits
      flags        character-string
      service      character-string
      regexp       character-string
      replacement  domain-name
```

2.2 Defining Services

When defining a new type of configuration or a new service to be discovered, one has only to standardize the different relevant NAPTR parameters, the most important being the name of the "service" tag. For the sake of illustration, the following services are defined.

2.2.1 Isatap

The isatap router service discovery within a site can be done using the following record:

```
flags = "",
service = "isatap",
regexp =  "",
replacement = Fully Qualified Domain Name of the isatap router
```

For example:

```
flags = "",
service = "isatap",
regexp =  "",
replacement = "r21.example.com"
```

2.2.2 Tunnel Broker

The Tunnel Broker discovery within an ISP can be done using the following record:

```
flags = "",
service = "TB",
regexp =  "",
replacement = Fully Qualified Domain Name of the Tunnel Broker
```

For example:

```
flags = "",
service = "TB",
regexp =  "",
replacement = "tb.example.com"
```

2.3 Populating the DNS

The administrative authority in charge of the service to be discovered using this method will populate the reverse path DNS tree associated to the address space it controls with the relevant records.

For example, a site deploying isatap will put isatap NAPTR records for every single node of the site in the reverse path DNS tree in the form:

```
9.1.6.10.in-addr.arpa. 0 IN NAPTR 10 10 "" isatap "" r21.example.com
```

In another example, an ISP deploying a tunnel broker service will put TB NAPRT records for every single node in the reverse path DNS tree for all its customers in the form:

```
9.1.6.1.in-addr.arpa. 0 IN NAPTR 10 10 "" TB "" tb.example.com
```

The administrative entity in charge of the reverse path DNS tree can use several methods to populate the tree with NATPR records. It can use scripts to generate the zone, use wildcards or use some extension to the auto-generation methods present in most DNS servers.

When wildcard are used, they are only working on the last level of delegation. That is, if there is a zone delegated under the zone where the wildcard is placed, that zone won't be covered by the wildcard. In practice, this means putting a wildcard in every terminating zone. This is not a problem in the reverse tree, as those zones usually already exist at the subnet boundaries for the PTR records and are most of the time populated via scripts. Note that using wildcards does not prevent to populate more specific address with different NAPTR records, as long as they are on the same zone.

When a very large number of NAPTR records have to be generated, an alternative to wildcards is to have the DNS server dynamically generate the corresponding records on demand according to predefined rules.

The entire tree does not have to be populated. An ISP could, for example, only populate the records for tunnel brokers for the IP addresses of its customers who actually subscribed for the service.

Note:

When several services are to be discovered using this method, several NAPTR records would be created per node in the reverse path DNS tree representing an IP address, as many as the number of services to be discovered. It is actually possible to have several NAPRT records for the same service, the quering host would then decide which one(s) to use.

3. Discovering Services

When a client node wants to discover a given service, it creates a corresponding NAPRT DNS query for its IP address and send it as a regular DNS query. For example, the node 10.6.1.9 trying to discover its isatap router will send the following query:

```
query(type=NAPTR, node=9.1.6.10.in-addr.arpa.)
```

and then filter all the responses to retain those which service field is equal to "isatap". The fully qualified domain name (FQDN) of the isatap router to use will be contained in the replacement fields. Note: several NAPTR records could match, and then the node will end up with as many potential isatap routers to try. Mapping this FQDN to an IP address will required a supplementary DNS request for an A record for that FQDN.

A similar algorithm can be used by clients willing to discover their ISP tunnel broker. The NAPTR query would be the same, but this time, the client will filter the responses to retain those which service field is equal to "TB"

4. Operational Considerations

This methods works well when finding the name of a server is enough to complete the service discovery bootstrap phase. As most of the time the DNS data is publicly readable, no sensitive data should be place within those NAPTR records.

DNS administrator concerned about not revealing to the outside world details about their internal service configuration can use two face DNS servers to only server those NAPTR records internally.

Great care should be used when choosing a TTL value for the NAPTR value. In first approach, a value similar as the one given to DHCP lease makes sense.

As the right hand side of the NATPR record format suggested here are names, DNS servers can be easily modified to return the A record associated to those names in the additional section, the same way this is done with CNA That would speed-up the resolution process, as one query/response will be enough instead of two.

In the case where NAT and private address space are expected to be used, the administrative authority in charge of the service to be discovered can pre-populate the RFC1918 [2] private address space with the relevant NAPTR records.

5. IANA Considerations

The definition of NAPTR service fields should be standardized at IETF and recorded with IANA. A special category should be created for that. Service fields used in this memo are there only to server as examples an in no way should be used like this.

6. Security Considerations

Administrator concerned about the security of the discovery mechanism discussed here should deploy DNSsec [3]. Limiting the propagation of DNS data linked to this mechanism to "internal" customers as described in section 4 is also a good way to limit security risks. Also, as DNS data always end up leaking, one should refrain from placing sensitive information in the DNS.

7. Authors Addresses

Alain Durand
SUN Microsystems, Inc
USA
Mail: Alain.Durand@sun.com

8. Normative References

[1] RFC3403. Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database. M. Mealling. October 2002.

8. Informative References

[2] RFC1918. Address Allocation for Private Internets. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear. February 1996.

[3] RFC2535. Domain Name System Security Extensions. D. Eastlake 3rd. March 1999.

10. Copyright Statement