

IPNG Working Group
INTERNET-DRAFT
Expires September 2001

DNS Discovery Design Team
Dave Thaler, Editor
July 12, 2001

Analysis of DNS Server Discovery Mechanisms for IPv6
<[draft-ietf-ipngwg-dns-discovery-analysis-00.txt](#)>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

Draft

DDDT Report

January 2002

There are any number of ways that IPv6 hosts can discover information required to enable name resolution, in the absence of a DHCP server. This document discusses the issues and provides a taxonomy of possible solutions, and evaluates them against various design criteria. Finally, it provides recommendations as input to the standards process.

1. Introduction

The function of name-to-address resolution (or vice versa) in IP is performed by the Domain Name Service (DNS) [RFC1034, [RFC1035](#)]. Using DNS requires that at least one DNS Server be known and reachable by a device desiring to resolution.

There is also underway, known as Multicast DNS (mDNS) [[MDNS](#)], on resolving names on the link in the absence of a DNS Server. In a managed environment with DNS Servers, mDNS is typically disabled (via the domain search path). As a result, it is required that a device be able to discover whether DNS Servers are available and discover its search path. Thus, the mechanisms analyzed in this report do not conflict with, but actually support the mDNS work.

In the absence of a DHCP server, the current IPv6 protocol suite does not yet provide a mechanism to discover DNS servers or search paths. To solve this problem, a design team was chartered by the IPNG Working Group to investigate possible solutions and provide a recommendation as input to the working group.

This document summarizes the approaches investigated, and provides an analysis of each, and describes its recommendations. The design team participants are listed in the Authors' Addresses section below.

2. Requirements

For a device to effectively resolve names, and potentially allow resolution of its name to be performed, the following information is required:

- o One or more addresses of DNS servers. If a list is obtained, a client need only rediscover DNS servers if all addresses in the

list are unreachable. However, if a list is obtained from a single point, such as one of the DNS servers, then a

Expires January 2002

[Page 2]

Draft

DDDT Report

January 2002

requirement exists that the list of servers be up-to-date and easily maintainable.

- o Domain name

- o Search path. It is currently common practice, for the search path to be computed by a device based on its domain name obtained. However, a DHCP option [[DOMSEARCH](#)] is being proposed in the DHC WG, and so search path configuration is likely to be a requirement in general.

It is a further requirement that the above information be obtained without using a DHCP server.

Automatic configuration of DNS servers, if no DNS servers exist within the IPv6 site, is not a requirement. However, it is expected that there may be devices on site boundaries which will want to relay messages containing the above types of DNS information across site boundaries. One such likely scenario would be a home gateway device, where a home is its own site, wanting to allow devices in the home to discover and use an external DNS server, provided by an ISP.

[3.](#) Criteria for Evaluation

Each mechanism can be evaluated against a number of criteria:

Scalability

Is the mechanism scalable to a huge number of devices within a site?

Security

Does the mechanism support authentication? What pre-configuration is assumed?

Time to Deploy

Can the mechanism be deployed immediately? What devices need to have software updated to deploy the mechanism? What devices need to have additional configuration done with existing software?

Business Motivation

Do the parties required to implement any new code have a business motivation to do so, in preference to other options?

Expires January 2002

[Page 3]

Draft

DDDT Report

January 2002

Do the parties required to deploy any new devices or software have a business motivation to do so, in preference to other options?

Standardization

Does the mechanism require anything new to be standardized? If so, is the standardization within the realm of another Working Group?

Fate Sharing

If deployed, does the mechanism introduce dependencies on other devices, protocols, or processes that would not normally be required to perform name resolution?

Convergence Time

Upon the failure of a DNS server, router, or link, how long does it take for the mechanism to converge?

Scenarios

Does the mechanism work in all scenarios? Scenarios of note include:

- o Isolated link with a DNS server but no routers
- o Site with routers, but no multicast routing enabled
- o Hosts connected over an NBMA link

[4.](#) Taxonomy

Mechanisms can be categorized by their choice of transport mechanism (e.g., are the messages multicast or unicast?), and by their choice of packet format.

Sections [5](#) and [6](#) cover these two axes, and describe the set of mechanisms evaluated.

[5](#). Transport Mechanisms

[5.1](#). Anycast for DNS server discovery only

This method is based on the use of a well-known site-scoped anycast address which is used during DNS server discovery only.

Expires January 2002

[Page 4]

Draft

DDDT Report

January 2002

We assume that IANA defines a well-known site-scoped anycast address, which can be assigned one or more servers. For optimal fate-sharing, we assume that these servers are DNS servers. The aim is to provide a mechanism that allows DNS servers assigned the well-known anycast address to be reachable within a site.

The proposal is based on a two-step discovery process. First, a device sends, using a connection-less protocol, a server-discovery message with the anycast address as a destination address. The message will eventually reach a topologically closest server with the anycast address. The server will send a reply with a unicast address as the source address, containing a list of unicast addresses of valid DNS servers, plus the domain name and search path. Thereafter, all communication for name resolution is done using one of the unicast addresses in the list obtained. Only if all addresses in the list are unreachable does the device need to repeat the discovery process.

Obviously, this requires that all IPv6 devices requiring DNS server discovery in the absence of a DHCP server must implement this mechanism.

There are three immediate deployment options:

- a) Run the servers on routers, and configure them to inject host routes for the anycast address into the site's routing infrastructure.
- b) Run a routing protocol on the servers, and configure them to inject host routes for the anycast address into the site's routing infrastructure. Note that this requires that a server and its router(s) must run the same routing protocol, at least for communication between the router(s) and the server(s) on the link. Note that, however, a server does not need to participate fully in the routing protocol, it only needs to be able to inject routes. For example, RIPng [[RIPNG](#)], configured for outbound advertisements only, would be sufficient.
- c) Run multiple servers on the same link(s), and configure their local router(s) to inject host routes for the anycast address into the site's routing infrastructure. Running multiple servers on the same link provides robustness to the failure of a server, while routing provides robustness to the loss of routers and other links. There may still be some failures,

Expires January 2002

[Page 5]

Draft

DDDT Report

January 2002

however, such as a unidirectional failure of the router's interface, which are not handled by this option.

If new code can be added to the router, a fourth option is also possible:

- d) Using Neighbor Discovery to track whether servers are present. In this case a router will be configured to solicit certain Site-scoped anycast addresses when booting. This can also be done periodically. Upon receiving a reply, the router will have the true address of the server in its Neighbor Cache and can inject a host route into the system for this Site-scoped anycast address. A variant of this option, which would not require configuring the routers with the addresses to solicit, would be to combine it with running a routing protocol on the servers. The router could then solicit whatever addresses were advertised in host routes, and take advantage of neighbor unreachability detection to quickly expire host routes.

Routers should have this feature and allow configuration to enable or disable it. Also the Site-scoped addresses can be configurable when enabling this feature. This is to ensure that network administrators can add new Site-scoped addresses without the need for new software. On some links, it may be required to not allow hosts to announce these services. Hence soliciting anycast addresses should not be enabled on those links.

It should be noted that this option does not require any protocol changes to ND, as it relies on the allocation of Well-known site-scoped anycast addresses to the hosts.

For a longer-term solution, a mechanism for communicating anycast group joins to routers is desired. It is expected that this would be done as an extension to the Multicast Listener Discovery [MLD] protocol, and the sockets API for joining multicast groups. There is now work in progress [[HOST-ANYCAST](#)] in this regard.

More details on generic anycast issues are available in [ITOJUN-ANYCAST].

[5.1.1.](#) Evaluation

Scalability

The use of anycast can scale up to an arbitrarily large number of devices within the site, since to perform DNS server discovery, devices will only contact their closest server. As a result, scalability can be achieved by adding more servers as the number of devices increases. Since all traffic is unicast traffic, no extra bandwidth is consumed on links other than those between a device and its closest server.

A large number of servers can also be supported easily, if desired, since only a single server is contacted to obtain a

list of DNS servers, and the list obtained need not contain all DNS servers in the site.

Security

The use of anycast is vulnerable to the same attacks as unicast. This mechanism can best be authenticated by having the content signed via a content-specific mechanism.

It may be possible to use IPsec to authenticate the discovery messages, but IKE cannot be used for anycast transmission, as [RFC 2373](#) does not allow an anycast address to be used as the source address of packets. As a result, barring future research and development of other key distribution protocols, using IPsec instead of a content-specific mechanism would require manual keying.

In addition to securing discovery messages, it is also necessary to secure the mechanism used to communicate anycast joins from DNS servers to routers. Without such authentication, the possibility would exist for a rogue server to redirect traffic to itself, and away from valid servers.

In option (a), this is easy since the server is on the same machine as the router. Likewise, in option (c), joins are done by manual configuration which is easily securable. Option (b), however, requires utilizing authentication mechanisms available with current routing protocols (e.g., MD5 with RIPng [[RIPMD5](#)]).

In option (d), only Neighbor Discovery messages need to be secured. However, authenticating Neighbor Discovery messages is needed regardless of the option chosen, even if only to

secure discovery by devices on the same link.

Time to Deploy

Anycast discovery can be deployed as soon as clients are available, using any of the immediate-term options listed above, alone or in any combination, at the choice of each site

administrator, without any need to coordinate with others.

For immediate option (a), one must update one or more routers to run a server, if none already do.

For immediate option (b), one must update one or more DNS servers to run an IPv6 routing protocol which can talk to its local router(s).

For immediate option (c), one must ensure that multiple servers are deployed on the same link(s). One then configures its local routers with a static host route for the anycast address.

To deploy option (d), routers on servers' links need to be able to inject routes based on responses received from ND solicitations. This would require SW upgrades for those existing routers.

In all four options, one must finally configure the servers with the well-known anycast address. No configuration is needed on other routers or devices, so configuration is confined to a small number of devices. It is expected that at least one of the above options will be acceptable to IPv6 site administrators for the immediate future.

Business Motivation

The primary business motivation is that it can be deployed immediately with a minimum of effort and cost, compared to other options. As time progresses, the incentive to upgrade to either option (d) or a dynamic joining protocol is that the site would not be limited to the three basic options.

Standardization

For immediate deployment, including option (d), nothing new is required. For a longer-term optimal solution, an anycast-joining protocol is required. It is expected that this would

Fate Sharing

This mechanism exhibits good fate sharing properties. No additional dependencies on any other devices, protocols, or processes exist, beyond those that are already required for performing actual name resolution using DNS (i.e., unicast reachability).

Convergence Time

When a router or link fails, the convergence time is the convergence time of the unicast routing protocol in the site, if the server is off-link, or the convergence time of Neighbor Discovery if the server is on-link. When a server fails, this is also the convergence time of DNS server discovery, but not for name resolution. Since unicast is used for actual name resolution, a host with a list of DNS servers may converge faster to another DNS server it previously discovered, when a DNS server fails.

Scenarios

This mechanism should work in all scenarios discussed. Specifically, it will work in the absence of routers and in the absence of multicast-capable media and routing protocols.

[5.2.](#) Anycast for name resolution

Assign a well-known site-scoped anycast address to DNS servers. Use anycast destination address on DNS queries. Anycast packet will eventually reach the topologically a closest DNS server. DNS server will send a reply.

While the other mechanisms evaluated try to obtain unicast IPv6 addresses of DNS servers, this approach uses anycast as the actual name resolution transport.

Acquiring the domain name and search path must still be done as in [section 5.1](#) above, however.

[5.2.1.](#) Configuration

Define a well-known site-scoped anycast address, and write it down on a document. Let us assume that it is fec0::9999 for now.

Configure clients to send DNS queries to the address. The configuration does not need to be modified even if the machine moves from a site to another, as the anycast address is well-known.

Configure DNS servers (server-1, server-2 to server-N) with the anycast address, and make them reply the query to fec0::9999. Make sure that packets with the anycast destination address get routed to DNS servers. (see a section below on this topic)

[5.2.2.](#) Actual use

- (1) When a DNS name resolution is necessary for a client, the client would transmit a DNS query (usually UDP) toward fec0::9999.

client -> fec0::9999 DNS query

- (2) The DNS query will reach one of the DNS servers (suppose it was "server-X"). The server will resolve the name by making a recursive query. The server will send a reply to the client. If we follow the [RFC 2373](#) restriction that an anycast address cannot be used as a source address, we need to use server-X's unicast address as the source. In this case, DNS clients should not check the source address of DNS replies. Some existing implementations do this to check if the packet was really from the server queried. The check is rather weak, however, as it is easy to spoof the source address. Instead, DNSSEC should be used here.

server-X -> client DNS reply

[5.2.3.](#) Evaluation

Note that there are multiple ways to handle anycast routing in a site. Some of them require us to inject a /128 route into the routing system, some of them does not.

Scalability

The scalability of the proposal is exactly the same as the

Draft

DDDT Report

January 2002

scalability implication in anycast routing. Note that, as the proposal uses a site-scoped anycast address, we need to scale up to a single site - we do not need to scale up to the whole Internet.

Regarding to the site network size, the approach should have no problem, except the possible delay imposed by extra hops between DNS clients and servers.

Regarding to the impact from the number of possible DNS servers to the site routing system, there should be no problem. Even if we are to inject a /128 route to the site routing system, we will have a single extra routing entry on site routers. As we share a single site-scoped anycast address among the DNS servers, we just need to carry a single /128 route in the site routing system.

Regarding to the load balancing among DNS servers, it depends on (1) the network topology in the site, (2) where the DNS servers are located, (3) where the DNS clients are located, and (4) the anycast routing mechanism we use. If we need to implement a perfect load balancing among DNS servers (equal load onto each DNS servers), we should just put DNS servers onto a single IPv6 subnet. Neighbor discovery for anycast address should take care of it. Refer to [\[ND\]](#), section 7.2.7.

There is no impact against the worldwide routing system.

Security

The security issues with this mechanism are the same as those discussed in [section 5.1](#).

Time to Deploy

The proposal is based on standardized mechanisms, including anycast, routing protocols (like RIPng) and DNS lookup over IPv6. This is just a matter of configuration to deploy this proposal.

There are widely-deployed implementations that support anycast address assignment. There are many IPv6 routing protocol

implementations.

As for DNS lookup over IPv6, there are servers (including ISC BIND9) and clients (including BIND9, NetBSD, OpenBSD, FreeBSD,

Expires January 2002

[Page 11]

Draft

DDDT Report

January 2002

and Linux) that support it.

The use of TCP with anycast needs more study (spec-wise), so DNS query/reply over IPv6 TCP may need some time to deploy.

Regarding to the sanity checks made by DNS client implementation, BIND-based implementation has a configurable option to turn off the source address validation on the DNS reply packet.

To deploy this proposal, there is no requirement to run additional servers.

Business Motivation

If there are operating system implementations without anycast support, we apparently cannot use the operating system implementation as a DNS server. If there are commercial operating systems that do not support anycast, they now have a good motivation to support it.

Standardization

As the proposal uses DNS as the content format, it is safe to say that the content is standardized well.

We already have couple of standardized IPv6 routing protocols. So even if we are to use routing protocol to inject a /128 route for the DNS server anycast address, there should be no problem.

For full discussions on anycast routing, refer to [[ADDRARCH](#)] and [[ITOJUN-ANYCAST](#)].

Fate Sharing

As DNS queries use site-scoped anycast address as the destination, DNS query/reply relies upon anycast routing

mechanism. There is no circular dependency between anycast routing and DNS lookups.

Convergence Time

There are couple of possible configurations for anycast packet delivery.

If we run anycast DNS servers on routers, the convergence time will be the same as the router failure recovery time. When a router with DNS server dies, we need to wait until another

Expires January 2002

[Page 12]

Draft

DDDT Report

January 2002

router to take it over.

If we inject /128 routes onto the site routing system for anycast packet delivery, the convergence time will depend on the routing information propagation time of the site routing system.

As to partial failure (like DNS server daemon died but the operating system is alive), there are many failure modes to mention and there are many implementation dependencies. Implementers may need to diagnose partial failure modes in detail.

Scenarios

This mechanism should work in all scenarios discussed.

[5.3.](#) Link-scoped multicast with router-only responses

This transport approach is modeled after IPv6 Neighbor Discovery [\[ND\]](#) transport mechanisms. It could be part of [\[ND\]](#) or another protocol using a similar transport mechanism.

This approach has two modes. The first is routers periodically advertise DNS information (e.g., DNS server addresses, default domain, and search path) to all nodes on a link by sending an advertisement to the all nodes link-scope multicast address (i.e., FF02::1).

The second mode is that nodes needing DNS information can send a solicitation to the all routers link-scope multicast address (e.g., FF02::2) requesting DNS information. Routers with the DNS information will respond with an advertisement with the requested DNS information. This advertisement will be sent to the unicast address of the node sending the solicitation.

[5.3.1.](#) Evaluation

Scalability

This transport approach has excellent scalability. It scales as well as ND and DNS does currently. All of the multicast traffic is local to the link. The periodic advertisement rates can be tuned to environment including turn it off completely and relying on the solicitation/advertisement mode.

Expires January 2002

[Page 13]

Draft

DDDT Report

January 2002

ND transport mechanisms work on a variety of link (e.g., multicast capable, point-to-point, shared media, etc.). It will well suited for both wired and wireless links.

Security

Mechanisms that are being developed to authenticate ND can be applied to this approach as well. The only preconfiguration is done in routers where the DNS information is held.

No new security issues are raised. All of the traffic is link-scoped. Any attacks require link access and if successful only affect nodes on the individual link.

Time to Deploy

This approach requires implementation in nodes wishing to learn DNS information and in routers to provide the information. The router will need to be configured with the DNS information needed in the advertisements.

If done as an extension to ND, it will be a small amount of work to extend the ND implementation to add the additional

functionality. If done in a new protocol it would be more difficult to create the new protocol.

It should be possible to deploy it quickly once the details are scoped out and the implementations are developed. There is no new network wide services such as multicast or anycast that need to be deployed. There are no dependencies with other protocols.

Business Motivation

This approach is a simple extension to existing protocols in existing products and there should be ample motivation to add it to existing products. No complex dependencies are created. No new network transport mechanisms (e.g., anycast) are required. No network operators have to deploy any new transport mechanisms (e.g., multicast and/or anycast).

Standardization

A single document would have to be written to describe an extension to ND or a new protocol. This would have to become

Expires January 2002

[Page 14]

Draft

DDDT Report

January 2002

an IETF standard. This is within the scope of the IPng working group. No other IETF working groups would need to be involved.

Fate Sharing

The only fate sharing issue raised is that it ties the discovery of DNS information with the operation of routers on a link. This is very similar to the common practice with IPv4 of using Bootp relays in routers to communicate with DHCP servers, so in practical terms the issue is very small.

No changes are made to DNS servers or way nodes communicate with DNS servers.

Convergence Time

Convergence time is similar to current IPv4 DNS operation using

DHCP for DNS discovery. No new convergence issues are created. If one of a set of DNS servers is unavailable (e.g., down, isolated, etc.), then normal DNS recover mechanisms will select an alternative DNS servers.

If one of a set of routers is unavailable, other routers on the link will continue providing the DNS information.

If the last DNS server is unavailable, then it is down. If the last router is unavailable, then new nodes will not be able to learn DNS information or reach any DNS server through the router.

Scenarios

This mechanism works over a broad range of scenarios. It should work as well on links that are high performance (e.g., LANs) and low performance (e.g., wireless).

However, this mechanism relies on routers and does not support DNS servers on isolated links. Other local DNS server discovery mechanisms (e.g., multicast DNS) would be needed. This approach is believed to be compatible with multicast DNS approach being developed in the IETF.

It can be used in either a periodic multicast advertisement, or a solicitation/advertisement mode.

[5.4.](#) Link-scoped multicast (with router assist)

In this approach, devices send a request to a (new) well-known link-scoped multicast address. DNS servers and routers both listen to this multicast address. If any DNS servers exist on the list, they can respond directly to the host.

In addition, routers are responsible for using some other mechanism to obtain the DNS server list (e.g., one of the other mechanisms evaluated in this document). Routers with the DNS information will respond with an advertisement with the requested

DNS information.

[5.4.1.](#) Evaluation

Scalability

This mechanism scales fine on each individual link. The scalability across the site depends on the scalability of the inter-router mechanism used.

Security

The use of multicast is vulnerable to the same attacks as unicast. This mechanism can best be authenticated by having the content signed via a content-specific mechanism.

It may be possible to use IPsec to authenticate the discovery messages, but IKE cannot be used for multicast transmission. As a result, barring future research and development of other key distribution protocols, using IPsec instead of a content-specific mechanism would require manual keying.

In contrast to the anycast approaches, it is not strictly necessary to secure the mechanism used to communicate multicast joins from DNS servers to routers, since a rogue server cannot redirect traffic to itself that would otherwise reach a valid server.

Time to Deploy

This approach can be deployed by devices immediately, as link-scoped multicast is already ubiquitous. However, the time to deploy router assist to discover off-link servers depends on the inter-router mechanism chosen.

Business Motivation

No strong business case for choosing this mechanism over the others evaluated is known.

Standardization

Nothing new is required, beyond whatever is required for the content mechanism and the inter-router mechanism.

Fate Sharing

Good fate sharing is preserved in this partial mechanism, as no additional dependencies on other devices or protocols is introduced, beyond whatever is required for the inter-router mechanism.

Convergence Time

The convergence time depends on the convergence time of the inter-router mechanism used.

Scenarios

This mechanism works on an isolated link, and can work in the absence of multicast routing (provided a non-multicast inter-router mechanism is used).

However, it does not support the NBMA link scenario.

[5.5.](#) Site-scoped multicast

Site-scoped multicast could be used to discover all DNS servers within a site. This would be analogous to the way multicast is currently used by IPv6 hosts to discover their neighboring routers, but over the span of a site rather than just a single link.

There are several possible ways for site-scoped multicast to be used for DNS (or any other) server discovery. The clients could send multicast query packets to a well-known, site-local, "all-site-DNS-servers" multicast address, to which all DNS servers would listen and respond. Alternatively, the servers themselves could send periodic multicast advertisement packets to a well-known, site-local, "all-site-DNS-clients" multicast address, to

which all clients would listen in order to discover the presence and address(es) of all of the site's DNS servers. Probably the best approach, from the point of view of responsiveness, scaling, and traffic volume is to use both of those techniques in combination, exactly as they are used for router discovery, as follows:

- o Two well-known, site-local scoped, multicast addresses are assigned by IANA, one for all-site-DNS-routers and one for all-site-DNS-clients.
- o When a DNS server starts up, it multicasts a small number (say, 3) of advertisement messages to the all-site-DNS-clients address at short (sub-second) intervals, and then continues to retransmit the advertisement messages but at much larger intervals (30 seconds or more). It also sends a unicast response to any (unicast or multicast) DNS discovery query message it receives.
- o When a client starts up (or when the first attempt to perform a DNS look-up occurs), if the client has not yet heard any advertisements from DNS servers, it multicast up to a small number (say, 3) of query messages to the all-site-DNS-servers address at short (sub-second) intervals, stopping as soon as it receives a response from a DNS server or it reaches the small retransmission limit. From that point on, the client does not send any multicast queries, but rather relies on passively discovering additional DNS servers (and DNS servers that fail and subsequently recover) by listening on the the all-site-DNS-clients address for advertisements.

The response and advertisement messages would ideally contain only the address of the sending DNS (rather than a list of multiple DNS servers, which would impose an undesirable requirement for configuration of, or a coordinating protocol among, the servers). That one address could be found either in the Source Address field of the IPv6 header or in the content of the message. Clients would build their lists of candidate DNS servers by taking the union of the responses/advertisements received.

To avoid the need for additional packet exchanges, the response and advertisement messages should also carry the other information

Draft

DDDT Report

January 2002

required by the client, i.e., the domain name and default domain search list to be used by clients within the site. [Open issue: what ought a client to do when not all DNS servers are advertising the same information?]

[5.5.1.](#) Evaluation

Scalability

The recommended technique, above, imposes a steady-state traffic load on a site of one multicast packet per DNS server, every (30 second or greater) advertisement transmission interval. Because almost all nodes would be expected to listen to those multicasts, they would effectively be site-wide broadcast messages. However, note that the retransmission interval can safely be made quite large to minimize the overhead of those messages, since the technique does not rely on a small interval for its responsiveness; rather, the periodic multicast advertisements serve the role of ensuring long-term consistency and recovering from rare failures (e.g., loss of three packets in a row, or site partitioning).

In addition to the overhead of the periodic advertisements, there is also the small (maximum 3 packet) burst load of queries/ responses/advertisements that occur whenever a client or DNS server starts up. Note that these costs are comparable to those incurred by the anycast techniques described in other sections. They can also be further reduced by a more sophisticated design, in which queries and/or responses are delayed for small, randomized time intervals in order to do "suppression" of identical queries or responses that occur very close together, e.g., when a site comes up after a site-wide power-failure.

Using this technique requires that all, or almost all, nodes within a site send multicast packets at some point or another. A multicast routing implementation that instantiates state for every active multicast source node would have a state cost on the order of the number of nodes in the site. It may be preferable to use a multicast routing protocol that instantiates state only for each active multicast source *subnet* (which is sufficient to support shortest-path multicast delivery), or for each multicast destination address

(so-called "shared-tree" multicast, which does not try to achieve shortest-path delivery, which is not important for this

particular application).

Security

The security evaluation for all types of multicast is discussed in [section 5.4.1](#).

Time to Deploy

It is expected that most sites will not have site-wide multicast deployed in the near-term future. As a result, it may be some time before a site-scoped multicast solution could be deployed.

Business Motivation

Given the perceived extra cost of managing a multicast-enabled infrastructure, when other solutions relying only on unicast routing exist, it is expected that no strong business case for choosing site-scoped multicast exists.

Standardization

Protocols needed for multicast connectivity are already on the standards track, or in progress.

Fate Sharing

This mechanism places an extra dependency on the correct functioning of the multicast routing system. There are no dependencies on any extra devices, however, beyond those that are already required for name resolution.

Convergence Time

When a router or link fails, the convergence time is the convergence time of the multicast routing protocol in the site.

When a server fails, this is also the convergence time of DNS server discovery, but not for name resolution. Since unicast is used for actual name resolution, a host with a list of DNS servers may converge faster to another DNS server it previously discovered, when a DNS server fails.

Scenarios

This mechanism works on an isolated link with a DNS server but no routers.

However, it does not work in a site with no multicast routing enabled, or among hosts connected over an NBMA link.

[5.6.](#) Hybrid

It is possible to combine the above approaches in some situations. For example, the option for link-scoped multicast with router assist requires one of the other options to form a complete solution.

It is also possible for one of the above mechanisms to be the default, but allow use of one of the other mechanisms based on local manual configuration, or on configuration obtained from information obtained from say Router Advertisements. Of course, if routers can distribute the address to use for discovery, then a router which does not have multicast routing enabled must not distribute a site-scoped multicast address.

The disadvantage with router overrides is that devices must be prepared to handle multiple mechanisms, increasing the complexity of the implementations.

[6.](#) Message Content Mechanisms

[6.1.](#) DHCP

In this mechanism, the messages sent to DNS servers are DHCP-format messages, and the DNS servers send DHCP messages in response.

The domain name can be obtained by having the DNS server include a Domain Name option in the response. A DHCP option to obtain a domain search path [[DOMSEARCH](#)] is currently being discussed in the DHC WG. Without this option, the search path behavior would be no different from the behavior today, where the search path is simply computed based on the domain name obtained. A list of DNS servers would also require a new DHCP option, which is already required if DHCP for IPv6 is to be implemented.

Expires January 2002

[Page 21]

Draft

DDDT Report

January 2002

[6.1.1.](#) Evaluation

Scalability

Only one round trip of messages is required, and there is no need to run any additional servers. In other respects, this mechanism has the same scalability properties as the underlying transport mechanism chosen.

Security

A means of securing DHCP content that does not rely on IPsec has been proposed in the DHC Working Group [[DHCPAUTH](#)].

Time to Deploy

This would require time to implement a DHCP parser in DNS servers. In addition, DHCP for IPv6 is still being defined by the DHC Working Group.

Business Motivation

Devices wanting to resolve names probably already have to implement a DHCP parser and be able to obtain DNS-related configuration information using DHCP messages. As a result, it is expected that little extra work would be required by stack

implementers beyond implementing DHCPv6.

Standardization

The DHCP for IPv6 content format is still being defined by the DHC Working Group.

Fate Sharing

Good fate sharing would require that DNS servers also implement a DHCP parser. Otherwise, it is necessary to run a DHCP service on the DNS servers which simply responds to requests for DNS information. In such a configuration, an additional dependency on the DHCP service would be introduced by this mechanism.

Convergence Time

This mechanism has the same convergence time as the underlying transport mechanism chosen.

Expires January 2002

[Page 22]

Draft

DDDT Report

January 2002

Scenarios

This mechanism works in the same scenarios as the underlying transport mechanism chosen.

[6.2.](#) DNS

DNS itself can be used, as long as the transport mechanism ensures that discovery messages reach DNS servers, to obtain the DNS server list, default domain name, and domain search path.

DNS has many record types that could be used. For example, SOA and NS records contain relevant information. However, the name resolution configuration information which an administrator desires that a host should use may not match the usual configuration of the DNS server (e.g., might use a different domain name). Hence, using information encoded in separate records is more flexible. It is also desirable that an administrator can give different configuration information to

hosts in different subnets, since this capability exists when a DHCP server is present.

There are many ways to accomplish the above. On the query side, the subnet prefix can be encoded in the hostname part of the query. On the response side, the server list, domain name, and search path can be encoded in the response, potentially using SRV records [[SRV](#)] with a special encoding, or using TXT records [[TXT](#)], or even defining a new record type.

[6.2.1](#). Evaluation

Scalability

The mechanism has one round trip to the DNS server and the security overhead. Using Secret Keys, if possible, will produce no more packets, but some processing on the DNS server to match the clients secret key using [[TSIG](#)]. Using Diffie-Hellman with [[DNSSEC](#)], if many clients attempt this at the same time will put extreme processing demands on the DNS server. But it is doubtful that one or a few DNS Servers will handle 1000's of clients. See Security Evaluation below.

Security

If the client can preconfigure a well known private or public

Expires January 2002

[Page 23]

Draft

DDDT Report

January 2002

key then TSIG or DNSSEC can be used with the same packets presented for the query. If this is not the case, then either TSIG keys will have to be negotiated using [[TKEY](#)] or a Diffie-Hellman Key [[DIFFSEC](#)] exchange will have to take place using DNSSEC. After the client has the proper key then the query can be performed.

Time to Deploy

This mechanism can be deployed now. If the address provided to the Client is an Anycast address then resolver implementations would have to not use the present mechanisms to verify the source address of a QUERY response is equivalent to the destination address of the QUERY to the DNS Server. Likewise

for Multicast DNS.

Business Motivation

All DNS suppliers are now implementing TSIG and DNSSEC. The biggest business motivation for this mechanism is that it requires the least new code of any of the mechanisms evaluated, and provides the greatest robustness since there are no dependencies on other protocols or services.

Standardization

If SRV or TXT records are used, there is no standardization required other than the specific mechanism document in the IPv6 Working Group. If a new record type is required, the DNSEXT Working Group would be involved.

Fate Sharing

There are no dependencies on other than the normal DNS processes implemented today, nor are there any new dependencies created from these content messages.

Convergence Time

There is no degradation in convergence time with this content set of messages, other than awaiting for a failed network to respond again or a DNS Server(s) to come back up after reboot. Once the resolver has done the DNS queries the knowledge from the server on most implementations becomes stateful and stored to non-volatile storage. In the case of embedded systems with no non-volatile storage the DNS Server address would have to be relearned.

Scenarios

This mechanism should work in all scenarios discussed.

Expires January 2002

[Page 24]

Draft

DDDT Report

January 2002

[6.3.](#) Node Information Query

ICMPv6 node information query [[NODEINFO](#)] provides a way to query IPv6 addresses assigned to a machine. This could be used to query IPv6 unicast address for a DNS server, from a DNS server IPv6 anycast/multicast address.

To get a local domain name and search path, new query types (QTypes) would need to be defined for these information types.

6.3.1. Evaluation

Scalability

As ICMPv6 node information query will be handled by the DNS server node itself, there is no need for running an additional server. This mechanism has the same scalability properties as the underlying transport mechanism chosen.

Security

IPsec is the only way to prevent malicious packets. It depends on the actual transport protocol used with ICMPv6 node information query, if IPsec is usable or not. For example, it is rather hard to use IPsec with anycast/multicast, so IPsec may not work well if we use anycast/multicast.

Time to Deploy

While there exist some implementations of node information queries, implementation is not ubiquitous.

Business Motivation

There does not seem to be any strong business motivation for implementing a node information query parser, in preference to other options.

Standardization

The ICMPv6 node information query has not made it to RFC status yet, but is owned by the IPv6 Working Group.

Fate Sharing

Depends on how we combine this with other mechanisms, and

Convergence Time

Depends on how we combine this with other mechanisms, and transport protocols.

Scenarios

This mechanism should work in all scenarios in which the transport mechanism works.

[6.4.](#) RA Extensions

The basic approach is to define a new ND option that would contain the DNS information. Existing ND transport mechanisms (i.e., advertisements and solicitations) mechanisms would be used. This would work in the same way that nodes learn about routers and prefixes, etc.

A ND new option would be defined that routers could send in router advertisements.

This approach has two modes of operation. The first is routers periodically send the DNS Configuration Option in their periodic router advertisements.

The second mode is that nodes needing DNS information can send a solicitation to the routers on the link requesting the DNS Configuration options.

This approach has an issue that the DNS information needs to be configured in the routers doing the advertisements. There are several approaches to this:

- a) Many routers may already have most of this information configured. Routers also function as hosts and may have DNS server addresses, default domain, and list of domains to search in their configuration file. For example, Cisco IOS [[IOS](#)] has the following commands:

`ip domain-name name`

Define a default domain name that the Cisco IOS software will use to complete unqualified host names.

`ip domain-list name`

Define a list of default domain names to complete unqualified host names.

`ip name-server server-addr1 [server-addr2...server-addr6]`

Specify one or more hosts that supply name information.

At least in the case of Cisco routers, and probably most others, no additional work is needed to add the DNS information to the routers.

- b) The next approach, that is consistent with current router management practice, is to configure the router manually with the DNS information that they would advertise w/ IPv6 ND. This could be done by CLI commands as shown above and/or by other mechanisms normally used to configure routers (e.g., web interfaces, proprietary tools, etc.)

The work to implement this is minor and part of implementation the feature in the router. No new management/distribution protocols.

- c) The next approach, also consistent with current router management practice, is that the router configuration files are created centrally and remotely installed on the router. This is done today with a mix to special tools, text editors, perl scripts, vendor management systems, etc. It is very common practice to create files of CLI commands and push them to the routers. No new management and/or distribution protocols are required.
- d) Create or extend an SNMP MIB (do we have an ND MIB?) that contains this information and use existing management tools to set the information in the router. Also common practice and no new management/distribution protocols. A MIB is required and extensions to management tools to support the new variables.

Draft

DDDT Report

January 2002

- e) Extend Router Renumbering (RR) to contain this information and use it to advertise it to the routers. This is mostly consistent with what RR does and allows control reasonable fine grained control of to allow different information per router or even per interface.

Without changing the intended usage of Router Advertisements, the natural transport mechanism used would be Link-scoped multicast with router-only response.

[6.4.1.](#) Evaluation

Scalability

This mechanism has excellent scalability properties. It scales as well as ND and DNS does currently. All of the multicast traffic is local to the link. The periodic advertisement rates can be tuned to environment including turn it off completely and relying on the solicitation/advertisement mode.

Security

ND does not currently have authentication mechanisms built in, but there is ongoing work to investigate using AH with ND. This approach would use what ever solution is selected for ND authentication.

Time to Deploy

This approach requires implementation in nodes wishing to learn DNS information and in routers to provide the information. The router will need to be configured with the DNS information needed in the advertisements.

It is a relatively minor amount of work to add a new option to an existing router and/or node implementations of ND. It should be possible to deploy it quickly once the details are scoped out and the implementations are developed. There is no new network wide services such as multicast or anycast that need to be deployed. There are no dependencies with other protocols.

Business Motivation

The business motivation is very good. The implementation,

Expires January 2002

[Page 28]

Draft

DDDT Report

January 2002

documentation, and support cost are minor and very much in line with existing ND features. There are no new protocols and/or transports to deploy, document, and support.

Devices wanting to discover DNS servers already have to have an RA parser in them. It is expected that implementors would find it easier to extend an existing parser than to add a new parser.

Standardization

This mechanism would require a new RA option format to be standardized. This would be done by the IPv6 Working Group.

Fate Sharing

The only fate sharing issue raised is that it ties the discovery of DNS information with the operation of routers on a link. This is very similar to the common practice with IPv4 of using Bootp relays in routers to communicate with DHCP servers, so in practical terms the issue is very small.

No changes are made to DNS servers or way nodes communicate with DNS servers.

Convergence Time

Convergence time is similar to current IPv4 DNS operation using DHCP for DNS discovery. No new convergence issues are created. If one of a set of DNS servers is unavailable (e.g., down, isolated, etc.), then normal DNS recover mechanisms will select an alternative DNS servers.

If one of a set of routers is unavailable, other routers on the link will continue providing the DNS information.

If the last DNS server is unavailable, then it is down. If the last router is unavailable, then new nodes will not be able to learn DNS information or reach any DNS server through the router.

Scenarios

This mechanism works over a broad range of scenarios. It should work as well on links that are high performance (e.g.,

Expires January 2002

[Page 29]

Draft

DDDT Report

January 2002

LANs) and low performance (e.g., wireless).

However, this mechanism relies on routers and does not support DNS servers on isolated links. Other local DNS server discovery mechanisms (e.g., multicast DNS) would be needed. This approach is believed to be compatible with multicast DNS approach being developed in the IETF.

It can be used in either a periodic multicast advertisement, or a solicitation/advertisement mode.

[6.5.](#) SLP

The Service Location Protocol [[RFC 2608](#)] provides a general framework for service discovery. Services are modeled on the basis of their type, location, and a set of attributes.

Clients use SLP to request services on the basis of the attributes required and retrieve both the location and attributes of all services fulfilling the request. In the case of DNS server discovery, a DNS client could discover the location, domain name and search path to use, all in one message round trip (SrvRqst/SrvRply) if the Attribute List Extension is used, or two round trips (SrvRqst/SrvRply, AttrRqst/AttrRply) if the Attribute List extension is not supported.

Without changing the basic SLPv2 protocol, the natural transport mechanism used would be Site-scoped multicast.

[6.5.1.](#) Evaluation

Scalability

This mechanism has the same scalability properties as the underlying transport mechanism chosen.

Security

SLPv2 provides its own security so that those who obtain service location and attribute information can verify the signature over it. These digital signatures are calculated using keys which are distributed by some external mechanism (SLPv2 does not provide mechanisms for cryptographic key

Expires January 2002

[Page 30]

Draft

DDDT Report

January 2002

distribution).

Time to Deploy

SLPv2 for IPv6 is not yet a proposed standard. It will shortly enter IETF last call.

There are no IPv6 SLPv2 implementations yet. SLPv2 over IPv4 has been implemented by many vendors and is used for a variety of purposes.

Business Motivation

Rather than each protocol supplying its own service discovery protocol, SLP can provide a general purpose mechanism which will minimize the software required and allow for common operational considerations and management.

Standardization

SLPv2 is a Proposed Standard. SLPv2 over IPv6 and the Attribute List Extension are both about to enter IETF last call before being advanced to Proposed Standard.

Fate Sharing

SLP would require an additional protocol to be used to configure DNS resolvers - namely, a SLP user agent library. DNS servers would need to be advertised using a SLP service agent - this functionality could be provided by a library invoked by a DNS server, or by an additional service running on the DNS server host (although this is more complicated and less assured of not advertising the DNS server when it is in fact not available).

Convergence Time

If no directory agents are deployed, convergence time is on the order of milliseconds: the only way that a DNS resolver could discover a DNS server which was not available would be if the DNS server failed after answering that it was available.

If directory agents are available, convergence depends on the soft state registration lifetime - which could be of any granularity on the order of seconds - but typically is on the

order of minutes. During this interval it is possible for a client to discover a service which has gone off line since the service location has been cached by a directory agent temporarily.

Scenarios

- o Isolated link with a DNS server but no routers

SLP requests multicast on the link would enable DNS clients to directly discover DNS servers, domain name and search path.

The DNS clients and servers would have to make use of a SLP library to accomplish this. Another alternative is a service advertising process co-resident on the DNS server

host which advertises the DNS server on its behalf. In this case, no modification of the DNS server would be necessary.

- o Site with routers, but no multicast routing enabled

Routers use SLP to discover DNS servers on attached links. The routers can then use other mechanisms to distribute the location of the DNS servers (add an anycast routing entry for each DNS server, send extensions to routing advertisements, etc.)

DNS clients could use link-scoped multicast SLP discovery, and routers could answer these requests on behalf of DNS servers. The problem with this approach is that it does not specify the way in which DNS server locations are propagated between routers.

Thus, SLP can be used as a mechanism to provide dynamic and decentralized service discovery for the system, but only between the routers and the DNS servers. The mechanism between the routers to propagate DNS service locations would have to be satisfied by some additional protocol (such as OSPF extensions).

In summary, a link-scoped multicast with router-assist transport mechanism would be needed in this scenario.

- o NBMA Link

This scenario could not be supported without changing the SLP protocol to work with anycast addresses. However, it is not expected that this is a very important scenario.

[6.6.](#) Something New

Any number of other mechanisms (e.g., HTTP, SNMP, etc.) could also be extended, but it is expected that any other mechanism

would take at least as long to deploy, and have no significant advantage over the other mechanisms evaluated above.

7. Recommendations

Based on team discussion, and WG consensus at IETF 49, the recommendation for transport mechanism is to use site-scoped anycast. That is, IANA should allocate a well-known site-local anycast address for DNS servers.

No specific recommendation is made regarding using anycast for discovery-only or for actual name resolution. However, we observe that choice can be made by individual sites as follows. Nodes will use the anycast address to discover DNS-specific information such as the domain name, search path, and DNS server list. If the DNS server list contains just the anycast address itself, the anycast address will also be used for name resolution.

Based on subsequent team discussion, the recommendation to the WG regarding the content mechanism is to use DNS. That is, the recommendation is that the WG should define the details of how the DNS server list, domain name, and search path, can be encoded in DNS records. The team's initial recommendation is that the IPv6 WG investigate whether SRV records are sufficient, and if not, then either TXT records or a new record type should be used. It is believed that the information should and can be encoded in a single response message.

Finally, it is recommended that the "Other stateful configuration" flag in the Router Advertisement be used to control whether to use DHCP or this mechanism. That is, the analysis and recommendations in this document apply only to links where the "Other stateful configuration" flag is zero.

8. [Appendix A](#): Summary Grid

Figure 1 summarizes information on transport mechanisms:

|Any(Disc)|Any(Res)|LnkM(Rtr)|LnkM(Gen)|SiteM

SW upgrades	-/S/SR	-/S/SR	R	R	UR
Reconfig changes	S	-	R	S	-
New dependencies	-	-	router, linkmcast	linkmcast	multicast routing, linkmcast
Convergence time	fast	per-rtg	fast	fast	fast
Can use IKE	no	no	no	no	no
Standards work	-/ipngwg	-/ipngwg	ipngwg	ipngwg	-
Deployable "now"	yes	yes	no	no	no

Figure 1: Transport Mechanism Summary

Key:

/ separates multiple deployment options

- = No devices

S = All DNS servers

SR = All routers with directly-connected DNS servers

UR = All routers which don't have multicast routing implemented

R = All routers

SW upgrades = what boxes, other than devices wanting to discover DNS servers, require software upgrades?

Reconfig changes = what boxes need to be reconfigured when the set of DNS servers changes?

New dependencies = what new dependencies are added?

Convergence Time = how long does it take to contact a new DNS server when a server/link/router fails? "Fast" = a device can immediately use an alternate server if reachable. "Per-rtg" = Device must wait until routing converges for the unreachable

address.

Can use IKE = can IKE be used for key negotiation?

Standards work = what WGs would be required to standardize new items?

Deployable "now" = could a new client using this mechanism be deployed immediately, without requiring implementation of new code for routers or servers?

Figure 2 summarizes information on content mechanisms:

	DHCP	DNS	NIQ	RA	SLP
Round trips used	1	1	1	1	1
Has own signatures	yes	yes	-	no	yes
Has own key dist	-	yes	-	-	-
Standards work	dhc	dnstxt?	ipngwg	ipngwg	svrloc
Need addl parser	in servers	-	yes	-	yes
Generalizable	yes	yes	yes	-	yes
Deployable "now"	no	yes	no	no	no

Figure 2: Content Mechanism Summary

Round trips used = How many round trips are required?

Has own signatures? = Does the content have its own signature facility? (a "no" means it relies on IPsec)

Standards work = what WGs would be required to standardize new items?

Need addl parser = besides any parsers that a device must already implement to perform basic name resolution, does the mechanism introduce a requirement for another parser?

Generalizable = can the mechanism be generalized to allow

Draft

DDDT Report

January 2002

discovery of other types of information or services?

Deployable "now" = could a new client using this mechanism be deployed immediately, without requiring implementation of new code for routers or servers?

Draft

DDDT Report

January 2002

9. Authors' Addresses

Bernard Aboba
Microsoft
One Microsoft Way
Redmond, WA 98052, USA
Email: aboba@internaut.com

Jim Bound
Compaq Computer Corporation
110 Spitbrook Road ZK3-3/U14
Nashua, NH 03062-2698
Email: bound@zk3.dec.com

Steve Deering
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706, USA
Email: deering@cisco.com

Erik Guttman
Sun Microsystems
Bahnstr. 2
74915 Waibstadt, Germany
Email: erik.guttman@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

Robert M. Hinden
Nokia
313 Fairchild Drive
Mountain View, CA 94043, USA

Email: hinden@iprg.nokia.com

Tatuya JINMEI
Research and Development Center, Toshiba Corporation
1 Komukai Toshiba-cho, Kawasaki-chi
Kanagawa 212-8582
Japan
Email: jinmei@isl.rdc.toshiba.co.jp

Expires January 2002

[Page 37]

Draft

DDDT Report

January 2002

Atsushi Onoe
Internet Systems Laboratory, IN Laboratories, Sony Corporation
6-7-35 Kitashinagawa, Shinagawa-ku, Tokyo 141-0001
Japan
Email: onoe@sm.sony.co.jp

Hesham Soliman
Ericsson Australia
61 Rigall St., Broadmeadows
Melbourne, Victoria 3047
Australia
Email: Hesham.Soliman@ericsson.com.au

David Thaler
Microsoft
One Microsoft Way
Redmond, WA 98052, USA
Email: dthaler@microsoft.com

10. References

[ANYCAST]

Partridge, C., Mendez, T., and W. Milliken, "Host Anycasting Service", [RFC 1546](#), November 1993.

[ADDRARCH]

Hinden, R., and S. Deering, "IP Version 6 Addressing Architecture", [RFC 2373](#), July 1998.

[DHCPAUTH]

Droms, R., and W. Arbaugh, "Authentication for DHCP Messages", [draft-ietf-dhc-authentication-16.txt](#), January 2001.

[DIFFSEC]

D. Eastlake, "Storage of Diffie-Hellman Keys in the Domain Name System (DNS)", [RFC 2539](#), March 1999.

[DNSSEC]

D. Eastlake, "Domain Name System Security Extensions", [RFC 2535](#), March 1999.

[DOMSEARCH]

B. Aboba, "DHCP Domain Search Option", [draft-aboba-dhc-](#)

Expires January 2002

[Page 38]

Draft

DDDT Report

January 2002

domsearch-01.txt, December 2000.

[HOST-ANYCAST]

Haberman, B., and D. Thaler, "Host-based Anycast using MLD", [draft-haberman-ipngwg-host-anycast-00.txt](#), February 2001.

[IOS]

Cisco IOS Release 12.0 Configuration Guides,
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/cbk>

[ITOJUN-ANYCAST]

Jun-ichiro Hagino and K. Ettikan, "An analysis of IPv6 anycast", [draft-itojun-ipv6-anycast-analysis-02.txt](#), February 2001.

[MDNS]

Esibov, L., Aboba, B., and D. Thaler, "Multicast DNS", [draft-ietf-dnsext-mdns-00.txt](#), November 2000.

[ND] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", [RFC 2461](#), December 1998.

[NODEINFO]

Matt Crawford, "IPv6 Node Information Queries", [draft-ietf-ipngwg-icmp-name-lookups-07.txt](#), August 2000.

[RFC1034]

P. Mockapetris, "Domain Names - Concepts and Facilities", STD 13, [RFC 1034](#), November 1987.

[RFC1035]

P. Mockapetris, "Domain Names - Implementation and Specifications", STD 13, [RFC 1035](#), November 1987.

[RIPMD5]

Baker, F., and R. Atkinson, "RIP-2 MD5 Authentication", [RFC 2082](#), January 1997.

[RIPNG]

Malkin, G., and R. Minnear, "RIPng for IPv6", [RFC 2080](#), January 1997.

[SLPv2]

Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2", [RFC 2608](#), June 1999.

Expires January 2002

[Page 39]

Draft

DDDT Report

January 2002

[SRV]

Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", [RFC 2782](#), February 2000.

[TSIG]

Vixie, P., Gudmundsson, O., Eastlake, D. and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", [RFC 2845](#), May 2000.

[TKEY]

D. Eastlake, "Secret Key Establishment for DNS (TKEY RR)" [RFC 2930](#), September 2000

[TXT]

R. Rosenbaum, "Using the Domain Name System To Store Arbitrary String Attributes", [RFC 1464](#), May 1993.

[11.](#) Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed

for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

Table of Contents

1 Introduction	2
2 Requirements	2
3 Criteria for Evaluation	3

4	Taxonomy	4
5	Transport Mechanisms	4
5.1	Anycast for DNS server discovery only	4
5.1.1	Evaluation	7
5.2	Anycast for name resolution	9
5.2.1	Configuration	10
5.2.2	Actual use	10
5.2.3	Evaluation	10
5.3	Link-scoped multicast with router-only responses	13
5.3.1	Evaluation	13
5.4	Link-scoped multicast (with router assist)	16
5.4.1	Evaluation	16
5.5	Site-scoped multicast	17
5.5.1	Evaluation	19
5.6	Hybrid	21
6	Message Content Mechanisms	21
6.1	DHCP	21
6.1.1	Evaluation	22
6.2	DNS	23
6.2.1	Evaluation	23
6.3	Node Information Query	25
6.3.1	Evaluation	25
6.4	RA Extensions	26
6.4.1	Evaluation	28
6.5	SLP	30
6.5.1	Evaluation	30
6.6	Something New	33
7	Recommendations	33
8	Appendix A: Summary Grid	34
9	Authors' Addresses	37
10	References	38
11	Full Copyright Statement	41