Network Working Group
Request for Comments: 4795
Category: Informational

B. Aboba
D. Thaler
L. Esibov
Microsoft Corporation

January 2007

Link-Local Multicast Name Resolution (LLMNR)

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

IESG Note

This document was originally intended for advancement as a Proposed Standard, but the IETF did not achieve consensus on the approach. The document has had significant review and input. At time of publication, early versions were implemented and deployed.

Abstract

The goal of Link-Local Multicast Name Resolution (LLMNR) is to enable name resolution in scenarios in which conventional DNS name resolution is not possible. LLMNR supports all current and future DNS formats, types, and classes, while operating on a separate port from DNS, and with a distinct resolver cache. Since LLMNR only operates on the local link, it cannot be considered a substitute for DNS.

Table of Contents

1. Introduction

This document discusses Link-Local Multicast Name Resolution (LLMNR), which is based on the DNS packet format and supports all current and future DNS formats, types, and classes. LLMNR operates on a separate port from the Domain Name System (DNS), with a distinct resolver cache.

Since LLMNR only operates on the local link, it cannot be considered a substitute for DNS. Link-scope multicast addresses are used to prevent propagation of LLMNR traffic across routers, potentially flooding the network. LLMNR queries can also be sent to a unicast address, as described in Section 2.4.

Propagation of LLMNR packets on the local link is considered sufficient to enable name resolution in small networks. In such networks, if a network has a gateway, then typically the network is able to provide DNS server configuration. Configuration issues are discussed in Section 3.1.

In the future, it may be desirable to consider use of multicast name resolution with multicast scopes beyond the link-scope. This could occur if LLMNR deployment is successful, the need arises for multicast name resolution beyond the link-scope, or multicast routing becomes ubiquitous. For example, expanded support for multicast name resolution might be required for mobile ad-hoc networks.

Once we have experience in LLMNR deployment in terms of administrative issues, usability, and impact on the network, it will be possible to reevaluate which multicast scopes are appropriate for use with multicast name resolution. IPv4 administratively scoped multicast usage is specified in "Administratively Scoped IP Multicast" [RFC2365].

Service discovery in general, as well as discovery of DNS servers using LLMNR in particular, is outside the scope of this document, as is name resolution over non-multicast capable media.

1.1. Requirements

In this document, several words are used to signify the requirements of the specification. 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 [RFC2119].

1.2. Terminology

This document assumes familiarity with DNS terminology defined in [RFC1035]. Other terminology used in this document includes:

Routable Address	An address other than a link-local address. This includes globally routable addresses, as well as private addresses.
Reachable	An LLMNR responder considers one of its addresses reachable over a link if it will respond to an Address Resolution Protocol (ARP) or Neighbor Discovery query for that address received on that link.
Responder	A host that listens to LLMNR queries, and responds to those for which it is authoritative.
Sender	A host that sends an LLMNR query.
UNIQUE	There are some scenarios when multiple responders may respond to the same query. There are other scenarios when only one responder may respond to a query. Names for which only a single responder is anticipated are referred to as UNIQUE. Name uniqueness is configured on the responder, and therefore uniqueness verification is the responder's responsibility.

2. Name Resolution Using LLMNR

LLMNR queries are sent to and received on port 5355. The IPv4 link-scope multicast address a given responder listens to, and to which a sender sends queries, is 224.0.0.252. The IPv6 link-scope multicast address a given responder listens to, and to which a sender sends all queries, is FF02:0:0:0:0:0:1:3.

Typically, a host is configured as both an LLMNR sender and a responder. A host MAY be configured as a sender, but not a responder. However, a host configured as a responder MUST act as a sender, if only to verify the uniqueness of names as described in Section 4. This document does not specify how names are chosen or configured. This may occur via any mechanism, including DHCPv4 [RFC2131] or DHCPv6 [RFC3315].

A typical sequence of events for LLMNR usage is as follows:

(a) An LLMNR sender sends an LLMNR query to the link-scope multicast address(es), unless a unicast query is indicated, as specified in Section 2.4.

(b) A responder responds to this query only if it is authoritative for the name in the query. A responder responds to a multicast query by sending a unicast UDP response to the sender. Unicast queries are responded to as indicated in Section 2.4.

(c) Upon reception of the response, the sender processes it.

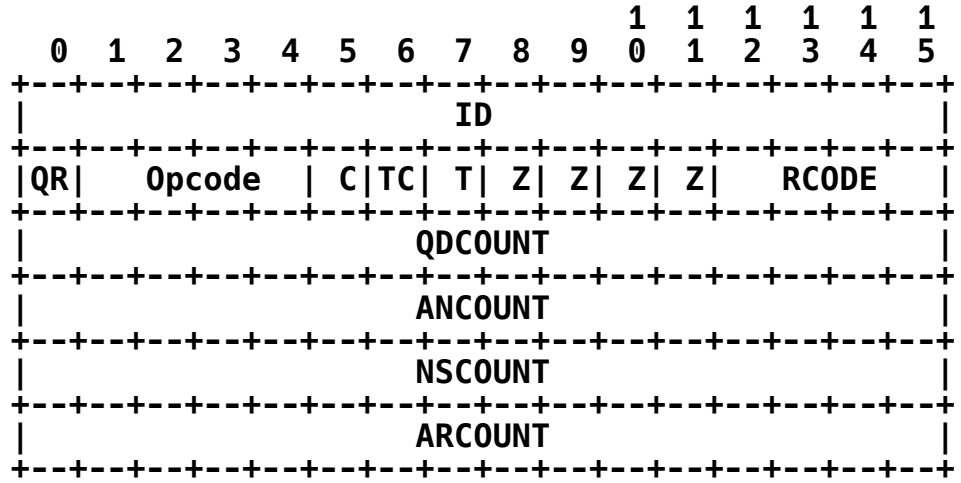
The sections that follow provide further details on sender and responder behavior.

2.1. LLMNR Packet Format

LLMNR is based on the DNS packet format defined in [RFC1035] Section 4 for both queries and responses. LLMNR implementations SHOULD send UDP queries and responses only as large as are known to be permissible without causing fragmentation. When in doubt, a maximum packet size of 512 octets SHOULD be used. LLMNR implementations MUST accept UDP queries and responses as large as the smaller of the link MTU or 9194 octets (Ethernet jumbo frame size of 9KB (9216) minus 22 octets for the header, VLAN tag and Cyclic Redundancy Check (CRC)).

2.1.1. LLMNR Header Format

LLMNR queries and responses utilize the DNS header format defined in [RFC1035] with exceptions noted below:



where:

ID A 16-bit identifier assigned by the program that generates any kind of query. This identifier is copied from the query to the response and can be used by the sender to match responses to outstanding queries. The ID field in a query SHOULD be set to a pseudo-random value. For advice on generation of pseudo-random values, please consult [RFC4086].

QR Query/Response. A 1-bit field, which, if set, indicates that the message is an LLMNR response; if clear, then the message is an LLMNR query.

OPCODE A 4-bit field that specifies the kind of query in this message. This value is set by the originator of a query and copied into the response. This specification defines the behavior of standard queries and responses (opcode value of zero). Future specifications may define the use of other opcodes with LLMNR. LLMNR senders and responders MUST support standard queries (opcode value of zero). LLMNR queries with unsupported OPCODE values MUST be silently discarded by responders.

C Conflict. When set within a query, the 'C'onflict bit indicates that a sender has received multiple LLMNR responses to this query. In an LLMNR response, if the name is considered UNIQUE, then the 'C' bit is clear; otherwise, it is set. LLMNR senders do not retransmit queries with the 'C' bit set. Responders MUST NOT respond to LLMNR queries with the 'C' bit set, but may start the uniqueness verification process, as described in Section 4.2.

TC TrunCation. The 'TC' bit specifies that this message was truncated due to length greater than that permitted on the transmission channel. The 'TC' bit MUST NOT be set in an LLMNR query and, if set, is ignored by an LLMNR responder. If the 'TC' bit is set in an LLMNR response, then the sender SHOULD resend the LLMNR query over TCP using the unicast address of the responder as the destination address. If the sender receives a response to the TCP query, then it SHOULD discard the UDP response with the TC bit set. See [RFC2181] and Section 2.4 of this specification for further discussion of the 'TC' bit.

T Tentative. The 'T'entative bit is set in a response if the responder is authoritative for the name, but has not yet verified the uniqueness of the name. A responder MUST ignore the 'T' bit in a query, if set. A response with the 'T' bit

set is silently discarded by the sender, except if it is a uniqueness query, in which case, a conflict has been detected and a responder MUST resolve the conflict as described in Section 4.1.

Z Reserved for future use. Implementations of this specification MUST set these bits to zero in both queries and responses. If these bits are set in a LLMNR query or response, implementations of this specification MUST ignore them. Since reserved bits could conceivably be used for different purposes than in DNS, implementers are advised not to enable processing of these bits in an LLMNR implementation starting from a DNS code base.

RCODE Response code. This 4-bit field is set as part of LLMNR responses. In an LLMNR query, the sender MUST set RCODE to zero; the responder ignores the RCODE and assumes it to be zero. The response to a multicast LLMNR query MUST have RCODE set to zero. A sender MUST silently discard an LLMNR response with a non-zero RCODE sent in response to a multicast query.

If an LLMNR responder is authoritative for the name in a multicast query, but an error is encountered, the responder SHOULD send an LLMNR response with an RCODE of zero, no RRs in the answer section, and the TC bit set. This will cause the query to be resent using TCP, and allow the inclusion of a non-zero RCODE in the response to the TCP query. Responding with the TC bit set is preferable to not sending a response, since it enables errors to be diagnosed. This may be required, for example, when an LLMNR query includes a TSIG RR in the additional section, and the responder encounters a problem that requires returning a non-zero RCODE. TSIG error conditions defined in [RFC2845] include a TSIG RR in an unacceptable position (RCODE=1) or a TSIG RR that does not validate (RCODE=9 with TSIG ERROR 17 (BADKEY) or 16 (BADSIG)).

Since LLMNR responders only respond to LLMNR queries for names for which they are authoritative, LLMNR responders MUST NOT respond with an RCODE of 3; instead, they should not respond at all.

LLMNR implementations MUST support EDNS0 [RFC2671] and extended RCODE values.

QDCOUNT An unsigned 16-bit integer specifying the number of entries in the question section. A sender MUST place only one question into the question section of an LLMNR query. LLMNR responders MUST silently discard LLMNR queries with QDCOUNT not equal to one. LLMNR senders MUST silently discard LLMNR responses with QDCOUNT not equal to one.

ANCOUNT An unsigned 16-bit integer specifying the number of resource records in the answer section. LLMNR responders MUST silently discard LLMNR queries with ANCOUNT not equal to zero.

NSCOUNT An unsigned 16-bit integer specifying the number of name server resource records in the authority records section. Authority record section processing is described in Section 2.9. LLMNR responders MUST silently discard LLMNR queries with NSCOUNT not equal to zero.

ARCOUNT An unsigned 16-bit integer specifying the number of resource records in the additional records section. Additional record section processing is described in Section 2.9.

2.2. Sender Behavior

A sender MAY send an LLMNR query for any legal resource record type (e.g., A, AAAA, PTR, SRV) to the link-scope multicast address. As described in Section 2.4, a sender MAY also send a unicast query.

The sender MUST anticipate receiving no responses to some LLMNR queries, in the event that no responders are available within the link-scope. If no response is received, a resolver treats it as a response that the name does not exist (RCODE=3 is returned). A sender can handle duplicate responses by discarding responses with a source IP address and ID field that duplicate a response already received.

When multiple valid LLMNR responses are received with the 'C' bit set, they SHOULD be concatenated and treated in the same manner that multiple RRs received from the same DNS server would be. However, responses with the 'C' bit set SHOULD NOT be concatenated with responses with the 'C' bit clear; instead, only the responses with the 'C' bit set SHOULD be returned. If valid LLMNR response(s) are received along with error response(s), then the error responses are silently discarded.

Since the responder may order the RRs in the response so as to indicate preference, the sender SHOULD preserve ordering in the response to the querying application.

2.3. Responder Behavior

An LLMNR response MUST be sent to the sender via unicast.

Upon configuring an IP address, responders typically will synthesize corresponding A, AAAA and PTR RRs so as to be able to respond to LLMNR queries for these RRs. An SOA RR is synthesized only when a responder has another RR in addition to the SOA RR; the SOA RR MUST NOT be the only RR that a responder has. However, in general, whether RRs are manually or automatically created is an implementation decision.

For example, a host configured to have computer name "host1" and to be a member of the "example.com" domain, with IPv4 address 192.0.2.1 and IPv6 address 2001:0DB8::1:2:3:FF:FE:4:5:6, might be authoritative for the following records:

```
host1. IN A 192.0.2.1
       IN AAAA 2001:0DB8::1:2:3:FF:FE:4:5:6

host1.example.com. IN A 192.0.2.1
       IN AAAA 2001:0DB8::1:2:3:FF:FE:4:5:6

1.2.0.192.in-addr.arpa. IN PTR host1.
       IN PTR host1.example.com.

6.0.5.0.4.0.E.F.F.F.3.0.2.0.1.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.
ip6.arpa IN PTR host1.  (line split for formatting reasons)
         IN PTR host1.example.com.
```

An LLMNR responder might be further manually configured with the name of a local mail server with an MX RR included in the "host1." and "host1.example.com." records.

In responding to queries:

(a) Responders MUST listen on UDP port 5355 on the link-scope multicast address(es) defined in Section 2, and on TCP port 5355 on the unicast address(es) that could be set as the source address(es) when the responder responds to the LLMNR query.

(b) Responders MUST direct responses to the port from which the query was sent. When queries are received via TCP, this is an inherent part of the transport protocol. For queries received by UDP, the responder MUST take note of the source port and use that as the destination port in the response. Responses MUST always be sent from the port to which they were directed.

(c) Responders MUST respond to LLMNR queries for names and addresses for which they are authoritative. This applies to both forward and reverse lookups, with the exception of queries with the 'C' bit set, which do not elicit a response.

(d) Responders MUST NOT respond to LLMNR queries for names for which they are not authoritative.

(e) Responders MUST NOT respond using data from the LLMNR or DNS resolver cache.

(f) If a responder is authoritative for a name, it MUST respond with RCODE=0 and an empty answer section, if the type of query does not match an RR that the responder has.

As an example, a host configured to respond to LLMNR queries for the name "foo.example.com." is authoritative for the name "foo.example.com.". On receiving an LLMNR query for an A RR with the name "foo.example.com.", the host authoritatively responds with an A RR(s) that contain IP address(es) in the RDATA of the resource record. If the responder has an AAAA RR, but no A RR, and an A RR query is received, the responder would respond with RCODE=0 and an empty answer section.

In conventional DNS terminology, a DNS server authoritative for a zone is authoritative for all the domain names under the zone apex except for the branches delegated into separate zones. Contrary to conventional DNS terminology, an LLMNR responder is authoritative only for the zone apex.

For example, the host "foo.example.com." is not authoritative for the name "child.foo.example.com." unless the host is configured with multiple names, including "foo.example.com." and "child.foo.example.com.". As a result, "foo.example.com." cannot respond to an LLMNR query for "child.foo.example.com." with RCODE=3 (authoritative name error). The purpose of limiting the name authority scope of a responder is to prevent complications that could be caused by coexistence of two or more hosts with the names representing child and parent (or grandparent) nodes in the DNS tree, for example, "foo.example.com." and "child.foo.example.com.".

Without the restriction on authority, an LLMNR query for an A resource record for the name "child.foo.example.com." would result in two authoritative responses: RCODE=3 (authoritative name error) received from "foo.example.com.", and a requested A record from "child.foo.example.com.". To prevent this ambiguity, LLMNR-enabled hosts could perform a dynamic update of the parent (or grandparent) zone with a delegation to a child zone; for example, a host

"child.foo.example.com." could send a dynamic update for the NS and glue A record to "foo.example.com.". However, this approach significantly complicates implementation of LLMNR and would not be acceptable for lightweight hosts.

2.4. Unicast Queries and Responses

Unicast queries SHOULD be sent when:

(a) A sender repeats a query after it received a response with the TC bit set to the previous LLMNR multicast query, or

(b) The sender queries for a PTR RR of a fully formed IP address within the "in-addr.arpa" or "ip6.arpa" zones.

Unicast LLMNR queries MUST be done using TCP and the responses MUST be sent using the same TCP connection as the query. Senders MUST support sending TCP queries, and responders MUST support listening for TCP queries. If the sender of a TCP query receives a response to that query not using TCP, the response MUST be silently discarded.

Unicast UDP queries MUST be silently discarded.

A unicast PTR RR query for an off-link address will not elicit a response, but instead, an ICMP Time to Live (TTL) or Hop Limit exceeded message will be received. An implementation receiving an ICMP message in response to a TCP connection setup attempt can return immediately, treating this as a response that no such name exists (RCODE=3 is returned). An implementation that cannot process ICMP messages MAY send multicast UDP queries for PTR RRs. Since TCP implementations will not retransmit prior to RTOmin, a considerable period will elapse before TCP retransmits multiple times, resulting in a long timeout for TCP PTR RR queries sent to an off-link destination.

2.5. "Off-Link" Detection

A sender MUST select a source address for LLMNR queries that is assigned on the interface on which the query is sent. The destination address of an LLMNR query MUST be a link-scope multicast address or a unicast address.

A responder MUST select a source address for responses that is assigned on the interface on which the query was received. The destination address of an LLMNR response MUST be a unicast address.

On receiving an LLMNR query, the responder MUST check whether it was sent to an LLMNR multicast addresses defined in Section 2. If it was sent to another multicast address, then the query MUST be silently discarded.

Section 2.4 discusses use of TCP for LLMNR queries and responses. In composing an LLMNR query using TCP, the sender MUST set the Hop Limit field in the IPv6 header and the TTL field in the IPv4 header of the response to one (1). The responder SHOULD set the TTL or Hop Limit settings on the TCP listen socket to one (1) so that SYN-ACK packets will have TTL (IPv4) or Hop Limit (IPv6) set to one (1). This prevents an incoming connection from off-link since the sender will not receive a SYN-ACK from the responder.

For UDP queries and responses, the Hop Limit field in the IPv6 header and the TTL field in the IPV4 header MAY be set to any value. However, it is RECOMMENDED that the value 255 be used for compatibility with early implementations of [RFC3927].

Implementation note:

> In the sockets API for IPv4 [POSIX], the IP_TTL and IP_MULTICAST_TTL socket options are used to set the TTL of outgoing unicast and multicast packets. The IP_RECVTTL socket option is available on some platforms to retrieve the IPv4 TTL of received packets with recvmsg(). [RFC3542] specifies similar options for setting and retrieving the IPv6 Hop Limit.

2.6. Responder Responsibilities

It is the responsibility of the responder to ensure that RRs returned in LLMNR responses MUST only include values that are valid on the local interface, such as IPv4 or IPv6 addresses valid on the local link or names defended using the mechanism described in Section 4. IPv4 Link-Local addresses are defined in [RFC3927]. IPv6 Link-Local addresses are defined in [RFC4291]. In particular:

(a) If a link-scope IPv6 address is returned in a AAAA RR, that address MUST be valid on the local link over which LLMNR is used.

(b) If an IPv4 address is returned, it MUST be reachable through the link over which LLMNR is used.

(c) If a name is returned (for example in a CNAME, MX, or SRV RR), the name MUST be resolvable on the local link over which LLMNR is used.

Where multiple addresses represent valid responses to a query, the order in which the addresses are returned is as follows:

(d) If the source address of the query is a link-scope address, then the responder SHOULD include a link-scope address first in the response, if available.

(e) If the source address of the query is a routable address, then the responder MUST include a routable address first in the response, if available.

2.7. Retransmission and Jitter

An LLMNR sender uses the timeout interval LLMNR_TIMEOUT to determine when to retransmit an LLMNR query. An LLMNR sender SHOULD either estimate the LLMNR_TIMEOUT for each interface or set a reasonably high initial timeout. Suggested constants are described in Section 7.

If an LLMNR query sent over UDP is not resolved within LLMNR_TIMEOUT, then a sender SHOULD repeat the transmission of the query in order to ensure that it was received by a host capable of responding to it. An LLMNR query SHOULD NOT be sent more than three times.

Where LLMNR queries are sent using TCP, retransmission is handled by the transport layer. Queries with the 'C' bit set MUST be sent using multicast UDP and MUST NOT be retransmitted.

An LLMNR sender cannot know in advance if a query sent using multicast will receive no response, one response, or more than one response. An LLMNR sender MUST wait for LLMNR_TIMEOUT if no response has been received, or if it is necessary to collect all potential responses, such as if a uniqueness verification query is being made. Otherwise, an LLMNR sender SHOULD consider a multicast query answered after the first response is received, if that response has the 'C' bit clear.

However, if the first response has the 'C' bit set, then the sender SHOULD wait for LLMNR_TIMEOUT + JITTER_INTERVAL in order to collect all possible responses. When multiple valid answers are received, they may first be concatenated, and then treated in the same manner that multiple RRs received from the same DNS server would. A unicast query sender considers the query answered after the first response is received.

Since it is possible for a response with the 'C' bit clear to be followed by a response with the 'C' bit set, an LLMNR sender SHOULD be prepared to process additional responses for the purposes of conflict detection, even after it has considered a query answered.

In order to avoid synchronization, the transmission of each LLMNR query and response SHOULD be delayed by a time randomly selected from the interval 0 to JITTER_INTERVAL. This delay MAY be avoided by responders responding with names that they have previously determined to be UNIQUE (see Section 4 for details).

2.8. RR TTL

The responder should insert a pre-configured TTL value in the records returned in an LLMNR response. A default value of 30 seconds is RECOMMENDED. In highly dynamic environments (such as mobile ad-hoc networks), the TTL value may need to be reduced.

Due to the TTL minimalization necessary when caching an RRset, all TTLs in an RRset MUST be set to the same value.

2.9. Use of the Authority and Additional Sections

Unlike the DNS, LLMNR is a peer-to-peer protocol and does not have a concept of delegation. In LLMNR, the NS resource record type may be stored and queried for like any other type, but it has no special delegation semantics as it does in the DNS. Responders MAY have NS records associated with the names for which they are authoritative, but they SHOULD NOT include these NS records in the authority sections of responses.

Responders SHOULD insert an SOA record into the authority section of a negative response, to facilitate negative caching as specified in [RFC2308]. The TTL of this record is set from the minimum of the MINIMUM field of the SOA record and the TTL of the SOA itself, and indicates how long a resolver may cache the negative answer. The owner name of the SOA record (MNAME) MUST be set to the query name. The RNAME, SERIAL, REFRESH, RETRY, and EXPIRE values MUST be ignored by senders. Negative responses without SOA records SHOULD NOT be cached.

In LLMNR, the additional section is primarily intended for use by EDNS0, TSIG, and SIG(0). As a result, unless the 'C' bit is set, senders MAY only include pseudo RR-types in the additional section of a query; unless the 'C' bit is set, responders MUST ignore the additional section of queries containing other RR types.

In queries where the 'C' bit is set, the sender SHOULD include the conflicting RRs in the additional section. Since conflict notifications are advisory, responders SHOULD log information from the additional section, but otherwise MUST ignore the additional section.

Senders MUST NOT cache RRs from the authority or additional section of a response as answers, though they may be used for other purposes, such as negative caching.

3. Usage Model

By default, an LLMNR sender SHOULD send LLMNR queries only for single-label names. Stub resolvers supporting both DNS and LLMNR SHOULD avoid sending DNS queries for single-label names, in order to reduce unnecessary DNS queries. An LLMNR sender SHOULD NOT be enabled to send a query for any name, except where security mechanisms (described in Section 5.3) can be utilized. An LLMNR query SHOULD only be sent for the originally requested name; a searchlist is not used to form additional LLMNR queries.

LLMNR is a peer-to-peer name resolution protocol that is not intended as a replacement for DNS; rather, it enables name resolution in scenarios in which conventional DNS name resolution is not possible. Where LLMNR security is not enabled as described in Section 5.3, if LLMNR is given higher priority than DNS among the enabled name resolution mechanisms, this would allow the LLMNR cache, once poisoned, to take precedence over the DNS cache. As a result, use of LLMNR as a primary name resolution mechanism is NOT RECOMMENDED.

Instead, it is recommended that LLMNR be utilized as a secondary name resolution mechanism, for use in situations where hosts are not configured with the address of a DNS server, where the DNS server is unavailable or unreachable, where there is no DNS server authoritative for the name of a host, or where the authoritative DNS server does not have the desired RRs.

When LLMNR is configured as a secondary name resolution mechanism, LLMNR queries SHOULD only be sent when all of the following conditions are met:

(1) No manual or automatic DNS configuration has been performed. If DNS server address(es) have been configured, a host SHOULD attempt to reach DNS servers over all protocols on which DNS server address(es) are configured, prior to sending LLMNR queries. For dual-stack hosts configured with DNS server address(es) for one protocol but not another, this implies that DNS queries SHOULD be sent over the protocol configured with a DNS server, prior to sending LLMNR queries.

(2) All attempts to resolve the name via DNS on all interfaces have failed after exhausting the searchlist. This can occur because DNS servers did not respond, or because they responded to DNS queries with RCODE=3 (Authoritative Name Error) or RCODE=0, and an empty answer section. Where a single resolver call generates DNS queries for A and AAAA RRs, an implementation MAY choose not to send LLMNR queries if any of the DNS queries is successful.

Where LLMNR is used as a secondary name resolution mechanism, its usage is in part determined by the behavior of DNS resolver implementations; robust resolver implementations are more likely to avoid unnecessary LLMNR queries.

[RFC1536] describes common DNS implementation errors and fixes. If the proposed fixes are implemented, unnecessary LLMNR queries will be reduced substantially, so implementation of [RFC1536] is recommended.

For example, [RFC1536] Section 1 describes issues with retransmission and recommends implementation of a retransmission policy based on round trip estimates, with exponential back-off. [RFC1536] Section 4 describes issues with failover, and recommends that resolvers try another server when they don't receive a response to a query. These policies are likely to avoid unnecessary LLMNR queries.

[RFC1536] Section 3 describes zero answer bugs, which if addressed will also reduce unnecessary LLMNR queries.

[RFC1536] Section 6 describes name error bugs and recommended searchlist processing that will reduce unnecessary RCODE=3 (authoritative name) errors, thereby also reducing unnecessary LLMNR queries.

As noted in [DNSPerf], a significant fraction of DNS queries do not receive a response, or result in negative responses due to missing inverse mappings or NS records that point to nonexistent or inappropriate hosts. Therefore, a reduction in missing records can prevent many unnecessary LLMNR queries.

3.1. LLMNR Configuration

LLMNR usage MAY be configured manually or automatically on a per-interface basis. By default, LLMNR responders SHOULD be enabled on all interfaces, at all times. Where this is considered undesirable, LLMNR SHOULD be disabled, so that hosts will neither listen on the link-scope multicast address, nor will they send queries to that address.

Where DHCPv4 or DHCPv6 is implemented, DHCP options can be used to configure LLMNR on an interface. The LLMNR Enable Option, described in [LLMNREnable], can be used to explicitly enable or disable use of LLMNR on an interface. The LLMNR Enable Option does not determine whether, or in which order, DNS itself is used for name resolution. The order in which various name resolution mechanisms should be used can be specified using the Name Service Search Option (NSSO) for DHCP [RFC2937], using the LLMNR Enable Option code carried in the NSSO data.

In situations where LLMNR is configured as a secondary name resolution protocol on a dual-stack host, behavior will be governed by both IPv4 and IPv6 configuration mechanisms. Since IPv4 and IPv6 utilize distinct configuration mechanisms, it is possible for a dual-stack host to be configured with the address of a DNS server over IPv4, while remaining unconfigured with a DNS server suitable for use over IPv6.

In these situations, a dual-stack host will send AAAA queries to the configured DNS server over IPv4. However, an IPv6-only host unconfigured with a DNS server suitable for use over IPv6 will be unable to resolve names using DNS. Automatic IPv6 DNS configuration mechanisms (such as [RFC3315] and [DNSDisc]) are not yet widely deployed, and not all DNS servers support IPv6. Therefore, lack of IPv6 DNS configuration may be a common problem in the short term, and LLMNR may prove useful in enabling link-local name resolution over IPv6.

Where a DHCPv4 server is available but not a DHCPv6 server [RFC3315], IPv6-only hosts may not be configured with a DNS server. Where there is no DNS server authoritative for the name of a host or the authoritative DNS server does not support dynamic client update over IPv6 or DHCPv6-based dynamic update, then an IPv6-only host will not be able to do DNS dynamic update, and other hosts will not be able to resolve its name.

For example, if the configured DNS server responds to an AAAA RR query sent over IPv4 or IPv6 with an authoritative name error (RCODE=3) or RCODE=0 and an empty answer section, then an AAAA RR query sent using LLMNR over IPv6 may be successful in resolving the name of an IPv6-only host on the local link.

Similarly, if a DHCPv4 server is available providing DNS server configuration, and DNS server(s) exist which are authoritative for the A RRs of local hosts and support either dynamic client update over IPv4 or DHCPv4-based dynamic update, then the names of local IPv4 hosts can be resolved over IPv4 without LLMNR. However, if no DNS server is authoritative for the names of local hosts, or the authoritative DNS server(s) do not support dynamic update, then LLMNR enables link-local name resolution over IPv4.

It is possible that DNS configuration mechanisms will go in and out of service. In these circumstances, it is possible for hosts within an administrative domain to be inconsistent in their DNS configuration.

For example, where DHCP is used for configuring DNS servers, one or more DHCP servers can fail. As a result, hosts configured prior to the outage will be configured with a DNS server, while hosts configured after the outage will not. Alternatively, it is possible for the DNS configuration mechanism to continue functioning while configured DNS servers fail.

An outage in the DNS configuration mechanism may result in hosts continuing to use LLMNR even once the outage is repaired. Since LLMNR only enables link-local name resolution, this represents a degradation in capabilities. As a result, hosts without a configured DNS server may wish to periodically attempt to obtain DNS configuration if permitted by the configuration mechanism in use. In the absence of other guidance, a default retry interval of one (1) minute is RECOMMENDED.

4. Conflict Resolution

By default, a responder SHOULD be configured to behave as though its name is UNIQUE on each interface on which LLMNR is enabled. However, it is also possible to configure multiple responders to be authoritative for the same name. For example, multiple responders MAY respond to a query for an A or AAAA type record for a cluster name (assigned to multiple hosts in the cluster).

To detect duplicate use of a name, an administrator can use a name resolution utility that employs LLMNR and lists both responses and responders. This would allow an administrator to diagnose behavior and potentially intervene and reconfigure LLMNR responders that should not be configured to respond to the same name.

4.1. Uniqueness Verification

Prior to sending an LLMNR response with the 'T' bit clear, a responder configured with a UNIQUE name MUST verify that there is no other host within the scope of LLMNR query propagation that is authoritative for the same name on that interface.

Once a responder has verified that its name is UNIQUE, if it receives an LLMNR query for that name with the 'C' bit clear, it MUST respond with the 'T' bit clear. Prior to verifying that its name is UNIQUE, a responder MUST set the 'T' bit in responses.

Uniqueness verification is carried out when the host:

- starts up or is rebooted
- wakes from sleep (if the network interface was inactive during sleep)
- is configured to respond to LLMNR queries on an interface enabled for transmission and reception of IP traffic
- is configured to respond to LLMNR queries using additional UNIQUE resource records
- verifies the acquisition of a new IP address and configuration on an interface

To verify uniqueness, a responder MUST send an LLMNR query with the 'C' bit clear, over all protocols on which it responds to LLMNR queries (IPv4 and/or IPv6). It is RECOMMENDED that responders verify uniqueness of a name by sending a query for the name with type='ANY'.

If no response is received, the sender retransmits the query, as specified in Section 2.7. If a response is received, the sender MUST check if the source address matches the address of any of its interfaces; if so, then the response is not considered a conflict, since it originates from the sender. To avoid triggering conflict detection, a responder that detects that it is connected to the same link on multiple interfaces SHOULD set the 'C' bit in responses.

If a response is received with the 'T' bit clear, the responder MUST NOT use the name in response to LLMNR queries received over any protocol (IPv4 or IPv6). If a response is received with the 'T' bit set, the responder MUST check if the source IP address in the response is lexicographically smaller than the source IP address in the query. If so, the responder MUST NOT use the name in response to LLMNR queries received over any protocol (IPv4 or IPv6). For the purpose of uniqueness verification, the contents of the answer section in a response is irrelevant.

Periodically carrying out uniqueness verification in an attempt to detect name conflicts is not necessary, wastes network bandwidth, and may actually be detrimental. For example, if network links are joined only briefly, and are separated again before any new communication is initiated, temporary conflicts are benign and no forced reconfiguration is required. LLMNR responders SHOULD NOT periodically attempt uniqueness verification.

4.2. Conflict Detection and Defense

Hosts on disjoint network links may configure the same name for use with LLMNR. If these separate network links are later joined or bridged together, then there may be multiple hosts that are now on the same link, trying to use the same name.

In order to enable ongoing detection of name conflicts, when an LLMNR sender receives multiple LLMNR responses to a query, it MUST check if the 'C' bit is clear in any of the responses. If so, the sender

SHOULD send another query for the same name, type, and class, this time with the 'C' bit set, with the potentially conflicting resource records included in the additional section.

Queries with the 'C' bit set are considered advisory, and responders MUST verify the existence of a conflict before acting on it. A responder receiving a query with the 'C' bit set MUST NOT respond.

If the query is for a UNIQUE name, then the responder MUST send its own query for the same name, type, and class, with the 'C' bit clear. If a response is received, the sender MUST check if the source address matches the address of any of its interfaces; if so, then the response is not considered a conflict, since it originates from the sender. To avoid triggering conflict detection, a responder that detects that it is connected to the same link on multiple interfaces SHOULD set the 'C' bit in responses.

An LLMNR responder MUST NOT ignore conflicts once detected, and SHOULD log them. Upon detecting a conflict, an LLMNR responder MUST immediately stop using the conflicting name in response to LLMNR queries received over any supported protocol, if the source IP address in the response is lexicographically smaller than the source IP address in the uniqueness verification query.

After stopping the use of a name, the responder MAY elect to configure a new name. However, since name reconfiguration may be disruptive, this is not required, and a responder may have been configured to respond to multiple names so that alternative names may already be available. A host that has stopped the use of a name may attempt uniqueness verification again after the expiration of the TTL of the conflicting response.

4.3. Considerations for Multiple Interfaces

A multi-homed host may elect to configure LLMNR on only one of its active interfaces. In many situations, this will be adequate. However, should a host need to configure LLMNR on more than one of its active interfaces, there are some additional precautions it MUST take. Implementers who are not planning to support LLMNR on multiple interfaces simultaneously may skip this section.

Where a host is configured to issue LLMNR queries on more than one interface, each interface maintains its own independent LLMNR resolver cache, containing the responses to LLMNR queries.

A multi-homed host checks the uniqueness of UNIQUE records as described in Section 4. The situation is illustrated in Figure 1.

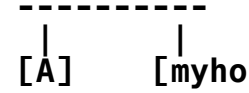


Figure 1. Link-scope name conflict

In this situation, the multi-homed myhost will probe for, and defend, its host name on both interfaces. A conflict will be detected on one interface, but not the other. The multi-homed myhost will not be able to respond with a host RR for "myhost" on the interface on the right (see Figure 1). The multi-homed host may, however, be configured to use the "myhost" name on the interface on the left.

Since names are only unique per link, hosts on different links could be using the same name. If an LLMNR client sends queries over multiple interfaces, and receives responses from more than one, the result returned to the client is defined by the implementation. The situation is illustrated in Figure 2.

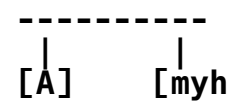


Figure 2. Off-segment name conflict

If host myhost is configured to use LLMNR on both interfaces, it will send LLMNR queries on both interfaces. When host myhost sends a query for the host RR for name "A", it will receive a response from hosts on both interfaces.

Host myhost cannot distinguish between the situation shown in Figure 2, and that shown in Figure 3, where no conflict exists.

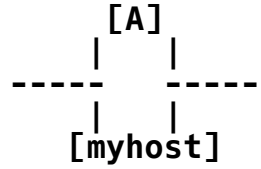


Figure 3. Multiple paths to same host

This illustrates that the proposed name conflict-resolution mechanism does not support detection or resolution of conflicts between hosts on different links. This problem can also occur with DNS when a multi-homed host is connected to two different networks with separated name spaces. It is not the intent of this document to address the issue of uniqueness of names within DNS.

4.4. API Issues

[RFC3493] provides an API that can partially solve the name ambiguity problem for applications written to use this API, since the sockaddr_in6 structure exposes the scope within which each scoped address exists, and this structure can be used for both IPv4 (using v4-mapped IPv6 addresses) and IPv6 addresses.

Following the example in Figure 2, an application on 'myhost' issues the request getaddrinfo("A", ...) with ai_family=AF_INET6 and ai_flags=AI_ALL|AI_V4MAPPED. LLMNR queries will be sent from both interfaces, and the resolver library will return a list containing multiple addrinfo structures, each with an associated sockaddr_in6

structure. This list will thus contain the IPv4 and IPv6 addresses of both hosts responding to the name 'A'. Link-local addresses will have a sin6_scope_id value that disambiguates which interface is used to reach the address. Of course, to the application, Figures 2 and 3 are still indistinguishable, but this API allows the application to communicate successfully with any address in the list.

5. Security Considerations

LLMNR is a peer-to-peer name resolution protocol designed for use on the local link. While LLMNR limits the vulnerability of responders to off-link senders, it is possible for an off-link responder to reach a sender.

In scenarios such as public "hotspots", attackers can be present on the same link. These threats are most serious in wireless networks, such as IEEE 802.11, since attackers on a wired network will require physical access to the network, while wireless attackers may mount attacks from a distance. Link-layer security, such as [IEEE-802.11i], can be of assistance against these threats if it is available.

This section details security measures available to mitigate threats from on and off-link attackers.

5.1. Denial of Service

Attackers may take advantage of LLMNR conflict detection by allocating the same name, denying service to other LLMNR responders, and possibly allowing an attacker to receive packets destined for other hosts. By logging conflicts, LLMNR responders can provide forensic evidence of these attacks.

An attacker may spoof LLMNR queries from a victim's address in order to mount a denial of service attack. Responders setting the IPv6 Hop Limit or IPv4 TTL field to a value larger than one in an LLMNR UDP response may be able to reach the victim across the Internet.

While LLMNR responders only respond to queries for which they are authoritative, and LLMNR does not provide wildcard query support, an LLMNR response may be larger than the query, and an attacker can generate multiple responses to a query for a name used by multiple responders. A sender may protect itself against unsolicited responses by silently discarding them.

5.2. Spoofing

LLMNR is designed to prevent reception of queries sent by an off-link attacker. LLMNR requires that responders receiving UDP queries check that they are sent to a link-scope multicast address. However, it is possible that some routers may not properly implement link-scope multicast, or that link-scope multicast addresses may leak into the multicast routing system. To prevent successful setup of TCP connections by an off-link sender, responders receiving a TCP SYN reply with a TCP SYN-ACK with TTL set to one (1).

While it is difficult for an off-link attacker to send an LLMNR query to a responder, it is possible for an off-link attacker to spoof a response to a query (such as an A or AAAA query for a popular Internet host), and by using a TTL or Hop Limit field larger than one (1), for the forged response to reach the LLMNR sender. Since the forged response will only be accepted if it contains a matching ID field, choosing a pseudo-random ID field within queries provides some protection against off-link responders.

When LLMNR is utilized as a secondary name resolution service, queries can be sent when DNS server(s) do not respond. An attacker can execute a denial of service attack on the DNS server(s), and then poison the LLMNR cache by responding to an LLMNR query with incorrect information. As noted in "Threat Analysis of the Domain Name System (DNS)" [RFC3833], these threats also exist with DNS, since DNS-response spoofing tools are available that can allow an attacker to respond to a query more quickly than a distant DNS server. However, while switched networks or link-layer security may make it difficult for an on-link attacker to snoop unicast DNS queries, multicast LLMNR queries are propagated to all hosts on the link, making it possible for an on-link attacker to spoof LLMNR responses without having to guess the value of the ID field in the query.

Since LLMNR queries are sent and responded to on the local link, an attacker will need to respond more quickly to provide its own response prior to arrival of the response from a legitimate responder. If an LLMNR query is sent for an off-link host, spoofing a response in a timely way is not difficult, since a legitimate response will never be received.

This vulnerability can be reduced by limiting use of LLMNR to resolution of single-label names as described in Section 3, or by implementation of authentication (see Section 5.3).

5.3. Authentication

LLMNR is a peer-to-peer name resolution protocol and, as a result, is often deployed in situations where no trust model can be assumed. Where a pre-arranged security configuration is possible, the following security mechanisms may be used:

(a) LLMNR implementations MAY support TSIG [RFC2845] and/or SIG(0) [RFC2931] security mechanisms. "DNS Name Service based on Secure Multicast DNS for IPv6 Mobile Ad Hoc Networks" [LLMNRSec] describes the use of TSIG to secure LLMNR, based on group keys. While group keys can be used to demonstrate membership in a group, they do not protect against forgery by an attacker that is a member of the group.

(b) IPsec Encapsulating Security Payload (ESP) with a NULL encryption algorithm MAY be used to authenticate unicast LLMNR queries and responses, or LLMNR responses to multicast queries. In a small network without a certificate authority, this can be most easily accomplished through configuration of a group pre-shared key for trusted hosts. As with TSIG, this does not protect against forgery by an attacker with access to the group pre-shared key.

(c) LLMNR implementations MAY support DNSSEC [RFC4033]. In order to support DNSSEC, LLMNR implementations MAY be configured with trust anchors, or they MAY make use of keys obtained from DNS queries. Since LLMNR does not support "delegated trust" (CD or AD bits), LLMNR implementations cannot make use of DNSSEC unless they are DNSSEC-aware and support validation. Unlike approaches [a] or [b], DNSSEC permits a responder to demonstrate ownership of a name, not just membership within a trusted group. As a result, it enables protection against forgery.

5.4. Cache and Port Separation

In order to prevent responses to LLMNR queries from polluting the DNS cache, LLMNR implementations MUST use a distinct, isolated cache for LLMNR on each interface. LLMNR operates on a separate port from DNS, reducing the likelihood that a DNS server will unintentionally respond to an LLMNR query.

If a DNS server is running on a host that supports LLMNR, the LLMNR responder on that host MUST respond to LLMNR queries only for the RRSets relating to the host on which the server is running, but MUST NOT respond for other records for which the DNS server is authoritative. DNS servers MUST NOT send LLMNR queries in order to resolve DNS queries.

6. IANA Considerations

This specification creates a new namespace: the LLMNR namespace.

In order to avoid creating any new administrative procedures, administration of the LLMNR namespace will piggyback on the administration of the DNS namespace.

The rights to use a fully qualified domain name (FQDN) within LLMNR are obtained by acquiring the rights to use that name within DNS. Those wishing to use an FQDN within LLMNR should first acquire the rights to use the corresponding FQDN within DNS. Using an FQDN within LLMNR without ownership of the corresponding name in DNS creates the possibility of conflict and therefore is discouraged.

LLMNR responders may self-allocate a name within the single-label namespace first defined in [RFC1001]. Since single-label names are not unique, no registration process is required.

7. Constants

The following timing constants are used in this protocol; they are not intended to be user configurable.

JITTER_INTERVAL	100 ms
LLMNR_TIMEOUT	1 second (if set statically on all interfaces)
	100 ms (IEEE 802 media, including IEEE 802.11)

8. References

8.1. Normative References

[RFC1001] NetBIOS Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, and End-to-End Services Task Force, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods", STD 19, RFC 1001, March 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

8.2. Informative References

[DNSPerf] Jung, J., et al., "DNS Performance and the Effectiveness of Caching", IEEE/ACM Transactions on Networking, Volume 10, Number 5, pp. 589, October 2002.

[DNSDisc] Durand, A., Hagino, I., and D. Thaler, "Well known site local unicast addresses to communicate with recursive DNS servers", Work in Progress, October 2002.

[IEEE-802.11i] Institute of Electrical and Electronics Engineers, "Supplement to Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security", IEEE 802.11i, July 2004.

[LLMNREnable] Guttman, E., "DHCP LLMNR Enable Option", Work in Progress, April 2002.

[LLMNRSec] Jeong, J., Park, J. and H. Kim, "DNS Name Service based on Secure Multicast DNS for IPv6 Mobile Ad Hoc Networks", ICACT 2004, Phoenix Park, Korea, February 9-11, 2004.

[POSIX] IEEE Std. 1003.1-2001 Standard for Information Technology -- Portable Operating System Interface (POSIX). Open Group Technical Standard: Base Specifications, Issue 6, December 2001. ISO/IEC 9945:2002. http://www.opengroup.org/austin

[RFC1536] Kumar, A., Postel, J., Neuman, C., Danzig, P., and S. Miller, "Common DNS Implementation Errors and Suggested Fixes", RFC 1536, October 1993.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2365] Meyer, D., "Administratively Scoped IP Multicast", BCP 23, RFC 2365, July 1998.

[RFC2937] Smith, C., "The Name Service Search Option for DHCP", RFC 2937, September 2000.

[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.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.

[RFC3542] Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei, "Advanced Sockets Application Program Interface (API) for IPv6", RFC 3542, May 2003.

[RFC3833] Atkins, D. and R. Austein, "Threat Analysis of the Domain Name System (DNS)", RFC 3833, August 2004.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4086] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

9. Acknowledgments

This work builds upon original work done on multicast DNS by Bill Manning and Bill Woodcock. Bill Manning's work was funded under DARPA grant #F30602-99-1-0523. The authors gratefully acknowledge their contribution to the current specification. Constructive input has also been received from Mark Andrews, Rob Austein, Randy Bush, Stuart Cheshire, Ralph Droms, Robert Elz, James Gilroy, Olafur Gudmundsson, Andreas Gustafsson, Erik Guttman, Myron Hattig, Christian Huitema, Olaf Kolkman, Mika Liljeberg, Keith Moore, Tomohide Nagashima, Thomas Narten, Erik Nordmark, Markku Savela, Mike St. Johns, Sander van Valkenburg, and Brian Zill.

Authors' Addresses

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 706 6605
EMail: bernarda@microsoft.com

Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 703 8835
EMail: dthaler@microsoft.com

Levon Esibov
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

EMail: levone@microsoft.com

Full Copyright Statement

Intellectual Property

Acknowledgement

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