Network Working Group
INTERNET-DRAFT
October 25, 2002
Expires April 2002

Alain Durand
SUN Microsystems, inc.
Jun-ichiro itojun Hagino
IIJ Research Laboratory
Dave Thaler
Microsoft

Well known site local unicast addresses to communicate with recursive DNS servers <draft-ietf-ipv6-dns-discovery-07.txt>

Status of this memo

Abstract

This documents specifies 3 well known addresses to configure stub resolvers on IPv6 nodes to enable them to communicate with recursive DNS server with minimum configuration in the network and without running a discovery protocol on the end nodes. This method may be used when no other information about the addresses of recursive DNS servers is available. Implementation of stub resolvers using this as default configuration must provide a way to override this.

Copyright notice

1. Introduction

RFC 2462 [ADDRCONF] provides a way to autoconfigure nodes with one or more IPv6 address and default routes.

However, for a node to be fully operational on a network, many other parameters are needed, such as the address of a name server that offer recursive service (a.k.a. recursive DNS server), mail relays, web proxies, etc. Except for name resolution, all the other services are usually described using names, not addresses, such as smtp.myisp.net or webcache.myisp.net. For obvious bootstrapping reasons, a node needs to be configured with the IP address (and not the name) of a recursive DNS server. As IPv6 addresses look much more complex than IPv4 ones, there is some incentive to make this configuration as automatic and simple as possible.

Although it would be desirable to have all configuration parameters configured/discovered automatically, it is common practice in IPv4 today to ask the user to do manual configuration for some of them by entering server names in a configuration form. So, a solution that will allow for automatic configuration of the recursive DNS server is seen as an important step forward in the autoconfiguration story.

The intended usage scenario for this proposal is a home or enterprise network where IPv6 nodes are plugged/unplugged with minimum management and use local resources available on the network to autoconfigure. This proposal is also useful in cellular networks where all mobile devices are included within the same site.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

2. Well known addresses vs discovery

Some of the discussions in the past around DNS server discovery have been trying to characterize the solution space into stateless versus stateful or server oriented versus severless. It is not absolutely clear how much state if any needs to be kept to perform DNS server discovery, and, although the semantic differences between a router and a server are well understood from a conceptual perspective, the current implementations tend to blur the picture. In another attempt to characterize different approaches, one can look at how much intelligence a client needs to have in order to use the service.

One avenue is to ask the IPv6 node to participate in a discovery protocol, such as SLP or DHCP, learn the address of the server and send packets to this server. Another one is to configure the IPv6 node with well known addresses and let the local routing system

forward packets to the right place. This document explores this later avenue of configuration using well known addresses that does not require participation of the end node in any discovery mechanism.

3. Reserved prefix and addresses

The mechanism described here is:
- intended for ongoing use and not not just for bootstrapping
- intended to populate a stub resolver's list of available recursive servers only if that list is otherwise unpopulated
- providing reliability through redundancy using three unicast addresses.

3.1 Stub resolver configuration

This memo reserved three well known IPv6 site local addresses.

In the absence of any other information about the addresses of recursive DNS servers, IPv6 stub-resolvers MAY use any of those three IPv6 addresses in their list of candidate recursive DNS servers.

3.2 Recursive DNS servers configuration

Within sites, one or more recursive DNS server SHOULD be configured with any of those three addresses. It is RECOMMENDED that large sites deploy 3 recursive DNS servers, one for each reserved address. Small site could use only one recursive DNS server and assign the 3 addresses to it.

3.3 Rationale for the choice of three addresses

Three was chosen based on common practice in many places in the industry. While it's true that if the first one fails, that it's unlikely the second one will succeed (due to there really being no DNS server at all), using multiple addresses is important so that when ones do exist, the host can fail over to a second server more quickly than routing converges. Three servers is a compromise between extra reliability and increased complexity (maintaining additional servers, having multiple entries in the routing system, additional delays before the stub resolver returns an error,...).

Another reason to have multiple addresses is to avoid the need to use of anycast addresses to achieve reliability through redundancy. On top of the classic problems (TCP sessions, ICMP messages,...) using an anycast address would hide the real locations of the recursive DNS

servers to the stub resolver, prohibiting it to keep track of which servers are performing correctly. For this particular matter, using well known addresses is no different than configuring the stub resolver with regular addresses taken from the local site.

3.4 Implementation considerations

Stub resolver implementation MAY be configured by default using those addresses. However, implementing only the mechanism described in this memo may end up causing some interoperability problems when operating in networks where no recursive DNS server is configured with any of the well known addresses. Thus, stub resolvers MUST implement mechanisms for overriding this default, for example: manual configuration, L2 mechanisms and/or DHCPv6.

4. Routing

A solution to enable the stub resolvers to reach the recursive DNS servers is to inject host routes in the local routing system. Examples of methods for injecting host routes and a brief discussion of their fate sharing properties are presented here:

a) Manual injection of routes by a router on the same subnet.
If the node running the recursive DNS server goes down, the router may or may not be notified and keep announcing the route.

b) Running a routing protocol on the same node running the DNS resolver.
If the process running the recursive DNS server dies, the routing protocol may or may not be notified and keep announcing the route.

c) Running a routing protocol within the same process running the recursive DNS server.
If the recursive DNS server and the routing protocol run in separated threads, similar concerns as above are true.

d) Developing an "announcement" protocol that the recursive DNS server could use to advertize the host route to the nearby router. Details of such a protocol are out of scope of this document, but something similar to [MLD] is possible. However, the three first mechanisms should cover most cases.

An alternate solution is to configure a link with the well known prefix and position the three recursive DNS servers on that link. The advantage of this method is that host routes are not necessary , the well known prefix is advertised to the routing system by the routers on the link. However, in the event of a problem on the physical link, all resolvers will become unreachable.

IANA considerations for this prefix are covered in Section 6.

5. Site local versus global scope considerations

The rationales for having a site local prefix are:

-a) Using a site local prefix will ensure that the traffic to the recursive DNS servers stays local to the site. This will prevent the DNS requests from accidentally leaking out of the site. However, the local resolver can implement a policy to forward DNS resolution of non-local addresses to an external DNS resolver.

-b) Reverse DNS resolution of site local addresses is only meaningful within the site. Thus, making sure that such queries are first sent to a recursive DNS server located within the site perimeter increase their likelihood of success.

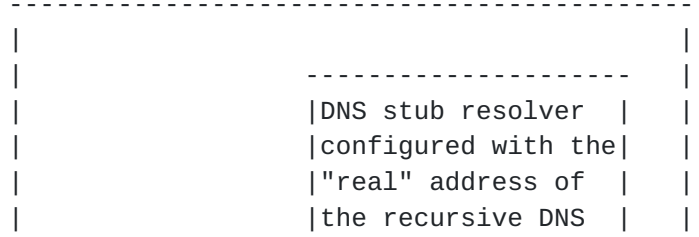
6. Examples of use

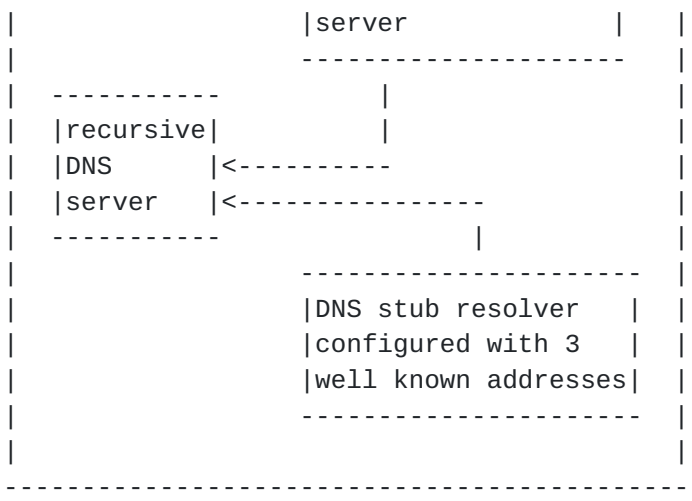
This section presents example scenarios showing how the mechanism described in this memo can co-exist with other techniques, namely manual configuration and DHCPv6 discovery.

Note: those examples are just there to illustrate some usage scenarios and in no way do they suggest any recommended practices.

6.1 Simple case, general purpose recursive DNS server

This example shows the case of a network that manages one recursive DNS server and a large number of nodes running DNS stub resolvers. The recursive DNS server is performing (and caching) all the recursive queries on behalf of the stub resolvers. The recursive DNS server is configured with an IPv6 address taken from the prefix delegated to the site and with the 3 well known addresses defined in this memo. The stub resolvers are either configured with the "real" IPv6 address of the recursive DNS server or with the well known site local unicast addresses defined in this memo.

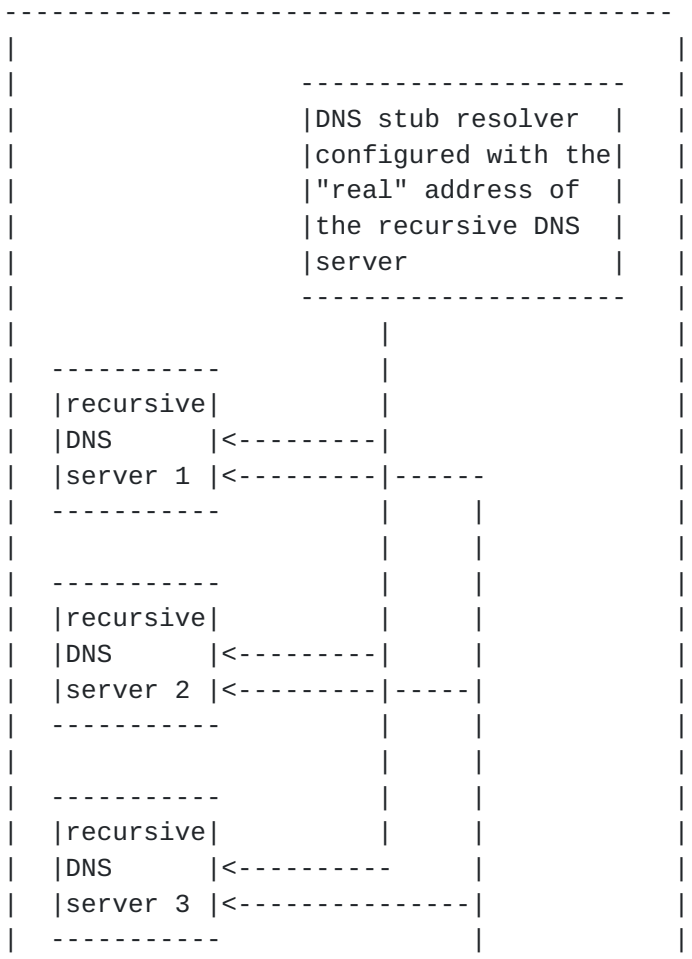


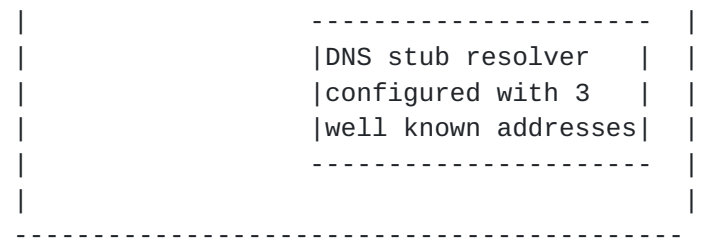


(The recursive DNS server is configured to listen both on its IPv6 address and on the well known address)

6.2 Three recursive DNS servers

This is a similar example as above, except that three recursive DNS resolvers are configured instead of just one.



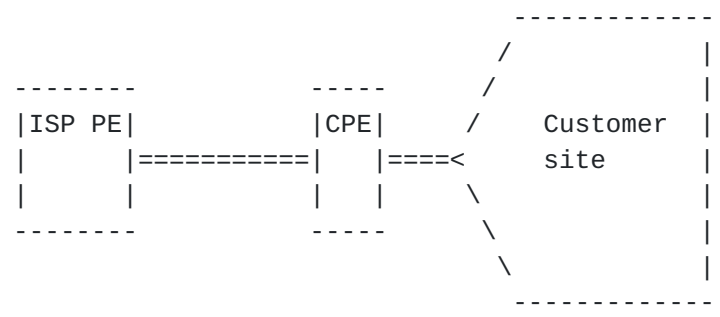


(The recursive DNS server is configured to listen both on its IPv6 address and on the well known address)

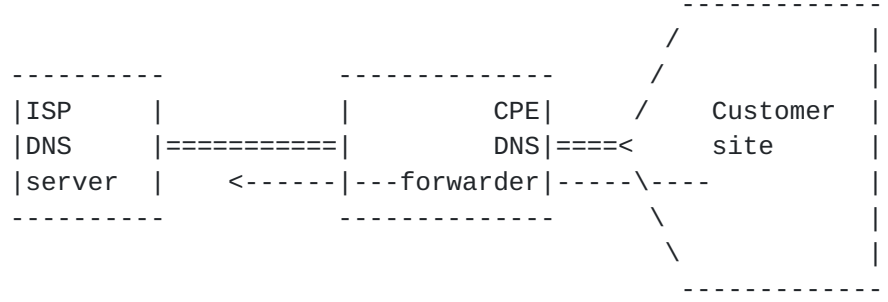
6.3 DNS forwarder

A drawback of the choice of site local scope for the reserved addresses for recursive DNS server is that, in the case of a home/small office network connected to an ISP, DNS traffic cannot be sent directly to the ISP recursive DNS server without having the ISP and all its customers share the same definition of site.

In this scenario, the home/small office network is connected to the ISP router (PE) via an edge router (CPE).



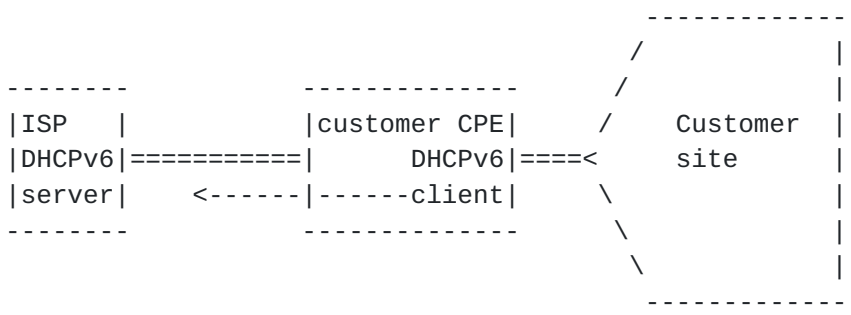
The customer router CPE could be configured on its internal interface with one of the reserved site local addresses and listen for DNS queries. It would be configured to use one (or several) of the well known site local unicast addresses within the ISP's site to send its own queries to. It would act as a DNS forwarder, forwarding queries received on its internal interface to the ISP's recursive DNS server.



In this configuration, the CPE is acting as a multi-sited router.

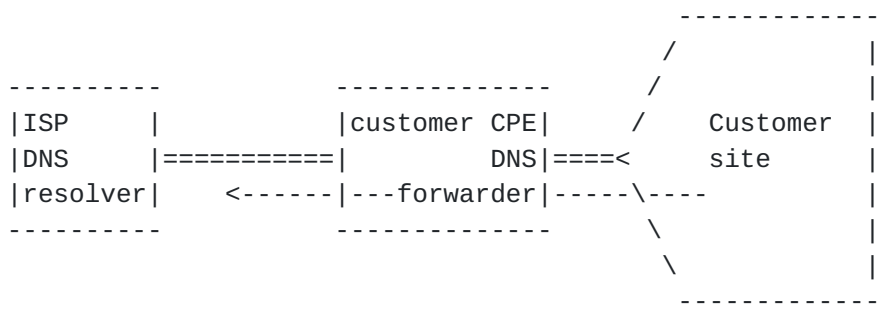
6.4 DNS forwarder with DHCPv6 interactions

In this variant scenario, DHCPv6 is be used between the PE and CPE to do prefix delegation [DELEG] and recursive DNS server discovery.

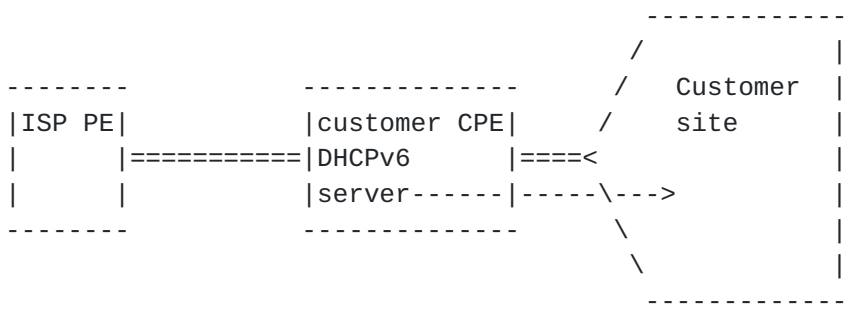


This example will show how DHCPv6 and well known site local unicast addresses cooperate to enable the internal nodes to access DNS.

The customer router CPE is configured on its internal interface with one of the reserved site local addresses and listen for DNS queries. It would act as a DNS forwarder, as in 5.2, forwarding those queries to the recursive DNS server pointed out by the ISP in the DHCPv6 exchange.

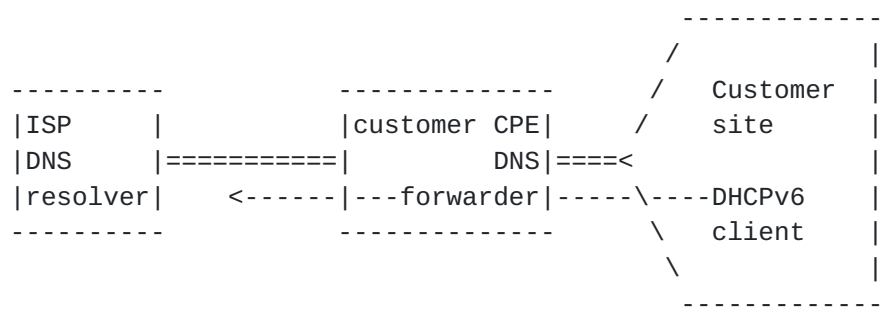


The same CPE router could also implement a local DHCPv6 server and advertizes itself as DNS forwarder.



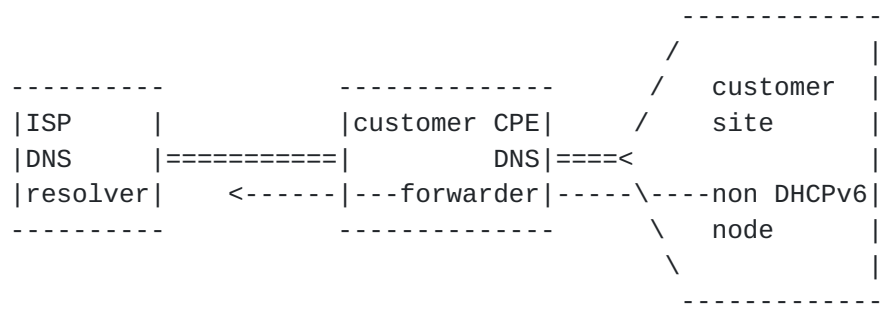
Within the site:

a) DHCPv6 aware clients use DHCPv6 to obtain the address of the DNS forwarder...



(The address of the DNS forwarder is acquired via DHCPv6.)

b) other nodes simply send their DNS request to the reserved site local addresses.



(Internal nodes use the reserved site local unicast address.)

A variant of this scenario is the CPE can decide to pass the global address of the ISP recursive DNS server in the DHCPv6 exchange with the internal nodes.

7. IANA considerations

The site local prefix fec0:0000:0000:ffff::/64 is to be reserved out of the site local fec0::/10 prefix.

The unicast addresses fec0:000:0000:ffff::1, fec0:000:0000:ffff::2 and fec0:000:0000:ffff::3 are to be reserved for recursive DNS server configuration.

All other addresses within the fec0:0000:0000:ffff::/64 are reserved for future use and are expected to be assigned only with IESG approval.

8. Security Considerations

Ensuring that queries reach a legitimate DNS server relies on the security of the IPv6 routing infrastructure. The issues here are the same as those for protecting basic IPv6 connectivity.

IPsec/IKE can be used as the well known addresses are used as unicast addresses.

The payload can be protected using standard DNS security techniques. If the client can preconfigure a well known private or public key then TSIG [TSIG] can be used with the same packets presented for the query. If this is not the case, then TSIG keys will have to be negotiated using [TKEY]. After the client has the proper key then the query can be performed.

The use of site local addresses instead of global addresses will ensure the DNS queries issued by host using this mechanism will not leak out of the site.

9. References

[KEYWORDS]
Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[ADDRCONF]
Thomson, S., and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[MLD]
Deering, S., Fenner, W., Haberman, B., "Multicast Listener Discovery (MLD) for IPv6", RFC2710, October 1999.

[TSIG]
Vixie, P., Gudmundsson, O., Eastlake, D. and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC2845, May 2000.

[TKEY]
D. Eastlake, "Secret Key Establishment for DNS (TKEY RR)", RFC2930, September 2000.

[DHCPv6]
Bound, J., Carney, M., Perkins, C., Lemon, T., Volz, B. and Droms, R. (ed.), "Dynamic host Configuration Protocol for IPv6

(DHCPv6)", draft-ietf-dhc-dhcpv6-27 (work in progress),
Februray 2002.

[DELEG]
Troan, O., Droms, R., "IPv6 Prefix Options for DHCPv6",
draft-troan-dhcpv6-opt-prefix-delegation-01.txt (work in progress),
February 2002.

10. Authors' Addresses

Alain Durand
SUN microsystems, inc.
17 Network Circle, UMPK 17-202
Menlo Park, CA 94025
Email: Alain.Durand@sun.com

Jun-ichiro itojun HAGINO
Research Laboratory, Internet Initiative Japan Inc.
Takebashi Yasuda Bldg.,
3-13 Kanda Nishiki-cho,
Chiyoda-ku, Tokyo 101-0054, JAPAN
Email: itojun@iijlab.net

Dave Thaler
Microsoft
One Microsoft Way
Redmond, CA 98052, USA
Email: dthaler@microsoft.com

11. Full Copyright Statement