Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 27, 2012

T. Savolainen
Nokia
J. Kato
NTT
T. Lemon
Nominum, Inc.

March 26, 2012

Improved DNS Server Selection for Multi-Interfaced Nodes
draft-ietf-mif-dns-server-selection-08

Abstract

A multi-interfaced node is connected to multiple networks, some of which may be utilizing private DNS namespaces. A node commonly receives DNS server configuration information from all connected networks. Some of the DNS servers may have information about namespaces other servers do not have. When a multi-interfaced node needs to utilize DNS, the node has to choose which of the servers to contact to. This document describes DHCPv4 and DHCPv6 options that can be used to configure nodes with information required to perform informed DNS server selection decisions.

Table of Contents

1. Introduction

A multi-interfaced node faces several problems a single-homed node does not encounter, as is described in [RFC6418]. This document studies in detail the problems private namespaces may cause for multi-interfaced nodes and provides a solution. The node may be implemented as a host or as a router.

We start from the premise that network operators sometimes include private namespaces in the answers they provide from DNS servers, and that those private namespaces are at least as useful to nodes as the answers from the public DNS. When private namespaces are visible for a node, some DNS servers have information other servers do not have. The node ought to be able to ask right server for the information it needs.

An example of an application that benefits from multi-interfacing is a web browser that commonly accesses many different destinations, each of which is available only on one network. The browser therefore needs to be able to communicate over different network interfaces, depending on the destination it is trying to reach.

In deployments where private namespaces are present, selection of correct route and destination and source addresses for the actual IP connection is crucial as well, as the resolved destination's IP addresses may be only usable on the network interface over which the name was resolved on. Hence solution described in this document is assumed to be commonly used in combination with tools for delivering additional routing and source and destination address selection policies.

This document is organized in the following manner. Background information about problem descriptions and example deployment scenarios are included in Section 2 and Section 3. Section 4 contains all normative descriptions for DHCP options and node behavior. Informative Section 5 illustrates behavior of a node implementing functionality described in the Section 4. Section 6 contains informational considerations about scalability. Section 7 contains normative guidelines related to creation of private namespaces. Informational Section 10 summarizes identified security considerations.

The Appendix A describes best current practices possible with tools preceding this document and that may be possibilities on networks not supporting the solution described in this document.

1.1. Requirements Language

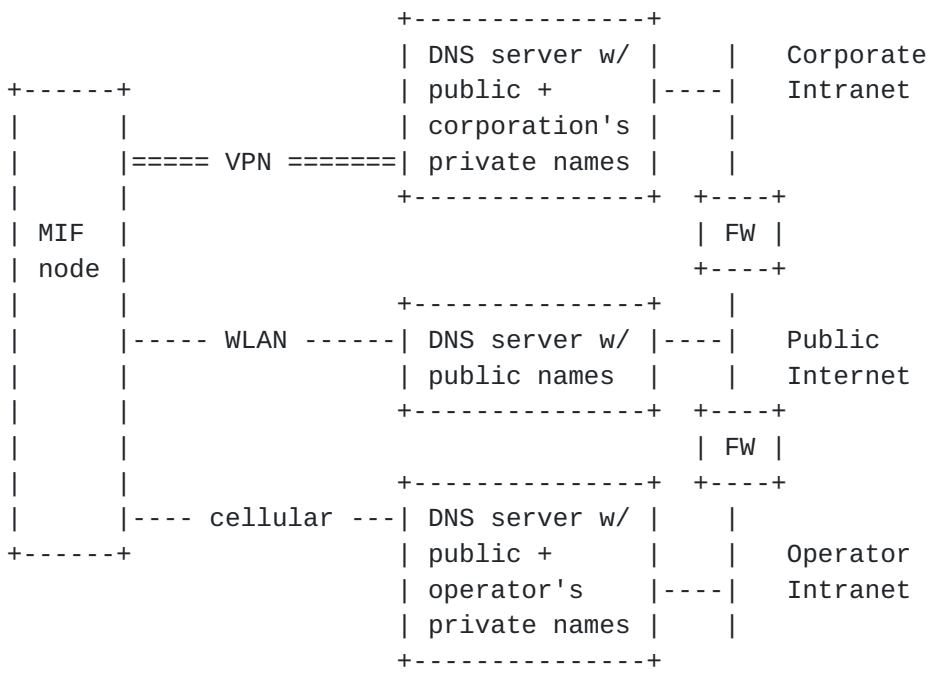
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Private namespaces and problems for multi-interfaced nodes

This section describes two node multi-interfacing related private namespace scenarios for which the procedure described in Section 4 provides a solution for. Additionally, Section 2.3 describes a problem for which this document provides only partial solution.

2.1. Fully qualified domain names with limited scopes

A multi-interfaced node may be connected to one or more networks that are using private namespaces. As an example, the node may have simultaneously open a wireless LAN (WLAN) connection to the public Internet, cellular connection to an operator network, and a virtual private network (VPN) connection to a corporate network. When an application initiates a connection establishment to an FQDN, the node needs to be able to choose the right DNS server for making a successful DNS query. This is illustrated in the figure 1. An FQDN for a public name can be resolved with any DNS server, but for an FQDN of corporation's or operator's service's private name the node needs to be able to correctly select the right DNS server for the DNS resolution, i.e. do also network interface selection already before destination's IP address is known.

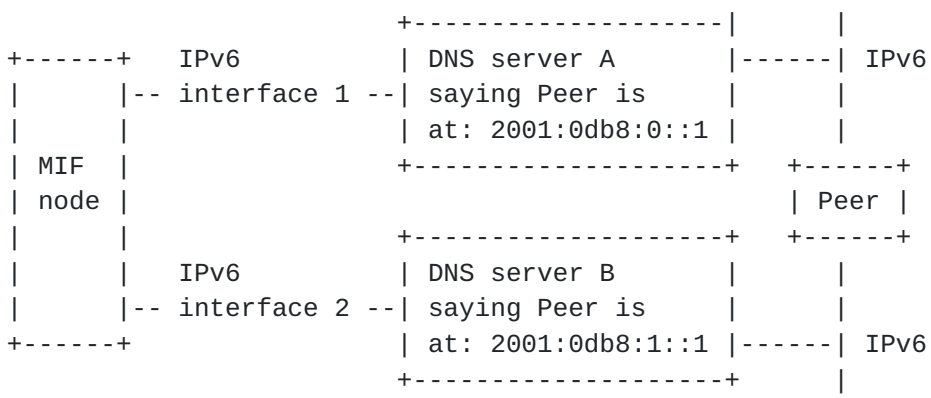


Private DNS namespaces illustrated

Figure 1

2.2. Network interface specific IP addresses

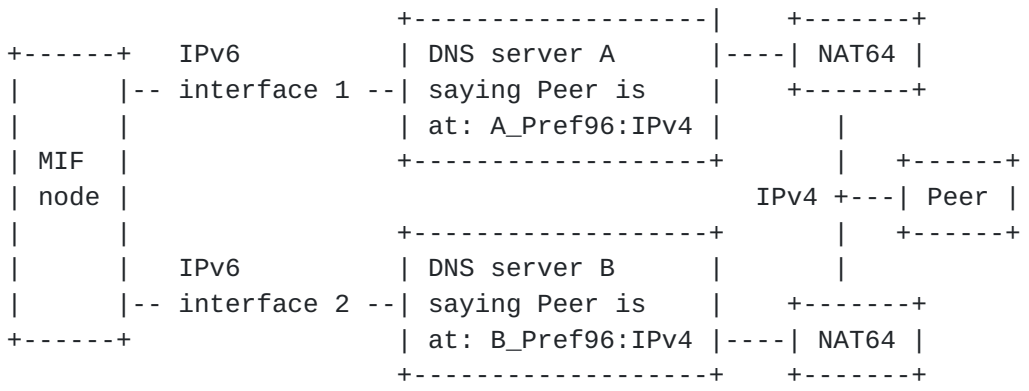
In the second problem an FQDN is valid and resolvable via different network interfaces, but to different and not necessarily globally reachable IP addresses, as is illustrated in the figure 2. Node's routing and source and destination address selection mechanism must ensure the destination's IP address is only used in combination with source IP addresses of the network interface the name was resolved on.



Private DNS namespaces and different IP addresses for an FQDN on interfaces 1 and 2.

Figure 2

Similar situation can happen with IPv6 protocol translation and AAAA record synthesis [RFC6147]. A synthetic AAAA record is guaranteed to be valid only on a network it was synthesized on. Figure 3 illustrates a scenario where the peer's IPv4 address is synthesized into different IPv6 addresses by DNS servers A and B.



AAAA synthesis results in network interface specific IPv6 addresses.

Figure 3

It is worth noting is that network specific IP addresses can cause problems also for a single-homed node, if the node retains DNS cache during movement from one network to another. After the network change, a node may have entries in its DNS cache that are no longer correct or appropriate for its new network position.

2.3. A problem not fully solved by the described solution

A more complex scenario is an FQDN, which in addition to possibly resolving into network interface specific IP addresses, identifies on different network interfaces completely different peer entities with potentially different set of service offerings. In even more complex scenario, an FQDN identifies unique peer entity, but one that provides different services on its different network interfaces. The solution described in this document is not able to tackle these higher layer issues. In fact, these problems may be solvable only by manual user intervention.

However, when DNSSEC is used, the DNSSEC validation procedure may provide assistance for selecting correct responses for some, but not all, use cases. A node may prefer to use the DNS answer that validates with the preferred trust anchor.

3. Deployment scenarios

This document has been written with three particular deployment scenarios in mind. First being a Consumer Premises Equipment (CPE) with two or more uplink VLAN connections. Second scenario involves a cellular device with two uplink Internet connections: WLAN and cellular. Third scenario is for VPNs, where use of local DNS server may be preferred for latency reasons, but corporate DNS server must be used to resolve private names used by the corporation.

3.1. CPE deployment scenario

A home gateway may have two uplink connections leading to different networks, as is described in [I-D.ietf-v6ops-ipv6-multihoming-without-ipv6nat]. In the two uplinks scenario only one uplink connection leads to the Internet, while another uplink connection leads to a private network utilizing private namespaces.

It is desirable that the CPE does not have to send DNS queries over both uplink connections, but instead CPE should send default queries to the DNS server of the interface leading to the Internet, and queries related to private namespace to the DNS server of the private network.

In this scenario the legacy hosts can be supported by deploying DNS proxy on the CPE and configuring hosts in the LAN to talk to the DNS proxy. However, updated hosts would be able to talk directly to the correct DNS servers of each uplink ISP's DNS server. It is deployment decision whether the updated hosts would be pointed to DNS

proxy or to actual DNS servers.

Depending on actual deployments, all VLAN connections might be considered as trusted.

3.2. Cellular network scenario

A cellular device may have both WLAN and cellular network interfaces up. In such a case it is often desirable to use WLAN by default, except for those connections cellular network operator wants to go over cellular interface. The cellular network may utilize private names and hence the cellular device needs to ask for those through the cellular interface.

In this scenario cellular interface can be considered trusted and WLAN oftentimes untrusted.

3.3. VPN scenario

Depending on a deployment, there may be interest to use VPN only for the traffic destined to a corporate network. The corporation may be using private namespace, and hence related DNS queries should be send over VPN to the corporate DNS server, while by default a DNS server of a local access network may be used.

In this scenario VPN interface can be considered trusted and local access network untrusted.

3.4. Dual-stack accesses

In all three scenarios one or more of the connected networks may support both IPv4 and IPv6. In such a case both or either of DHCPv4 and DHCPv6 can be used to learn DNS server selection information.

4. Improved DNS server selection

This section describes DHCP options and a procedure that a (stub / proxy) resolver may utilize for improved DNS server selection in the face of private namespaces and multiple simultaneously active network interfaces.

4.1. Procedure for prioritizing DNS servers and handling responses

A resolver SHALL build a priority list of DNS servers it will contact to depending on the query. To build the list in an optimal way, a node SHOULD ask with DHCP which DNS servers of each network interface are most likely to be able to successfully serve forward lookup

requests matching to specific domain or reverse (PTR record) lookup requests matching to specific network addresses (later referred as "network"). For security reasons the DNS server selection information MUST be used only when it is safe to do so, see Section 4.4 for details.

The node SHOULD create a node specific route for the DNS server addresses learned via DHCP. The route must point to the interface DNS server address was learned on. This is required to ensure DNS queries are sent out via the right network interface.

A resolver lacking more specific information shall assume that all information is available from any DNS server of any network interface. The DNS servers learnt by other DNS server address configuration methods MUST be handled as medium priority default servers.

When a DNS query needs to be made, the resolver SHOULD give highest precedence to the DNS servers explicitly known to serve matching domain or network. The resolver MUST take into account differences in trust levels of pieces of received DNS server selection information. The resolver MUST prefer DNS servers of trusted interfaces. The DNS servers of untrusted interfaces may be of highest priority only if trusted interfaces specifically configure low priority DNS servers. The non-exhaustive list on figure 4 illustrates how the different trust levels of received DNS server selection information SHOULD influence the DNS server selection logic.

Trustworthiness of an interface and configuration information received over the interface is implementation and/or node deployment dependent. Trust may be based on, for example, on the nature of an interface. For example, an authenticated and encrypted VPN or layer 2 connections to a trusted home network may be considered as trusted, and an unauthenticated and unencrypted connection to an unknown visited network may be considered as untrusted. In some occasions an interface may be considered trusted only if explicitly configured to be trusted.

A resolver SHOULD prioritize between equally trusted DNS servers with help of the DHCP option preference field. The resolver MUST NOT prioritize less trusted DNS servers higher than trusted, even in the case of less trusted server would apparently have additional information. In the case of all other things being equal the resolver shall make the prioritization decision based on its internal preferences.

	Information from more trusted interface A	Information from less trusted interface B	Resulting DNS server priority selection
1.	**Medium priority** default	Medium priority default	Default: A, then B
2.	**Medium priority** default	High priority default High priority specific	Default: A, then B Specific: A, then B
3.	**Low priority default**	Medium priority default	Default: B, then A
4.	**Low priority default** High priority specific	Medium priority default	Default: B, then A Specific: A, then B

Figure 4: DNS server selection in the case of different trust levels

Because DNSSEC provides cryptographic assurance of the integrity of DNS data, data that can be validated under DNSSEC is necessarily to be preferred over data that cannot be. There are two ways that a node can determine that data is valid under DNSSEC. The first is to perform DNSSEC validation itself. The second is to have a secure connection to an authenticated DNS server, and to rely on that DNS server to perform DNSSEC validation (signalling that it has done so using the AD bit). If a DNS response is not proven to be unmolested using DNSSEC, then a node cannot make a decision to prefer data from any interface with any great assurance: any response could be forged, and there is no way to detect the forgery without DNSSEC.

A node SHALL send requests to DNS servers in the order defined by the priority list until an acceptable reply is received, all replies are received, or a time out occurs. In the case of a requested name matching to a specific domain or network rule accepted from any interface, a DNSSEC-aware resolver MUST NOT proceed with a reply that cannot be validated using DNSSEC until all DNS servers on the priority list have been contacted or timed out. This protects against possible redirection attacks. In the case of the requested name not matching to any specific domain or network, first received response from any DNS server MAY be considered acceptable. A DNSSEC-aware node MAY always contact all DNS server in an attempt to receive a response that can be validated, but contacting all DNS servers is not mandated for the default case as in some deployments that would consume excess resources.

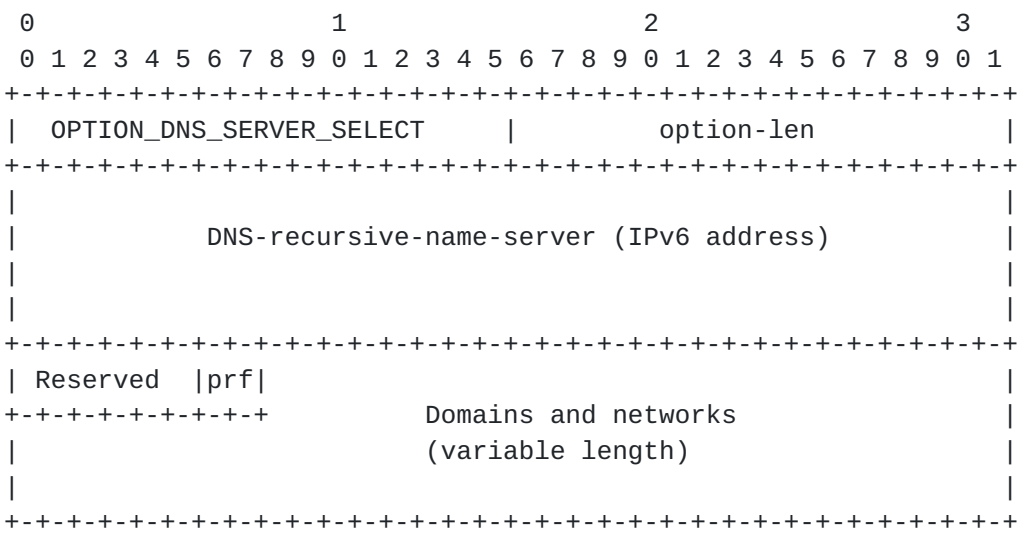
The resolver SHOULD avoid sending queries over different network interfaces in parallel as that wastes resources such as energy. The

amount of wasted energy can be significant, for example when radio interfaces has to be started just for the queries.

In the case of validated NXDOMAIN response being received from a DNS server that can provide answers for the queried name a node MUST NOT accept non-validated replies from other DNS servers (see Appendix B for considerations related to multiple trust anchors.

4.2. DNS server selection DHCPv6 option

DHCPv6 option described below can be used to inform resolvers which DNS server should be contacted when initiating forward or reverse DNS lookup procedures.



option-code: OPTION_DNS_SERVER_SELECT (TBD)

option-len: Length of the option in octets

DNS-recursive-name-server: An IPv6 address of a recursive DNS server

Reserved: Field reserved for the future. MUST be set to zero.

prf: DNS server preference, for selecting between
equally trusted DNS servers:
    01 High
    00 Medium
    11 Low
    10 Reserved

Domains and networks: The list of domains for forward DNS
lookup and networks for reverse DNS lookup the DNS server
has special knowledge about. Field MUST be encoded as
specified in Section "Representation and use of
domain names" of [RFC3315].
Special domain of "." is used to indicate
capability to resolve global names and act as a
default name server. Lack of "."
domain on the list indicates DNS server only has
information related to listed domains and networks.
Networks for reverse mapping are encoded as
defined for ip6.arpa [RFC3596] or in-addr.arpa [RFC2317].

DHCPv6 option for explicit domain configuration

Figure 5

A node SHOULD include an OPTION_ORO option in a DHCPv6 request with the OPTION_DNS_SERVER_SELECT option code to inform the DHCPv6 server about the support for the improved DNS server selection logic. DHCPv6 server receiving this information MAY then choose to provision DNS server addresses only with the OPTION_DNS_SERVER_SELECT.

The OPTION_DNS_SERVER_SELECT contains one or more domains the related DNS server has particular knowledge of. The option can occur multiple times in a single DHCPv6 message, if multiple DNS servers are to be configured.

IPv6 networks should cover all the domains configured in this option. Networks should be as long as possible to avoid potential collision with information received on other option instances or with options received from DHCPv6 servers of other network interfaces. Overlapping IPv6 networks are interpreted so that the resolver can use any of the DNS servers for queries matching the networks.

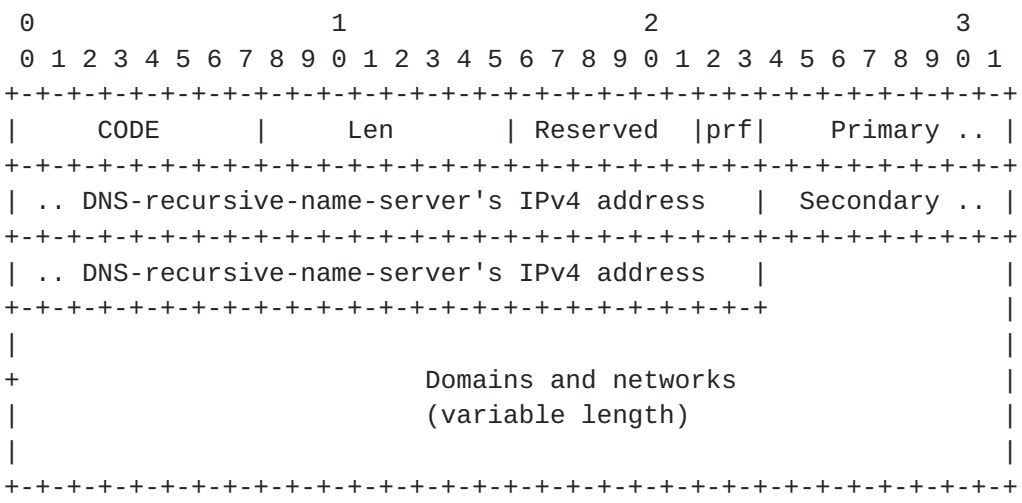
If the OPTION_DNS_SERVER_SELECT contains a DNS server address already learned from other DHCPv6 servers of the same network, and contains new domains or networks, the node SHOULD append the information to the information received earlier. The node MUST NOT remove previously obtained information. However, the node SHOULD NOT extent lifetime of earlier information either. In the case of conflicting DNS server address is learned from less trusted interface, the node MUST ignore the option.

As the DNS options of [RFC3646], the OPTION_DNS_SERVER_SELECT option MUST NOT appear in any other than the following DHCPv6 messages: Solicit, Advertise, Request, Renew, Rebind, Information-Request, and Reply.

The information conveyed in OPTION_DNS_SERVER_SELECT is considered valid until changed or refreshed by general events that trigger DHCPv6 action. In the event that it is desired for the client to request a refresh of the information, use of generic DHCPv6 Information Refresh Time Option, as specified in [RFC4242] is RECOMMENDED.

4.3. DNS server selection DHCPv4 option

DHCPv4 option described below can be used to inform resolvers which DNS server should be contacted when initiating forward or reverse DNS lookup procedures.



option-code: OPTION_DNS_SERVER_SELECT (TBD)

option-len: Length of the option in octets

Reserved: Field reserved for the future. MUST be set to zero.

prf: DNS servers preference, for selecting between
equally trusted DNS servers:

- 01 High
- 00 Medium
- 11 Low
- 10 Reserved

Primary DNS-recursive-name-server's IPv4 address: Address of
a primary recursive DNS server

Secondary DNS-recursive-name-server's IPv4 address: Address of
a secondary recursive DNS server or 0.0.0.0 if
not configured.

Domains and networks: The list of domains for forward DNS lookup
and networks for reverse DNS lookup the DNS servers
have special knowledge about. Field MUST be encoded as
specified in Section "Representation and use of
domain names" of [RFC3315].
Special domain of "." is used to indicate
capability to resolve global names and act as
default name servers. Lack of "."
domain on the list indicates DNS servers only have
information related to listed domains and networks.
Networks for reverse mapping are encoded as
defined for ip6.arpa [RFC3596] or in-addr.arpa [RFC2317].

DHCPv4 option for explicit domain configuration

Figure 6

The OPTION_DNS_SERVER_SELECT contains one or more domains the primary and secondary DNS servers have particular knowledge of. If the length of the domains and networks field causes option length to exceed the maximum permissible for a single option (255 octets), then multiple options MAY be used, as described in "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)" [RFC3396]. When multiple options are present, the data portions of all option instances are concatenated together.

If the OPTION_DNS_SERVER_SELECT contains a DNS server address already learned from other DHCPv4 servers of the same network, and contains new domains or networks, the node SHOULD append the information to the information received earlier. The node MUST NOT remove previously obtained information. However, the node SHOULD NOT extent lifetime of earlier information either. In the case of conflicting DNS server address is learned from less trusted interface, the node MUST ignore the option.

4.4. Limitations on use

Use of OPTION_DNS_SERVER_SELECT is ideal in the following environments, but SHOULD NOT be enabled by default otherwise:

1. The server selection option is delivered across a secure, trusted channel.

2. The server selection option is not secured, but the client on a node does DNSSEC validation.

3. The server selection option is not secured, the resolver does DNSSEC validation, and the client communicates with the resolver configured with server selection option over a secure, trusted channel.

4. The DNS server IP address that is being recommended in the server selection option is known and trusted by the client; that is, the server selection option serves not to introduce the client to a new server, but rather to inform it that a server it has already been configured to trust is available to it for resolving certain domains.

4.5. Coexistence with RFC3646 and RFC2132

The OPTION_DNS_SERVER_SELECT is designed to coexist with DHCPv6 OPTION_DNS_SERVERS defined in [RFC3646] and DHCPv4 Domain Name Server

Option defined in [RFC2132]. The DNS servers configured via OPTION_DNS_SERVERS or Domain Name Server Option MUST be considered as default name servers with medium preference. When both options are received from the same network interface and the OPTION_DNS_SERVER_SELECT contains default DNS server address, the resolver MUST make the decision which one to prefer based on DNS preference field value. If OPTION_DNS_SERVER_SELECT defines medium preference then DNS server from OPTION_DNS_SERVER_SELECT SHALL be selected.

If OPTION_DNS_SERVERS or Domain Name Server Option and OPTION_DNS_SERVER_SELECT contain the same DNS server(s) IP address(es), a node SHALL add the same address of a DNS server to the server list only once.

If a node had indicated support for OPTION_DNS_SERVER_SELECT in DHCPv6 request, the DHCPv6 server may choose to omit sending of OPTION_DNS_SERVERS. This enables offloading use case where network administrator wishes to only advertise low priority default DNS servers.

4.6. Considerations on follow-up queries

Any follow-up queries that are performed on the basis of an answer received on an interface MUST continue to use the same interface, irrespective of the DNS server selection settings on any other interface. For example, if a node receives a reply with a canonical name (CNAME) or delegation name (DNAME) the follow-up queries MUST be sent to DNS server(s) of the same interface, or to same DNS server, irrespectively of the FQDN received. Otherwise referrals may fail.

5. Example of a node behavior

Figure 7 illustrates node behavior when it initializes two network interfaces for parallel usage and learns domain and network information from DHCPv6 servers.

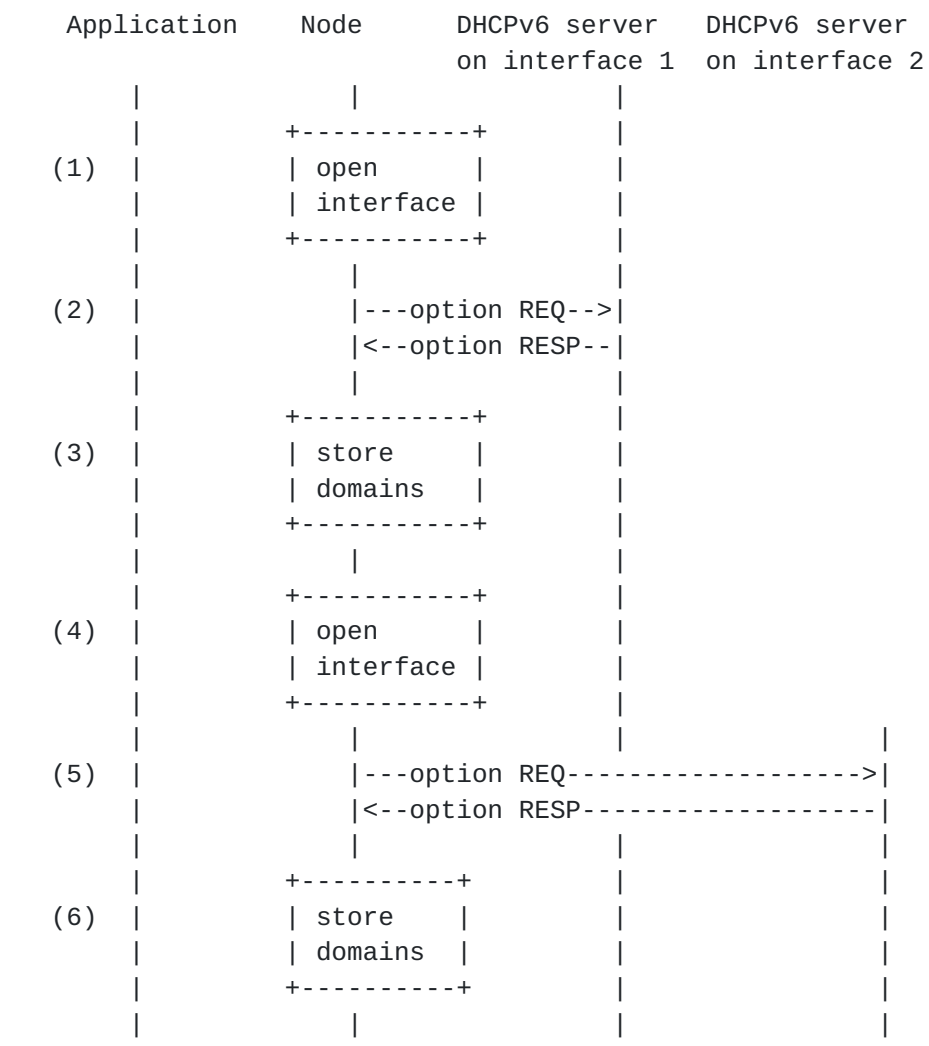


Illustration of learning domains

Figure 7

Flow explanations:

1. A node opens its first network interface

2. The node obtains domain 'domain1.example.com' and IPv6 network '0.8.b.d.0.1.0.0.2.ip6.arpa' for the new interface 1 from DHCPv6 server

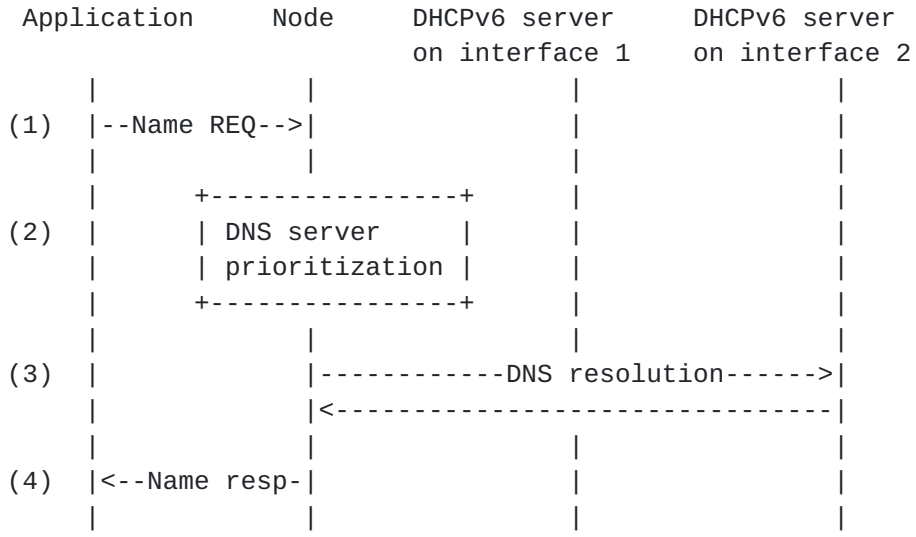
3. The node stores the learned domains and IPv6 networks for later use

4. The node opens its second network interface 2

5. The node obtains domain 'domain2.example.com' and IPv6 network information, say '1.8.b.d.0.1.0.0.2.ip6.arpa' for the new interface 2 from DHCPv6 server

6. The node stores the learned domains and networks for later use

Figure 8 below illustrates how a resolver uses the learned domain information. Network information use for reverse lookups is not illustrated, but that would go as the figure 7 example.



Example on choosing interface based on domain

Figure 8

Flow explanations:

1. An application makes a request for resolving an FQDN, e.g. 'private.domain2.example.com'

2. A node creates list of DNS servers to contact to and uses configured DNS server information and stored domain information on prioritization decisions.

3. The node has chosen interface 2, as from DHCPv6 it was learned earlier that the interface 2 has domain 'domain2.example.com'. The node then resolves the requested name using interface 2's DNS server to an IPv6 address

4. The node replies to application with the resolved IPv6 address

6. Scalability considerations

The size limitations of DHCP messages limit the number of domains and networks that can be carried in configuration options. Including the domains and networks in a DHCP option is best suited for deployments where relatively few carefully selected domains and networks are adequate.

7. Considerations for network administrators

Network administrators deploying private namespaces should assist advanced nodes in their DNS server selection process by providing information described within this document.

Private namespaces MUST be globally unique in order to keep DNS unambiguous and henceforth avoiding caching related issues and destination selection problems (see Section 2.3). Exceptions to this rule are domains utilized for local name resolution (such as .local).

Private namespaces MUST only consist of subdomains of domains for which the relevant operator provides authoritative name service. Thus, subdomains of example.com are permitted in the private namespace served by an operator's DNS servers only if the same operator provides an SOA record for example.com.

To counter against attacks against private namespaces, administrators utilizing this tool SHOULD deploy DNSSEC for their zone.

8. Acknowledgements

The author would like to thank following people for their valuable feedback and improvement ideas: Mark Andrews, Jari Arkko, Marcelo Bagnulo, Brian Carpenter, Stuart Cheshire, Lars Eggert, Tomohiro Fujisaki, Peter Koch, Suresh Krishnan, Murray Kucherawy, Edward Lewis, Kurtis Lindqvist, Arifumi Matsumoto, Erik Nordmark, Steve Padgett, Fabien Rapin, Matthew Ryan, Dave Thaler, Margaret Wasserman, Dan Wing, and Dec Wojciech. Ted Lemon and Julien Laganier receive special thanks for their contributions to security considerations.

This document was prepared using xml2rfc template and the related web-tool.

9. IANA Considerations

This memo requests IANA to assign two new option codes. First option

code is requested to be assigned for DHCPv4 DNS Server Selection option (TBD) from the DHCP option code space defined in section "New DHCP option codes" of RFC 2939. Second option code is requested to be assigned to the DHCPv6 DNS Server Selection option (TBD) from the DHCPv6 option code space defined in section "IANA Considerations" of RFC 3315.

10. Security Considerations

10.1. Attack vectors

It is possible that attackers might try to utilize OPTION_DNS_SERVER_SELECT option to redirect some or all DNS queries sent by a resolver to undesired destinations. The purpose of an attack might be denial-of-service, preparation for man-in-the-middle attack, or something akin.

Attackers might try to lure specific traffic by advertising domains and networks from very small to very large scope or simply by trying to place attacker's DNS server as the highest priority default server.

The best countermeasure for nodes is to implement validating DNSSEC aware resolvers. Trusting on validation done by a DNS server is a possibility only if a node trusts the DNS server and can use a secure channel for DNS messages.

10.2. Trust levels of network interfaces

Decision on trust levels of network interfaces depends very much on deployment scenario and types of network interfaces. For example, unmanaged WLAN may be considered less trustworthy than managed cellular or VPN connections. An implementation may not be able to determine trust levels without explicit configuration provided by user or administrator. Therefore, for example, an implementation may not by default trust configuration received even over VPN interfaces.

The decision on levels of trust may be made by implementation, by node administrators, or for example by other standards defining organizations as part of system design work.

10.3. Importance of following the algorithm

The Section 4 uses normative language for describing node internal behavior in order to ensure nodes would not open up new attack vectors by accidental use of DNS server selection options. During the standards work consensus was that it is safer to not to enable

this option always by default, but only when deemed useful and safe.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC2317] Eidnes, H., de Groot, G., and P. Vixie, "Classless IN-ADDR.ARPA delegation", BCP 20, RFC 2317, March 1998.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, November 2005.

11.2. Informative References

[I-D.ietf-v6ops-ipv6-multihoming-without-ipv6nat] Matsushima, S., Okimoto, T., Troan, O., Miles, D., and D. Wing, "IPv6 Multihoming without Network Address Translation", draft-ietf-v6ops-ipv6-multihoming-without-ipv6nat-04 (work in progress), February 2012.

[RFC3397] Aboba, B. and S. Cheshire, "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", RFC 3397, November 2002.

[RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

[RFC6418] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, November 2011.

Appendix A. Possible alternative practices for DNS server selection

On some private namespace deployments explicit policies for DNS server selection are not available. This section describes ways for nodes to mitigate the problem by sending wide-spread queries and by utilizing possibly existing indirect information elements as hints.

A.1. Sending queries out on multiple interfaces in parallel

A possible current practice is to send DNS queries out of multiple interfaces and pick up the best out of the received responses. A node SHOULD implement DNSSEC in order to be able to reject responses that cannot be validated. Selection between legitimate answers is implementation specific, but replies from trusted servers should be preferred.

A downside of this approach is increased consumption of resources. Namely power consumption if an interface, e.g. wireless, has to be brought up just for the DNS query that could have been resolved also via cheaper interface. Also load on DNS servers is increased. However, local caching of results mitigates these problems, and a node might also learn interfaces that seem to be able to provide 'better' responses than other and prefer those - without forgetting fallback required for cases when node is connected to more than one network using private namespaces.

A.2. Search list option for DNS forward lookup decisions

A node can learn the special domains of attached network interfaces from IPv6 Router Advertisement DNS Search List Option [RFC6106] or DHCP search list options; DHCPv4 Domain Search Option number 119 [RFC3397] and DHCPv6 Domain Search List Option number 24 [RFC3646]. The node behavior is very similar as is illustrated in the example at Section 5. While these options are not intended to be used in DNS server selection, they may be used by the nodes as hints for smarter DNS server prioritization purposes in order to increase likelihood of fast and successful DNS query.

Overloading of existing DNS search list options is not without problems: resolvers would obviously use the domains learned from search lists also for name resolution purposes. This may not be a problem in deployments where DNS search list options contain few domains like 'example.com, private.example.com', but can become a problem if many domains are configured.

A.3. More specific routes for reverse lookup decision

[RFC4191] defines how more specific routes can be provisioned for nodes. This information is not intended to be used in DNS server selection, but nevertheless a node can use this information as a hint about which interface would be best to try first for reverse lookup procedures. A DNS server configured via the same interface as more specific routes is more likely capable to answer reverse lookup questions correctly than DNS server of an another interface. The likelihood of success is possibly higher if DNS server address is received in the same RA [RFC6106] as the more specific route information.

A.4. Longest matching prefix for reverse lookup decision

A node may utilize the longest matching prefix approach when deciding which DNS server to contact for reverse lookup purposes. Namely, the node may send a DNS query to a DNS server learned over an interface having longest matching prefix to the address being queried. This approach can help in cases where ULA [RFC4193] addresses are used and when the queried address belongs to a node or server within the same network (for example intranet).

Appendix B. DNSSEC and multiple answers validating with different trust anchors

When validating DNS answers with DNSSEC, a validator might order the list of trust anchors it uses to start validation chains, in terms of

the node's preferences for those trust anchors. A node could use this ability in order to select among alternative DNS results from different interfaces. Suppose that a node has a trust anchor for the public DNS root, and also has a special-purpose trust anchor for example.com. An answer is received on interface i1 for www.example.com, and the validation for that succeeds by using the public trust anchor. Also, an answer is received on interface i2 for www.example.com, and the validation for that succeeds by using the trust anchor for example.com. In this case, the node has evidence for relying on i2 for answers in the example.com zone.

Authors' Addresses

Teemu Savolainen
Nokia
Hermiankatu 12 D
TAMPERE, FI-33720
FINLAND

Email: teemu.savolainen@nokia.com

Jun-ya Kato
NTT
9-11, Midori-Cho 3-Chome Musashino-Shi
TOKYO, 180-8585
JAPAN

Email: kato@syce.net

Ted Lemon
Nominum, Inc.
2000 Seaport Boulevard
Redwood City, CA 94063
USA

Phone: +1 650 381 6000
Email: Ted.Lemon@nominum.com