IPv6 WG M. Crawford
Internet-Draft Fermilab
Expires: November 6, 2005 B. Haberman, Ed.
JHU APL
May 5, 2005

IPv6 Node Information Queries
draft-ietf-ipngwg-icmp-name-lookups-11

Status of this Memo

Copyright Notice

Abstract

This document describes a protocol for asking an IPv6 node to supply certain network information, such as its hostname or fully-qualified domain name. IPv6 implementation experience has shown that direct queries for a hostname are useful, and a direct query mechanism for other information has been found useful in serverless environments and for debugging.

Table of Contents

1. Introduction

This document specifies a mechanism for discovering information about names and addresses. The applicability of these mechanics is currently limited to diagnostic and debugging tools. In the global internet, the Domain Name System[1][2] is the authoritative source of such information and this specification is not intended to supplant or supersede it. And in fact, in a well-supported network the names and addresses dealt with by this mechanism will be the same ones, and with the same relationships, as those listed in the DNS.

This new Node Information protocol does provide facilities which are not found in the DNS - for example discovering relationships between addresses without reference to names. And the functions that do overlap with the DNS may be useful in serverless environments, for debugging, or in regard to link-local and unique-local addresses [3] which often will not be listed in the DNS.

2. Applicability Statement

IPv6 Node Information Queries include the capability to provide forward and reverse name lookups independent of the DNS by sending packets directly to IPv6 nodes or groups of nodes.

The applicability of these mechanics is currently limited to diagnostic and debugging tools. These mechanisms can be used to learn the addresses and names for nodes on the other end of a point-to-point link or nodes on a shared-medium link such as an Ethernet. This is very useful when debugging problems or when bringing up IPv6 service where there isn't global routing or DNS name services available. IPv6's large auto-configured addresses make debugging network problems and bringing up IPv6 service difficult without these mechanisms. An example of a IPv6 debugging tool using IPv6 Node Information Queries is the ping6 program in the KAME, USAGI, and other IPv6 implementations <http://www.kame.net>.

The mechanisms defined in this document may have wider applicability in the future (for example, name lookups in zero configuration networks, global reverse name lookups, etc.), but any use beyond debugging and diagnostic tools is left for further study and is beyond the scope of this document.

3. Terminology

A "Node Information (or NI) Query" message is sent by a "Querier" node to a "Responder" node in an ICMPv6 packet addressed to the "Queried Address." The Query concerns a "Subject Address" (which may differ from the Queried Address) or a "Subject Name". The Responder

sends a "Node Information Reply" to the Querier, containing information associated with the node at the Queried Address. A node receiving a NI Query will be termed a Responder even if it does not send a reply.

The word "name" in this document refers to a hostname with or without the domain. Where necessary, the cases of fully-qalified and single-label names will be distinguished.

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 [4].

Packet fields marked "unused" must be zero on transmission and, aside from inclusion in checksums or message integrity checks, ignored on reception.

4. Node Information Messages

Two types of Node Information messages, the NI Query and the NI Reply, are carried in ICMPv6 [5] packets. They have the same format.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Qtype             |             Flags             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                             Nonce                             +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
/                             Data                              /
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fields:

o Type

   * 139 - NI Query

   * 140 - NI Reply

o Code

  * For NI Query

    + 0 - Indicates that the Data field contains an IPv6 address which is the Subject of this Query.

    + 1 - Indicates that the Data field contains a name which is the Subject of this Query, or is empty, as in the case of a NOOP or Supported Qtypes query.

    + 2 - Indicates that the Data field contains an IPv4 address which is the Subject of this Query.

  * For NI Reply

    + 0 - Indicates a successful reply. The Reply Data field may or may not be empty.

    + 1 - Indicates that the Responder refuses to supply the answer. The Reply Data field will be empty.

    + 2 - Indicates that the Qtype of the Query is unknown to the Responder. The Reply Data field will be empty.

o Checksum - The ICMPv6 checksum.

o Qtype - A 16-bit field which designates the type of information requested in a Query or supplied in a Reply. Its value in a Reply is always copied from the corresponding Query by the Responder. Five values of Qtype are specified in this document.

o Flags - Qtype-specific flags which may be defined for certain Query types and their Replies. Flags not defined for a given Qtype must be zero on transmission and ignored on reception, and must not be copied from a Query to a Reply unless so specified in the definition of the Qtype.

o Nonce - An opaque 64-bit field to help avoid spoofing and/or to aid in matching Replies with Queries. Its value in a Query is chosen by the Querier. Its value in a Reply is always copied from the corresponding Request by the Responder.

o Data - In a Query, the Subject Address or Name. In a Reply, Qtype-specific data present only when the ICMPv6 Code field is zero. The length of the Data may be inferred from the IPv6 header's Payload Length field [2460], the length of the fixed portion of the NI packet and the lengths of the ICMPv6 header and

intervening extension headers.

Note that the type of information present in the Data field of a Query is declared by the ICMP Code, while the type of information, if any, in the Data field of a Reply is determined by the Qtype.

When the Subject of a Query is a name, the name MUST be in DNS wire format [2]. The name may be either a fully-qualified domain name, including the terminating zero-length label, or a single DNS label followed by two zero-length labels. Since a Query contains at most one name, DNS name compression MUST NOT be used.

5. Message Processing

The Querier constructs an ICMP NI Query and sends it to the address from which information is wanted. When the Subject of the Query is an IPv6 address, that address will normally be used as the IPv6 destination address of the Query, but need not be if the Querier has useful a priori information about the addresses of the target node. An NI Query may also be sent to a multicast address of link-local scope [3].

When the Subject is a name, either fully-qualified or single-component, and the Querier does not have a unicast address for the target node, the query MUST be sent to a link-scope multicast address formed in the following way. The Subject Name is converted to the canonical form defined by DNS Security [6], which is uncompressed with all alphabetic characters in lower case. (If additional DNS label types or character sets for host names are defined, the rules for canonicalizing those labels will be found in their defining specification.) Compute the MD5 hash [10] of the first label of the Subject Name -- the portion beginning with the first one-octet length field and up to, but excluding, any subsequent length field. Append the first 32 bits of that 128-bit hash to the prefix FF02:0:0:0:0:2::/96. The resulting multicast address will be termed the "NI Group Address" for the name. A node will support an "NI Group Address" for each associated Subject Name.

The Nonce should be a random or good pseudo-random value to foil spoofed replies. An implementation which allows multiple independent processes to send NI queries MAY use the Nonce value to deliver Replies to the correct process. Nonetheless, such processes MUST check the received Nonce and ignore extraneous Replies.

If true communication security is required, IPsec [11] must be used. Providing the infrastructure to authenticate NI Queries and Replies may be quote difficult outside of a well-defined community.

Upon receiving a NI Query, the Responder must check the Query's IPv6 destination address and discard the Query without further processing unless it is one of the Responder's unicast or anycast addresses, or a link-local scope multicast address which the Responder has joined. Typically the latter will be a NI Group Address for a name belonging to the Responder or a NI Group Address for a name for which the Responder is providing proxy service. A node MAY be configurable to discard NI Queries to multicast addresses other than its NI Group Address(es) but if so, the default configuration MUST be not to discard them.

A Responder must also silently discard a Query whose Subject Address or Name (in the Data field) does not belong to that node, unless it is providing proxy service for that Subject. A single-component Subject Name matches any fully-qualified name whose first label matches the Subject. All name matching is done in a case-independent manner consistent with DNSSEC name canonicalization [6].

Next, if Qtype is unknown to the Responder, it must return a NI Reply with ICMPv6 Code = 2 and no Reply Data. The Responder should rate-limit such replies as it would ICMPv6 error replies [5].

Next, the Responder should decide whether to refuse an answer, based on local policy. (See "Security Considerations" for recommended default behavior.) If an answer is refused, the Responder may send a NI Reply with ICMPv6 Code = 1 and no Reply Data. Again, the Responder should rate-limit such replies as it would ICMPv6 error replies [5].

Finally, if the Qtype is known and the response is allowed by local policy, the Responder must fill in the Flags and Reply Data of the NI Reply in accordance with the definition of the Qtype and transmit the NI Reply with an ICMPv6 source address equal to the Queried Address, unless that address was an anycast or a multicast address. If the Queried Address was anycast or multicast, the source address for the Reply SHOULD be one belonging to the interface on which the Query was received.

If the Query was sent to an anycast or multicast address, transmission of the Reply MUST be delayed by a random interval between zero and MAX_ANYCAST_DELAY_TIME, as defined by IPv6 Neighbor Discovery [7].

6. Defined Qtypes

The following Qtypes are defined. Qtypes 0, 2, and 3 MUST be supported by any implementation of this protocol. Qtype 4 SHOULD be supported by any implementation of this protocol on an IPv4/IPv6

dual-stack node and MAY be supported on an IPv6-only node.

Qtype Value	Qtype Name
0	NOOP
1	unused
2	Node Name
3	Node Addresses
4	IPv4 Addresses

6.1 NOOP

This NI type has no defined flags and never has a Data field. A Reply to a NI NOOP Query tells the Querier that a node with the Queried Address is up and reachable, implements the Node Information protocol, and incidentally happens to reveal whether the Queried Address was an anycast address. On transmission, the ICMPv6 Code in a NOOP Query must be set to 1 and the Code in a NOOP Reply must be 0. On reception of a NOOP Query or Reply, the Code must be ignored.

6.2 Node Name

The NI Node Name Query requests the fully-qualified or single-component name corresponding to the Subject Address or Name. The Reply Data has the following format.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              TTL                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Node Names ...                       |
+                                                               +
/                                                               /
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- TTL - MUST be zero. Any non-zero value received MUST be treated as zero.

- Node Names - The fully-qualified or single-component name or names of the Responder which correspond(s) to the Subject Address or Name, in DNS wire format [2]. Each name MUST be fully-qualified if the responder knows the domain suffix, and otherwise be a

single DNS label followed by two zero-length labels. When multiple node names are returned and more than one of them is fully-qualified, DNS name compression [2] SHOULD be used, and the offsets are counted from the first octet of the Data field. An offset of 4, for example, will point to the beginning of the first name.

The Responder must fill in the TTL field of the Reply with zero.

Only one TTL is included in the reply.

If the Responder does not know its name at all it MUST send a Reply with TTL=0 and no Node Names (or a Reply with Code=1 indicating refusal to answer). The Querier will be able to determine from the packet length that the Data field contains no names.

6.3 Node Addresses

The NI Node Addresses Query requests some set of the Responder's IPv6 unicast addresses. The Reply Data is a sequence of 128-bit IPv6 addresses, each address preceded by separate a 32-bit TTL value, with Preferred addresses listed before Deprecated addresses [7], but otherwise in no special order. Five flag bits are defined in the Query, and six in the Reply.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Qtype=3             |       unused      |G|S|L|C|A|T|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- G - If set to 1, Global-scope addresses [8] are requested.

- S - If set to 1, Site-local addresses [8] are requested.

- L - If set to 1, Link-local addresses [8] are requested.

- C - If set to 1, IPv4-compatible and IPv4-mapped addresses [3] are requested.

- A - If set to 1, all the Responder's unicast addresses (of the specified scope(s)) are requested. If 0, only those addresses are requested which belong to the interface (or any one interface) which has the Subject Address, or which are associated with the Subject Name.

- T   Defined in a Reply only, indicates that the set of addresses is incomplete for space reasons.

Flags G, S, L, C and A are copied from a Query to the corresponding Reply.

The TTL associated with each address MUST be zero.

IPv4-mapped addresses can only be returned by a Node Information proxy, since they represent addresses of IPv4-only nodes, which perforce do not implement this protocol.

6.4 IPv4 Addresses

The NI IPv4 Addresses Query requests some set of the Responder's IPv4 unicast addresses. The Reply Data is a sequence of 32-bit IPv4 addresses, each address preceded by a 32-bit TTL value. One flag bit is defined in the Query, and two in the Reply.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Qtype=4            |       unused              |A|T|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- A - If set to 1, all the Responder's unicast addresses are requested. If 0, only those addresses are requested which belong to the interface (or any one interface) which has the Subject Address.

- T   Defined in a Reply only, indicates that the set of addresses is incomplete for space reasons.

Flag A is copied from a Query to the corresponding Reply.

The TTL associated with each address MUST be zero.

6.4.1 Discussion

It is possible that a node may treat IPv4 interfaces and IPv6 interfaces as distinct, even though they are associated with the same hardware. When such a node is responding to a NI Query having a Subject Address of one type requesting the other type, and the Query has the A flag set to 0, it SHOULD consider IP interfaces, other than tunnels, associated with the same hardware as being the same interface.

7. IANA Considerations

ICMPv6 type values 139 and 140 were previously assigned by IANA for this protocol. This document defines three values of the ICMPv6 Code

field for each of these ICMPv6 Type values. Additional Code values may be defined only by IETF Consensus [12].

This document defines five values of Qtype, numbers 0 through 4. Following the policies outlined in [12], new values, and their associated Flags and Reply Data, are to be defined by IETF Consensus.

The IANA is requested to assign the IPv6 multicast prefix FF02:0:0:0:0:2::/96 for use in Node Information Queries as defined in section Section 5.

8. Security Considerations

This protocol shares the security issues of ICMPv6 that are documented in the "Security Considerations" section of [5].

This protocol has the potential of revealing information useful to a would-be attacker. An implementation of this protocol SHOULD have a default configuration which refuses to answer queries from global-scope [3] addresses.

Implementations SHOULD apply rate-limiting to NI responses to avoid being used in a denial of service attack.

The anti-spoofing Nonce does not give any protection from spoofers who can eavesdrop the Query or the Reply.

The information learned via this protocol SHOULD not be trusted for making security relevant decisions unless some other mechanisms beyond the scope of this document is used to authenticate this information.

An implementation of this protocol SHOULD provide the ability to control the dissemination of information related to IPv6 Privacy Addresses [13]. The default action of this policy SHOULD NOT provide a reponse to a Query that contains a node's Privacy Addresses.

9. Acknowledgments

Alain Durand contributed to this specification and valuable feedback and implementation experience was provided by Jun-Ichiro Hagino and Tatuya Jinmei. Other useful comments were received from Robert Elz and Keith Moore. Bob Hinden and Brian Haberman have acted as document editors during the IETF advancement process.

This document is not the first proposal of a direct query mechanism for address-to-name translation. The idea had been discussed briefly in the IPng working group and RFC 1788 [14] describes such a

mechanism for IPv4.

10. References

10.1 Normative References

[1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[2] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[3] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

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

[5] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.

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

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

[8] Hinden, R. and S. Deering, "An IPv6 Aggregatable Global Unicast Address Format", RFC 2374, July 1998.

[9] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

10.2 Informative References

[10] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[11] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[12] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[13] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.

[14] Simpson, W., "ICMP Domain Name Messages", RFC 1788, April 1995.

Authors' Addresses

Matt Crawford
Fermilab
PO Box 500
Batavia, IL 60510
US

Phone: +1 630 840 3461
Email: crawdad@fnal.gov

Brian Haberman (editor)
Johns Hopkins University Applied Physics Lab
11100 Johns Hopkins Road
Laurel, MD 20723-6099
US

Phone: +1 443 778 1319
Email: brian@innovationslab.net

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.