Internet Draft
draft-irtf-rrg-ilnp-eng-06.txt
Expires: 10 JAN 2013
Category: Experimental

RJ Atkinson
Consultant
SN Bhatti
U. St Andrews

10 July 2012

ILNP Engineering Considerations
draft-irtf-rrg-ilnp-eng-06.txt

Status of this Memo

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/1id-abstracts.html

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

This document is not on the IETF standards-track and does not specify any level of standard. This document merely provides information for the Internet community.

The ILNP document set has had extensive review within the IRTF Routing Research Group. ILNP is one of the recommendations made by the RG Chairs. Separately, various refereed research papers on ILNP have also been published during this decade. So the ideas contained herein have had much broader review than IRTF Routing RG. The views in this document were considered controversial by the Routing RG, but the RG reached a consensus that the document still should be published. The Routing RG has had remarkably little consensus on anything, so virtually all Routing RG outputs are considered controversial.

Abstract

This document describes common (i.e. version independent) engineering details for the Identifier-Locator Network Protocol (ILNP), which is an experimental, evolutionary enhancement to IP. This document is a product of the IRTF Routing RG.

Table of Contents

1. Introduction .....?
2. ILNP Identifiers.....?
3. Encoding of Identifiers and Locators for ILNPv6.....?
4. Transport Layer Changes.....?
5. ILNP Communication Cache (ILCC).....?
6. Handling Location/Connectivity Changes.....?
7. Subnetting.....?
8. DNS Considerations.....?
9. IP Security for ILNP.....?
10. Backwards Compatibility Incremental Deployment.....?
11. Security Considerations.....?
12. Privacy Considerations.....?
13. Operational Considerations.....?

14. Referrals and Application Programming Interfaces.....?
15. IANA Considerations.....?
16. References.....?

1. INTRODUCTION

At present, the Internet research and development community are exploring various approaches to evolving the Internet Architecture to solve a variety of issues including, but not limited to, scalability of inter-domain routing [RFC4984]. A wide range of other issues (e.g. site multi-homing, node multi-homing, site/subnet mobility, node mobility) are also active concerns at present. Several different classes of evolution are being considered by the Internet research & development community. One class is often called "Map and Encapsulate", where traffic would be mapped and then tunnelled through the inter-domain core of the Internet. Another class being considered is sometimes known as "Identifier/Locator Split". This document relates to a proposal that is in the latter class of evolutionary approaches.

The Identifier Locator Network Protocol (ILNP) is an experimental network protocol that provides evolutionary enhancements to IP. ILNP is backwards-compatible with IP and also is incrementally deployable. The best starting point for learning about ILNP is the ILNP Architectural Description, which includes a document roadmap [ILNP-ARCH].

1.1 Document roadmap

This document describes engineering and implementation considerations that are common to both ILNPv4 and ILNPv6.

The ILNP architecture can have more than one engineering instantiation. For example, one can imagine a "clean-slate" engineering design based on the ILNP architecture. In separate documents, we describe two specific engineering instances of ILNP. The term ILNPv6 refers precisely to an instance of ILNP that is based upon, and backwards compatible with, IPv6. The term ILNPv4 refers precisely to an instance of ILNP that is based upon, and backwards compatible with, IPv4.

Many engineering aspects common to both ILNPv4 and ILNPv6 are described in [ILNP-ENG]. A full engineering specification for either ILNPv6 or ILNPv4 is beyond the scope of this document.

Readers are referred to other related ILNP documents for details not described here:

a) [ILNP-ARCH] is the main architectural description of ILNP, including the concept of operations.

b) [ILNP-DNS] defines additional DNS resource records that support ILNP.

c) [ILNP-ICMPv6] defines a new ICMPv6 Locator Update message used by an ILNP node to inform its correspondent nodes of any changes to its set of valid Locators.

d) [ILNP-NONCEv6] defines a new IPv6 Nonce Destination Option used by ILNPv6 nodes (1) to indicate to ILNP correspondent nodes (by inclusion within the initial packets of an ILNP session) that the node is operating in the ILNP mode and (2) to prevent off-path attacks against ILNP ICMP messages. This Nonce is used, for example, with all ILNP ICMPv6 Locator Update messages that are exchanged among ILNP correspondent nodes.

e) [ILNP-ICMPv4] defines a new ICMPv4 Locator Update message used by an ILNP node to inform its correspondent nodes of any changes to its set of valid Locators.

f) [ILNP-v4OPTS] defines a new IPv4 Nonce Option used by ILNPv4 nodes to carry a security nonce to prevent off-path attacks against ILNP ICMP messages and also defines a new IPv4 Identifier Option used by ILNPv4 nodes.

g) [ILNP-ARP] describes extensions to ARP for use with ILNPv4.

h) [ILNP-ADV] describes optional engineering and deployment functions for ILNP. These are not required for the operation or use of ILNP and are provided as additional options.

1.2 Terminology

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

Several technical terms (e.g., "ILNP session") that are used by this document are defined in [ILNP-ARCH]. It is strongly recommended that one read [ILNP-ARCH] before reading this document.

2. ILNP IDENTIFIERS

All ILNP nodes must have at least one Node Identifier (or just "Identifier") value. However, there are various options for generating those Identifier values. We describe in this section the relevant engineering issues related to Identifier generation and usage.

Note well that ILNP Node Identifiers name an ILNP-capable node, and are NOT bound to a specific interface of that node. So a given ILNP Node Identifier is valid on all active interfaces of the node to which that ILNP Identifier is bound. This is true even if the bits used to form the Identifier value happened to be taken from a specific interface as an engineering convenience.

2.1 Syntax

ILNP Identifiers are always unsigned 64-bit strings, and may be realised as 64-bit unsigned integers. Both ILNPv4 and ILNPv6 use the Modified EUI-64 [IEEE-EUI] syntax that is used by IPv6 Interface Identifiers [RFC4291, Sec 2.5.1], as shown in Figure 2.1.

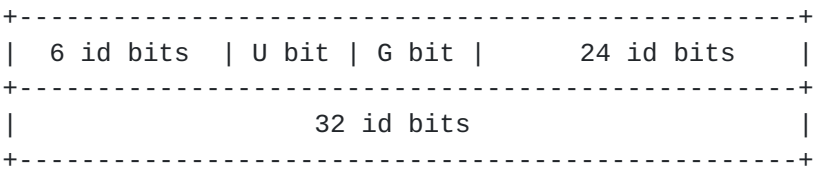


Figure 2.1. Node Identifier format as used for IPv6, using the same syntax as in RFC4291 Sec 2.5.1.

That syntax contains two special reserved bit flags. One flag (the U bit) indicates whether the value has "universal" (i.e. global) scope (1) or "local" (0) scope. The other flag (the G bit) indicates whether the value is an "individual" address (1) or "group" (i.e multicast) (0) address.

However, this format does allow other values to be set, by use of administrative or other policy control, as required, by setting the U bit to "local".

2.1 Default values for an Identifier

By default, this value, including the U bit and G bit, are set as described in Sec 2.5.1 of RFC4291 [RFC4291]. Where no other value of Identifier is available for an ILNP node, this is the value that MUST be used.

Because ILNP Identifiers might have local scope, and also to

handle the case where two nodes at different locations happen to be using the same global scope Identifier (e.g. due to a manufacturing fault in a network chipset or card), implementers must be careful in how ILNP Identifiers are handled within an end system's networking implementation. Some details are discussed in Section 4 below.

2.2 Local-scoped Identifier values

ILNP Identifiers for a node also MAY have the Scope bit of the Node Identifier set to "local"" scope. Locally unique identifiers MAY be Cryptographically Generated, created following the procedures used for IPv6 Cryptographically Generated Addresses (CGAs) [RFC3972] [RFC4581] [RFC4982].

Also, locally unique identifiers MAY be used to create the ILNP equivalent to the Privacy Extensions for IPv6, generating ILNP Identifiers following the procedures used for IPv6 [RFC4941].

2.3 Multicast Identifiers

An ILNP Identifier with the G bit set to "group" names an ILNP multicast group, while an ILNP Identifier with the G bit set to "individual" names an individual ILNP node. However, this usage of multicast for Identifiers for ILNP is currently undefined: ILNP uses IPv6 multicast for ILNPv6 and IPv4 multicast for ILNPv4 and uses the multicast address formats defined as appropriate.

The use of multicast Identifiers and design of an enhanced multicast capability for ILNPv6 and ILNPv4 is currently work in progress.

2.4 Administration of Identifier values

Note that just as IPv6 does not need global, centralised administrative management of its interface identifiers, so ILNPv6 does not need global, centralised administrative management of the NID values.

3.0 ENCODING OF IDENTIFIERS AND LOCATORS FOR ILNPv6

3.1 Encoding of I and L values

With ILNPv6, the Identifier and Locator values within a packet are encoded in the the existing space for the IPv6 address. In general, the ILNPv6 Locator has the same syntax and semantics as the current

IPv6 unicast routing prefix, as shown in Figure 3.1:

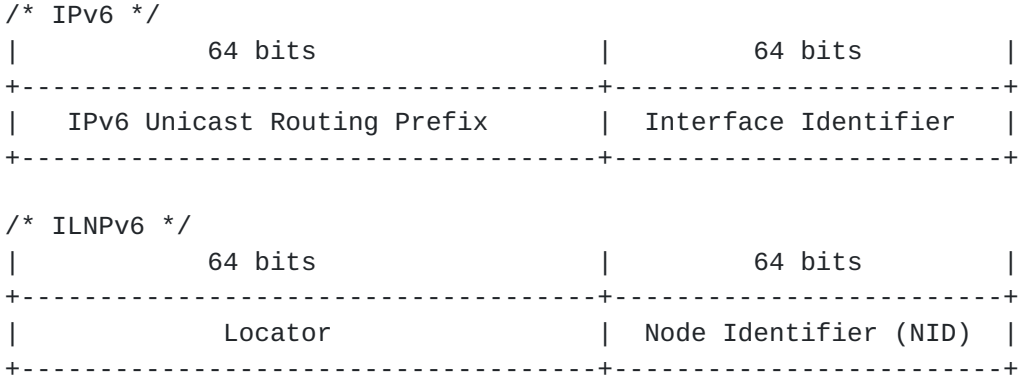


Figure 3.1 The general format of encoding of I/NID and L values for ILNPv6 into theIPv6 address bits.

The syntactical structure of the IPv6 address spaces remains as given in section 2.5.4 of [RFC4291], and an example is shown in Figure 3.2, which is based in part on [RFC3177].

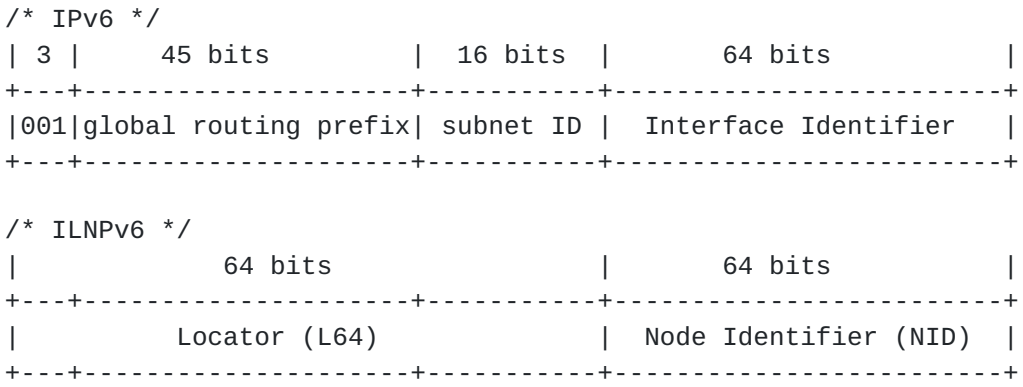


Figure 3.2: Example of IPv6 address format as used in ILNPv6. The global routing prefix bits and subnet ID bits above are as for [RFC3177], but could be different, e.g. as for [RFC6177].

The ILNPv6 Locator uses the upper 64-bits of the 128-bit IPv6 address space. It has the same syntax and semantics as today's IPv6 routing prefix. So, an ILNPv6 packet carrying a Locator value can be used just like an IPv6 packet today as far as core routers are concerned.

The example in Figure 3.2 happens to use a /48 prefix, as was recommended by [RFC3177]. However, more recent advice is that prefixes need not be fixed at /48 and could be up to /64 [RFC6177]. This change, however, does not impact the syntax or semantics of the Locator value.

The ILNPv6 Identifier value uses the lower 64-bits of the 128-bit IPv6 address. It has the same syntax as an IPv6 identifier, but different semantics. This provides a fixed-length non-topological name for a node. Identifiers are bound to nodes, not to interfaces of a node. All ILNP Identifiers MUST comply with the modified EUI-64 syntax already specified for IPv6's "Interface Identifier" values, as described in Section 2.1.

IEEE EUI-64 Identifiers can have either global-scope or local-scope. So ILNP Identifiers also can have either global-scope or local-scope. A reserved bit in the modified EUI-64 syntax clearly indicates whether a given Identifier has global-scope or local-scope. A node is not required to use a global-scope Identifier, although that is the recommended practice. Note that the syntax of the Node Identifier field has exactly the same syntax as that defined for IPv6 address in Section 2.5.1 of RFC 4291 [RFC4291]. (This is based on the IEEE EUI-64 syntax [IEEE-EUI], but is not the same.)

Most commonly, Identifiers have global-scope and are derived from one or more IEEE 802 or IEEE 1394 'MAC Addresses' (sic) already associated with the node, following the procedure already defined for IPv6 [RFC4291]. Global-scope identifiers have a high probability of being globally unique. This approach eliminates the need to manage Identifiers, among other benefits.

Local-scope Identifiers MUST be unique within the context of their Locators. The existing mechanisms of the IPv4 Address Resolution Protocol [RFC826] and IPv6 State-Less Address Auto-Configuration (SLAAC) [RFC4862] automatically enforce this constraint.

For example, on an Ethernet-based IPv4 subnetwork the ARP Reply message is sent via link-layer broadcast, thereby advertising the current binding between an IPv4 address and a MAC address to all nodes on that IPv4 subnetwork. (Note also that a well-known, long standing, issue with ARP is that it cannot be authenticated.) Local-scope Identifiers MUST NOT be used with other Locators without first ensuring uniqueness in the context of those other Locators e.g. by using IPv6 Neighbour Discovery's Duplicate Address Detection mechanism when using ILNPv6 or by sending an ARP Request when using ILNPv4.

Other methods might be used to generate local-scope Identifiers. For example, one might derive Identifiers using some form of cryptographic generation or using the methods specified in the IPv6 Privacy Extensions [RFC4941] to State-Less Address Auto-Configuration (SLAAC) [RFC4862]. When cryptographic generation of

Identifiers using methods described in RFC3972 is in use, only the Identifier is included, never the Locator, thereby preserving roaming capability. One could also imagine creating a local-scope Identifier by taking a cryptographic hash of a node's public key. Of course, in the unlikely event of a Identifier collision, for example when a node has chosen to use a local- scope Identifier value, the node remains free to use some other local-scope Identifier value(s).

It is worth remembering here that an IPv6 address names a specific network interface on a specific node, but an ILNPv6 Identifier names the node itself, not a specific interface on the node. This difference in definition is essential to providing seamless support for mobility and multi-homing, which are discussed in more detail later in this note.

3.2 Network-level packet formats

ILNPv6 Locator and Identifier values are encoded into IPv6 address space and ILNPv6 uses directly the Classic IPv6 packet format, as shown in Figure 3.3. This is also the view of an ILNPv6 packet as seen by core routers - they simply use the Locator value (top 64-bits of the address field) just as they would use an IPv6 prefix today (e.g. either as /48 or as /64 when using sub-network routing).

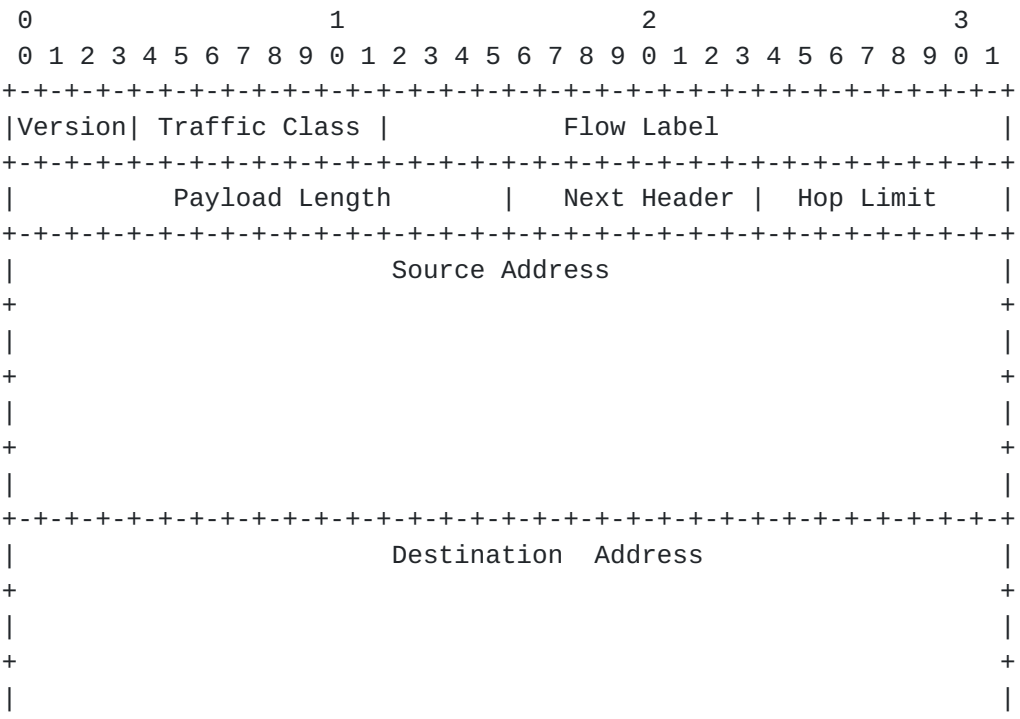


Figure 3.3: Existing ("Classic") IPv6 Header

In essence, the Locator names a subnetwork. (Locators can also be referred to as Routing Prefixes if discussing Classic IPv6). Of course, backwards compatibility requirements mean that ILNPv6 Locators use the same number space as IPv6 routing prefixes. This ensures that no changes are needed to deployed IPv6 routers when deploying ILNPv6.

The low-order 64-bits of the IPv6 address become the Identifier. Details of the Identifier were discussed above. The Identifier is only used by end-systems, so Figure 3.4 shows the view of the same packet format, but as viewed by an ILNPv6 node. As this only needs to be parsed in this way by the end-system, so ILNPv6 deployment is enabled incrementally by updating end-systems as required.

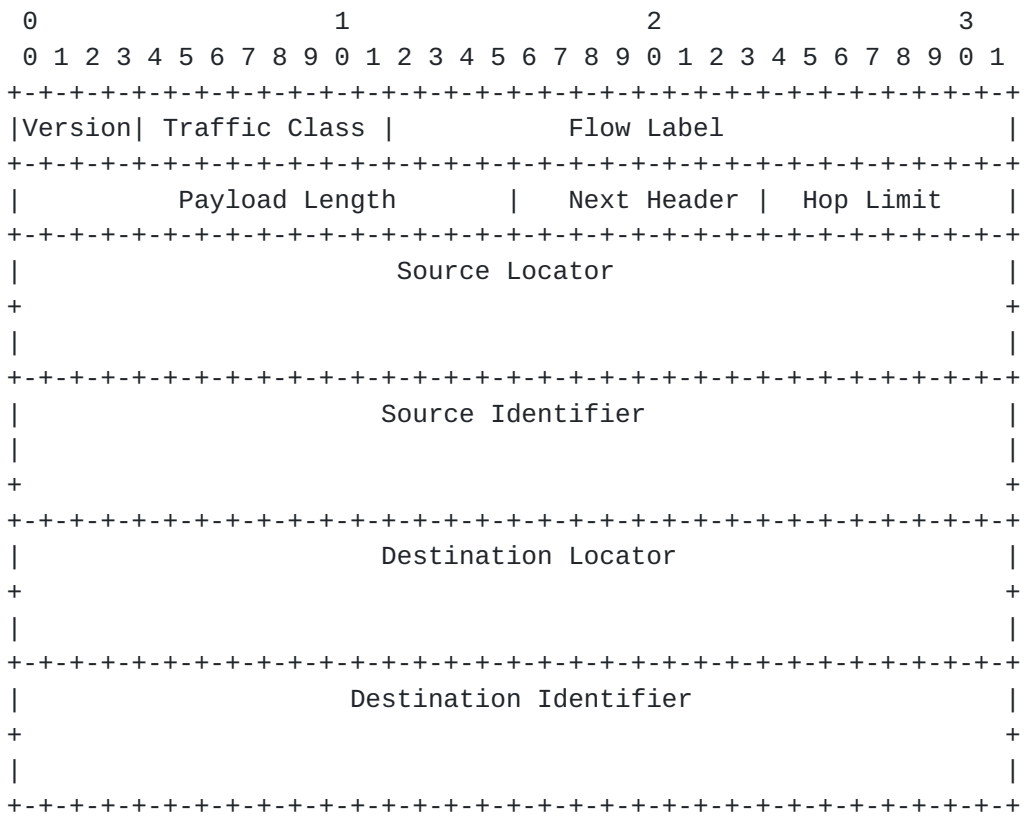


Figure 3.4: ILNPv6 Header as seen by ILNPv6-enabled end-systems

3.3 Encoding of identifiers and locators for ILNPv4

Encoding of Identifier and Locator values for ILNPv4 is not as straight-forward as for ILNPv6. In analogy to ILNPv6, in ILNPv4, the Locator value is a routing prefix for IPv4, but is at most 30 bits. Source Locator values are carried in the source address field of the IPv4 header, and destination Locator values in the destination address field. So, just like for ILNPv6, for ILNPv4, packet routing can be performed by routers examining existing prefix values in the IPv4 header.

However, for ILNPv4, additional option headers have to be used to carry the Identifier value as there is not enough room in the normal IPv4 header fields. A 64-bit Identifier value is carried in an option header. So, the detailed explanation of the ILNPv4 packet header is to be found in [ILNP-v4OPTS].

4. TRANSPORT-LAYER CHANGES

ILNP uses an Identifier value in order to form the invariant end-system state for end-to-end protocols. Currently, transport protocols such as TCP and UDP use all the bits of an IP address to form such state. So, transport protocol implementations MUST be modified in order to operate over ILNP.

4.1 End-system state

Currently, TCP and UDP, for example, use the 4-tuple:

<local port, remote port, local IP address, remote IP address>

for the end-system state for a transport layer end-point. For ILNP, implementations must be modified to instead use:

<local port, remote port, local Identifier, remote Identifier>

4.2 TCP/UDP Checksum Handling

In IP-based implementations, the TCP or UDP pseudo-header checksum calculations include all the bits of the IP address. By contrast, when calculating the TCP or UDP pseudo-header checksums for use with ILNP, only the Identifier values are included in the TCP or UDP pseudo-header checksum calculations.

To minimise the changes required within transport protocol implementations, and to maximise interoperability, current implementations are modified to zero the Locator fields (only for

the purpose of TCP or UDP checksum calculations). For example, for ILNPv6, this means that the existing code for IPv6 can be used, with the ILNPv6 Identifier bits occupying the lower 64 bits of the IPv6 address field, and the upper 64 bits of the IPv6 address filed being set to zero. For ILNPv4, the Identifier fields are carried in an IPv4 Option [ILNP-v4OPTS].

Section 7 describes methods for incremental deployment of this ILNP-specific change and backwards compatibility with non-upgraded nodes (e.g. classic IPv4 or IPv6 nodes) in more detail.

4.3 ICMP Checksum Handling

To maximise backwards compatibility, the ILNPv6 ICMP checksum is always calculated in the same way as for IPv6 ICMP. Similarly, the ILNPv4 ICMP checksum is always calculated in the same way as for IPv4 ICMP.

5. ILNP COMMUNICATION CACHE (ILCC)

For operational purposes, implementations need to have a local cache of state information that allow communication end-points to be constructed and for communication protocols to operate. Such cache information is common today, e.g. IPv4 nodes commonly maintain an Address Resolution Protocol (ARP) cache with information relating to current and recent Correspondent Nodes; IPv6 nodes maintain a Neighbor Discovery (ND) table with information relating to current and CNs. Likewise, ILNP maintains an Identifier-Locator Communication Cache (ILCC) with information relating to the operation of ILNP.

The ILCC is a (logical) set of data values required for ILNP to operate. These values are maintained by the endpoints of each ILNP session.

In theory, this cache is within the ILNP network-layer. However, many network protocol implementations do not have strict protocol separation or layering. So there is no requirement that the ILCC be kept partitioned from transport-layer protocols.

Note that in many implementations, much of the information required for the ILCC may already be present. Where some additional information is required for ILNP, from an engineering viewpoint, the ILCC could be implemented by extending or enhancing existing data structures within existing implementations. For example, by adding appropriate flags to the

data structures in existing implementations.

Note that the ILCC does not impose any extra state maintenance requirements for applications or applications servers. For example, in the case of, say, HTTP, there will be no additional state for a server to maintain, and any TCP state will be handled by the ILNP code in the OS just as for IP.

5.1 Formal Definition

The ILCC contains information about both the local node and also about current or recent correspondent nodes, as follows.

Information about the local node:

- Each currently valid Identifier value,
  including its Identifier Precedence
  and whether it is active at present.

- Each currently valid Locator value, including
  its associated local interface(s),
  its Locator Precedence, and
  whether it is active at present.

- Each currently valid IL Vector (I-LV), including
  whether it is active at present.

Information about each correspondent node:

- Most recent set of Identifiers,
  including lifetime and validity for each.

- Most recent set of Locators,
  including lifetime and validity for each.

- Nonce value for packets from the local host
  to the correspondent.

- Nonce value for packets from the correspondent
  to the local host.

In the above list for the ILNP Communication Cache:

- A "valid" item is usable, from an administrative point of view, but might or might not be in use at present.

- The "validity" parameter for the correspondent node indicates

one of several different states for a datum. These include at least the following:

- "valid" : data is usable and has not expired.

- "active" : data is usable, has not expired, and is in active use at present.

- "expired" : data is still in use at present, but is beyond its expiration (i.e. without a replacement value).

- "aged" : data was recently in use, but is not in active use at present, and is beyond its expiration.

- The "lifetime" parameter is an implementation-specific representation of the validity lifetime for the associated data element. In normal operation, the Lifetime for a correspondent node's Locator(s) are learned from the DNS Time-To-Live (DNS TTL) value associated with DNS records (NID, L32, L64 etc) of the FQDN owner name of the correspondent node. For time, a node might use UTC (e.g. via Network Time Protocol) or perhaps some node-specific time (e.g. seconds since node boot).

5.2 Ageing ILCC Entries

As a practical engineering matter, it is not sensible to flush all Locator values associated with an existing ILNP session's correspondent node even if the DNS TTL associated with those Locator values expires.

In some situations, a CN might be disconnected briefly when moving location (e.g. immediate handover, which sometimes is called "break before make"). If this happens, there might be a brief pause before the Correspondent Node can (a) update its own L values in the DNS, and (b) send an ICMP Locator Update message to the local node with information about its new location. Implementers ought to try to maintain ILNP sessions even when such events occur.

Instead, Locator values cached for a correspondent node SHOULD be marked as "aged" when their TTL has expired, but retained until either the next Locator Update message is received, there is other indication that a given Locator is not working any longer, there is positive indication that the Correspondent Node has terminated the ILNP session (e.g. TCP RST if the only

transport-layer session for this ILNP session is a TCP session), until some appropriate timeout (e.g. 2*MSL for TCP if the only transport-layer session for this ILNP session is a TCP session), or the ILNP session has been inactive for several minutes (e.g. no transport-layer session exists for this ILNP session) and the storage space associated with the aged entry needs to be reclaimed.

Separately, received authenticated Locator Update messages cause the ILCC entries listed above to be updated.

Similarly, if there is indication that an ILNP session with a Correspondent Node remains active and the DNS TTL associated with that Correspondent Node's active Identifier value(s) has expired, those remote Identifier value(s) ought to be marked as "expired", but retained since they are in active use.

5.3 Large Numbers of Locators

Implementers should keep in mind that a node or site might have a large number of concurrent Locators, and should ensure that a system fault does not arise if the system receives an authentic ICMP Locator Update containing a large number of Locator values.

5.4 Lookups into the ILCC

For received packets containing an ILNP Nonce Option, lookups in the ILCC MUST use the <remote Identifier, Nonce> tuple as the lookup key.

For all other ILNP packets, lookups in the ILNP Correspondent Cache MUST use the <remote Locator, remote Identifier> tuple, i.e. the remote I-LV, as the lookup key.

These two checks between them facilitate situations where, perhaps due to deployment of Local-scope Identifiers, more than one correspondent node is using the same Identifier value.

(NOTE: Other mechanisms, such as IPv6 Neighbor Discovery, ensure that 2 different nodes are incapable of using a given IL-V at the same location i.e., on the same link.)

While Locators are omitted from the transport-layer checksum, an implementation SHOULD use Locator values to distinguish between correspondents coincidentally using the same Identifier value (e.g. due to deployment of Local-scope Identifier values) when demultiplexing to determine which application(s) should receive the user data delivered by the transport-layer protocol.

6. HANDLING LOCATION/CONNECTIVITY CHANGES

In normal operation, an ILNP node uses the DNS for initial rendezvous in setting up ILNP sessions. The use of DNS for initial rendezvous with mobile nodes was earlier proposed by others [PHG02] and then separately re-invented by the current authors later on.

6.1 Node Location/Connectivity Changes

To handle the move of a node or a change to the upstream connectivity of a multi-homed node, we add a new ICMP control message [ILNP-ICMPv4] [ILNP-ICMPv6]. The ICMP Locator Update (LU) message is used by a node to inform its existing CNs that the set of valid Locators for the node has changed. This mechanism can be used to add newly valid Locators, to remove no longer valid Locators, or to do both at the same time. The LU mechanisms is analogous to the Binding Update mechanism in Mobile IPv6, but in ILNP, such messages are used any time Locator value changes need to be notified to CNs, e.g. for multi-homed hosts as well as for mobile hosts.

Further, if the node wishes to be able to receive new incoming ILNP sessions, the node normally uses Secure Dynamic DNS Update [RFC3007] to ensure that a correct set of Locator values are present in the appropriate DNS records (i.e. L32, L64) in the DNS for that node [ILNP-DNS]. This enables any new correspondents to correctly initiate a new ILNP session with the node at its new location.

While the Locator Update control message could be an entirely new protocol running over UDP, for example, there is no obvious advantage to creating a new protocol rather than using a new ICMP message. So ILNP defines a new ICMP Locator Update message for both IPv4 and IPv6.

6.2 Network Connectivity/Locator Changes

As a DNS performance optimisation, the LP DNS resource record MAY be used to avoid requiring each node on a subnetwork to update its DNS L64 record entries when that subnetwork's location (e.g. upstream connectivity) changes [ILNP-DNS]. This can reduce the number of DNS updates required when a subnetwork moves from Order(number of nodes on subnetwork) to Order(1).

In this case, the nodes on the subnetwork each would have an LP record pointing to a common Fully-Qualified Domain Name (FQDN) used to name that subnetwork. In turn, that subnetwork's domain

name would have one or more L64 record(s) in the DNS. Since the contents of an LP record are stable, relatively long DNS TTL values can be associated with these records facilitating DNS caching. By contrast, the DNS TTL of an L32 or L64 record for a mobile or multi-homed node should be small. Experimental work at the University of St Andrews indicates that the DNS continues to work well even with very low (e.g. zero) DNS TTL values [BA11].

Correspondents of a node on a mobile subnetwork using this DNS performance optimisation would initially perform a normal FQDN lookup for a node. If that lookup returned another FQDN in an LP record as additional data, then the correspondent would perform a lookup on that FQDN and expect an L32 or L64 record returned as additional data, in order to learn the Locator value to use to reach that target node. (Of course, a lookup that did not return any ILNP-related DNS records would result in an ordinary IPv4 session or ordinary IPv6 session being initiated, instead.)

7. SUBNETTING

For ILNPv4 and ILNPv6, the Locator value includes the subnetting information, as that also is topological information. As well as being architecturally correct, the placement of subnetting as part of the Locator is also convenient from an engineering point of view in both IPv4 and IPv6.

We consider that a Locator value, L consists of two parts:

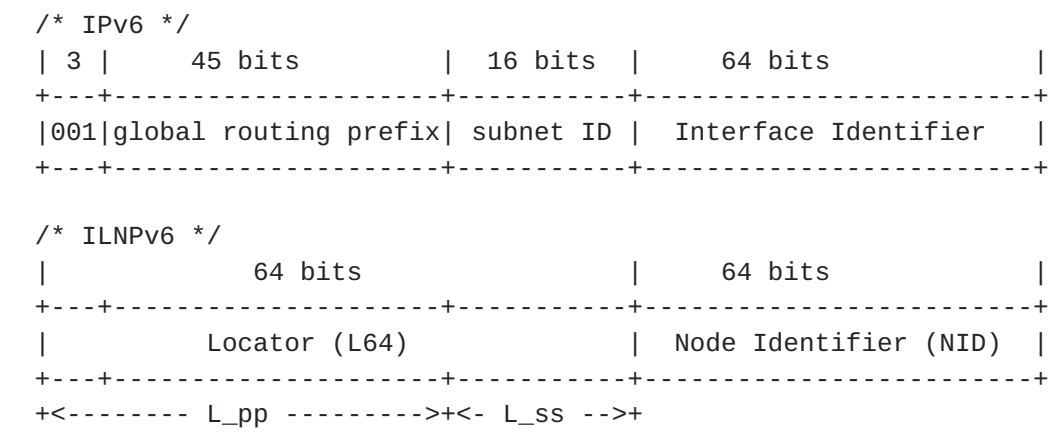
- L_pp: the Locator prefix part, which occupies the most significant bits in the address (for both ILNPv4 and ILNPv6).

- L_ss: Locator subnetwork selector, which occupies bits just after the L_pp.

For each of ILNPv4 and ILNPv6, L_pp gets its value from the provider-assigned routing prefix for IPv4 and IPv6, respectively. For L_ss, in each case of ILNPv4 and ILNPv6, the L_ss bits are located in the part of the address space which you might expect them to be located if IPv4 or IPv6 addresses were being used, respectively.

7.1 Subnetting for ILNPv6

For ILNPv6, recall that the Locator value is encoded to be syntactically similar to an IPv6 address prefix, as shown in Figure 7.1.



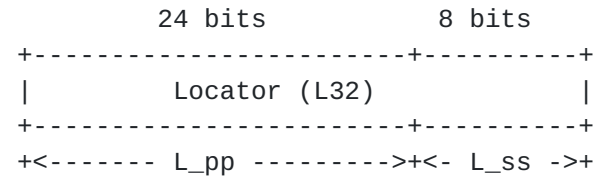
L_pp = Locator prefix part (assigned IPv6 prefix)
L_ss = Locator subnet selector (locally managed subnet ID)

Figure 7.1: IPv6 address format [RFC3587] as used in ILNPv6, showing how subnets can be identified.

Note that the subnet ID forms part of the Locator value. Note also that [RFC6177] allows the global routing prefix to be more than 45 bits, and for the subnet ID to be smaller, but still preserving the 64-bit size of the Locator.

7.2 Subnetting for ILNPv4

For ILNPv4, the L_pp value is an IPv4 routing prefix as used today, which is typically less than 32 bits. However, the ILNPv4 Locator value is carried in the 32-bit IP address space, so the bits not used for the routing prefix could be used for L_ss, e.g. for a /24 IPv4 prefix, the situation would be as shown in Fig 7.2.



L_pp = Locator prefix part (assigned IPv4 prefix)
L_ss = Locator subnet selector (locally managed subnet ID)

Figure 7.2: IPv4 address format for /24 IPv4 prefix, as used in ILNPv4, showing how subnets can be identified.

Note that the L_ss occupies bits that in an IPv4 address would normally be the host part of the address, which the site network could use for sub-netting in any case.

7.3 Subnetting for router-router links in IPv6/ILNPv6

There is a special case of /127 prefixes used in router-router, point-to-point links links for IPv6 [RFC6164]. ILNPv6 does not preclude such use.

8. DNS CONSIDERATIONS

ILNP makes use of DNS for name resolution, as does IP. Unlike IP, ILNP also uses DNS to support features such as mobility and multi-homing. While such usage is appropriate use of the DNS, it is important to discuss operational and engineering issues that may impact DNS usage.

8.1 Secure Dynamic DNS Update

When a host that expects incoming connections changes one or more of its Locator values, the host normally uses the IETF Secure Dynamic DNS Update protocol [RFC3007] to update the set of currently valid Locator values associated with its Fully Qualified Domain Name (FQDN). This ensures that the authoritative DNS server for its FQDN will be able to generate an accurate set of Locator values if the DNS server receives DNS name resolution request for its FQDN.

Liu & Albitz [LA06] report that Secure Dynamic DNS Update has been supported on the client-side for several years now in widely deployed operating systems (e.g. MS Windows, Apple MacOS X, UNIX, and Linux) and also in DNS server software (e.g. BIND). Publicly available product data sheets indicate that some other DNS server software packages, such as that from Nominum, also support this capability.

For example, Microsoft Windows XP (and later versions), the freely distributable BIND DNS software package (used in Apple MacOS X and in most UNIX systems), and the commercial Nominum DNS server all implement support for Secure Dynamic DNS Update and are known to interoperate [LA06]. There are credible reports that when a site deploys Microsoft's Active Directory, the site (silently) automatically deploys Secure Dynamic DNS Update [LA06]. So, many sites have already deployed Secure Dynamic DNS Update even though they are not actively using it (and might not be aware they have already deployed that protocol) [LA06].

So DNS update via Secure Dynamic DNS Update is not only standards-based, but also readily available in widely deployed systems today.

8.3 New DNS RR types

As part of this proposal, additional DNS Resource Records have been proposed in a separate document [ILNP-DNS]. These new records are summarised in Table 6.1.

new DNS RR type	Purpose
NID	store the value of a Node Identifier
L32	store the value of a 32-bit Locator for ILNPv4
L64	store the value of a 64-bit Locator for ILNPv6
LP	points to a (several) L32 and/or L64 record(s)

Table 6.1: Summary of new DNS RR types for ILNP

With this proposal, mobile or multi-homed nodes and sites are expected to use the existing "Secure Dynamic DNS Update" protocol to keep their Node Identifier (NID) and Locator (L32 and/or L43) records correct in their authoritative DNS server(s) [RFC3007] [ILNP-DNS].

Reverse DNS lookups, to find a node's Fully Qualified Domain Name from the combination of a Locator and related Identifier value, can be performed as at present.

8.4 DNS TTL values for ILNP RRS types

Existing DNS specifications require that DNS clients and DNS resolvers honour the TTL values provided by the DNS servers. In the context of this proposal, short DNS TTL values are assigned to particular DNS records to ensure that the ubiquitous DNS caching resolvers do not cache volatile values (e.g. Locator records of a mobile node) and consequently return stale information to new requestors.

The time-to-live (TTL) values for L32 and L64 records may have to be relatively low (perhaps a few seconds) in order to support mobility and multi-homing. Low TTL values may be of concern to administrators who might think that this would reduce efficacy of DNS caching increase DNS load significantly.

Previous research by others indicates that DNS caching is largely ineffective, with the exception of NS records and the addresses of DNS servers referred to by NS records [SBK02]. This means DNS caching performance and DNS load will not be adversely affected by assigning very short TTL values (down to zero) to the Locator records of typical nodes for a edge site [BA11]. It

also means that it is preferable to deploy the DNS server function on nodes that have longer DNS TTL values, rather than on nodes that have shorter DNS TTL values.

LP records normally are stable and will have relatively long TTL values, even if the L32 or L64 records they point to have values that have relatively low TTL values.

Identifier values might be very long-lived (e.g. days) when they have been generated from an IEEE MAC address on the system. Identifier values might have a shorter lifetime (e.g. hours or minutes) if they have been cryptographically-generated [RFC3972], or have been created by the IPv6 Privacy Extensions [RFC4941], or otherwise have the EUI-64 scope bit set to "local-scope". Note that when ILNP is used, the cryptographic generation method described in RFC 3972 is used only for the Identifier, omitting the Locator, thereby preserving roaming capability. Note that a given ILNP session normally will use a single Identifier value for the lifetime of that ILNP session.

8.5 IP/ILNP dual operation and transition

During a long transition period, a node that is ILNP-capable SHOULD have not only have NID and L32/L64 (or NID and LP) records present in its authoritative DNS server, but also SHOULD have A/AAAA records in the DNS for the benefit of non-upgraded nodes. Then, when any CN performs an FQDN lookup for that node, it will receive the A/AAAA with the appropriate NID, L32/L64 and/or LP records as "additional data".

Existing DNS specifications require that a DNS resolver or DNS client ignore unrecognised DNS record types. So gratuitously appending NID and Locator (i.e., L32, L64, or LP) records as "additional data" in DNS responses to A/AAAA queries ought not to create any operational issues. So, IP only nodes would use the A/AAAA RRs, but ILNP-capable nodes would be able to use the NID, L32/L64 and/or LP records are required.

There is nothing to prevent this capability being implemented strictly inside a DNS server, whereby the DNS server synthesises a set of A/AAAA records to advertise from the NID and Locator (i.e., L32, L64, or LP) values that the node has kept updated in that DNS server. Indeed, such a capability may be desirable, reducing the amount of manual configuration required for a site, and reducing the potential for errors as the A/AAAA records would be automatically generated.

9. IP Security for ILNP

The primary conceptual difference from ordinary IP Security (IPsec) is that ILNP IP Security omits all use of, and all reference to, Locator values. This leads to several small, but important, changes to IP Security when it is used with ILNP sessions.

9.1 IPsec Security Associations enhancements for ILNP

IPsec Security Associations for ILNP only include the Identifier values for the endpoints, and omit the Locator values. As an implementation detail, ILNP implementations MUST be able to distinguish between different Security Associations with ILNP correspondents (at different locations, with different ILNP Nonce values in use) that happen to use the same Identifier values (e.g. due to an inadvertent Identifier collision when using identifier values generated by using the IPv6 Privacy Addressing extension). One possible way to distinguish between such different ILNP sessions is to maintain a mapping between the IPsec Security Association Database (SAD) entry and the corresponding ILCC entry.

Consistent with this enhancement to the definition of an IPsec Security Association, when processing received IPsec packets associated with an ILNP session, ILNP implementations ignore the Locator bits of the received packet and only consider the Identifier bits. This means, for example, that if an ILNP correspondent node moves to a different subnetwork, and thus is using a different Source Locator in the header of its ILNP IPsec packets, the ILNP session will continue to work and will continue to be secure.

Since implementations of ILNP are also required to support IP, implementers need to ensure that ILNP IPsec Security Associations can be distinguished from ordinary IPsec Security Associations. The details of this are left to the implementer. As an example, one possible implementation strategy would be to retain a single IPsec Security Association Database (SAD), but add an internal flag bit to each entry of that IPsec Security Association Database (SAD) to indicate whether ILNP is in use for that particular IPsec Security Association.

9.2 IP Authentication Header enhancements for ILNP

Similarly, for an ILNP session using IPsec, the IPsec Authentication Header (AH) only includes the Identifier values for the endpoints in its authentication calculations, and omits the Source Locator and Destination Locator fields from its authentication calculations. This enables IPsec AH to work well

even when used with ILNP localised numbering [ILNP-ADV] or other situations where a Locator value might change while the packet travels from origin to destination.

9.3 Key Management Considerations

In order to distinguish at the network-layer between multiple ILNP nodes that happen to be using the same Node Identifier values (e.g. because the identifier values were generated using the IPv6 Privacy Addressing method), key management packets being used to setup an ILNP IPsec session MUST include the ILNP Nonce option.

Similarly, key management protocols used with IPsec are enhanced to deprecate use of IP addresses as identifiers and to substitute the use of the new Node Identifier values for that purpose. This results in an ILNP IPsec Security Association that is independent of the Locator values that might be used.

For ILNPv6 implementations, the ILNP Node Identifier (64-bits) is smaller than the IPv6 Address (128-bits). So support for ILNPv6 IPsec is accomplished by zeroing the upper-64 bits of the IPv6 Address fields in the application-layer key management protocol, while retaining the Node Identifier value in the lower-64 bits of the application-layer key management protocol.

For ILNPv4 implementations, enhancements to the key management protocol likely will be needed, because existing key management protocols rely on 32-bit IPv4 addresses, while ILNP Node Identifiers are 64-bits. Such enhancements are beyond the scope of this specification.

10. BACKWARDS COMPATIBILITY & INCREMENTAL DEPLOYMENT

Experience with IPv6 deployment over the past many years has shown that it is important for any new network protocol to provide backwards compatibility with the deployed IP base and should be incrementally deployable, ideally requiring modification of only those nodes that wish to use ILNP and not requiring the modification of nodes that do not intend to use ILNP. The two instances of ILNP, ILNPv4 and ILNPv6, are intended to be, respectively, backwards compatible with, and incrementally deployable on, the existing IPv4 and IPv6 installed bases. Indeed, ILNPv4 and ILNPv6 can each be seen, from an engineering viewpoint, as supersets of the IPv4 and IPv6, respectively.

However, in some cases, ILNP introduces functions that supersede

equivalent functions available in IP. For example, ILNP has a mobility model, and so does not need to use the models for Mobile IPv4 or Mobile IPv6.

As ILNP changes the use of end-to-end namespaces, for the most part, it is only end-systems that need to be modified. However, in order to leverage existing engineering (e.g. existing protocols), in some cases, there is a compromise, and these are highlighted in this section.

10.1 Priorities in the design of ILNPv6 and ILNPv4

In the engineering design of ILNPv6 and ILNPv4, we have used the following priorities. In some ways, this choice is arbitrary, and it may be equally valid to "invert" these priorities for a different architectural and engineering design.

1. Infrastructure

   As much of the deployed IP network infrastructure should be used without change. That is, routers and switches should require minimal or zero modifications in order to run ILNP. As much as possible of the existing installed base of core protocols should be re-used.

2. Core protocols

   As much of the deployed network control protocols, such as routing, should be used without change. That is, existing routing protocols and switch configuration should require minimal or zero modifications in order to run ILNP.

3. Scope of end-system changes

   Any nodes that do not need to run ILNP should not need to be upgraded. It should be possible to have a site network that has a mix of IP-only and ILNP-capable nodes without any changes required to the IP-only nodes.

4. Applications

   There should be minimal impact on applications, even though ILNP requires end-to-end protocols to be upgraded. Indeed, for those applications that are "well-behaved" (e.g. do not use IP address values directly for application state or application configuration), there should be little or no effort required in enabling them to operate over ILNP.

Each of these items is discussed in its own section below.

10.2 Infrastructure

ILNP is designed to be deployed on existing infrastructure. No new infrastructure is required to run ILNP as it will be implemented as a software upgrade impacting only end-to-end protocols. Existing routing protocols can be re-used: no new routing protocols are required. This means that network operators and service providers do not need to learn about, test, and deploy new protocols, or change the structure of their network in order for ILNP to be deployed. Exceptionally, edge routers supporting ILNPv4 hosts will need to support an enhanced version of ARP.

10.3 Core protocols

Existing routing and other control protocols should not need to change in devices such as switches and routers. We believe this to be true for ILNPv6. However, for ILNPv4, we believe that ARP will need to be enhanced in edge routers (or layer-3 switches) that support ILNPv4 hosts. Backbone and transit routers still ought not require changes for either ILNPv4 or ILNPv6.

For both ILNPv4 and ILNPv6, the basic packet format for packets re-uses that format that is seen by routers for IPv4 and IPv6 respectively. Specifically, as the ILNP Locator value is always a routing prefix (either IPv4 or IPv6), routing protocols should work unchanged.

Both ILNPv4 and ILNPv6 introduce new header options (e.g Nonce Option messages) and ICMP messages (e.g. Locator Update messages) which are used to enable end-to-end signalling. For packet forwarding, depending on the forwarding policies used by some providers or site border routers, there may need to be modifications to those policies to allow the new header options and new ICMP messages to be forwarded. However, as the header options and new ICMP messages are end-to-end, such modifications are likely to be in configuration files (or firewall policy on edge routers), as core routers do NOT need to parse and act upon the information contained in the header options or ICMP messages.

10.4 Scope of end-system changes

Only end-systems that need to use ILNP need to be updated in order for ILNP to be used at a site.

There are three exceptions to this statement as follows:

a) ILNPv4 ARP: as the Identifier value for IPv4 cannot fit into the normal 20-byte IPv4 packet header (a header extension is used), ARP must be modified. This only impacts end-systems that use ILNPv4 and those switches or site-border routers that are the first hop from an ILNPv4 node. For ILNPv6, as the I and L values fit into the existing basic IPv6 packet, IPv6 Neighbour Discovery can operate without modification

b) Use of IP NAT: Where IP NAT or NAPT is in use for a site, existing NAT/NAPT device will re-write address fields in ILNPv4 packets or ILNPv6 packets. To avoid this, the NAT should either (i) be configured to allow the pass-through of packets originating from ILNP-capable nodes (e.g. by filtering on source address fields in the IP header); or (ii) should be enhanced to recognise ILNPv4 or ILNPv6 packets (e.g. by looking for the ILNP Nonce option).

c) Site border routers (SBRs) in ILNP Advanced Deployment scenarios: There are options to use an ILNP-capable site border router (SBR) as described in another document [ILNP-ADV]. In such scenarios, the SBR(s) need to be ILNP-capable.

Other than these exceptions, it is entirely possible to have a site that uses a mix of IP and ILNP nodes and requires no changes to nodes other than the nodes that wish to use ILNP. For example, if a user on a site wishes to have his laptop use ILNPv6, only that laptop would need to have an upgraded stack: no other devices (end-systems, layer-2 switches or routers) at that site would need to be upgraded.

10.5 Applications

As noted, in the Architecture Description [ILNP-ARCH], those applications that do not use IP address values in application state or configuration data are considered to be "well-behaved". Applications that work today through a NAT or NAPT device without application-specific support are also considered "well behaved". Such applications might use DNS FQDNs or application-specific name spaces. (Note Well: application-specific name spaces should not be derived from IP address values).

For well-behaved applications, replacing IP with ILNP should have no impact. That is, well-behaved applications should work unmodified over ILNP.

Those applications that use directly IP address values in application state or configuration will need to be modified for operation over ILNP. Examples of such applications include:

- FTP: which uses IP address values in the application layer protocol. In practice, use of Secure Copy (SCP) is growing, while use of FTP is either flat or declining, in part due to the improved security provided by SCP.

- SNMP: which uses IP address values in MIB definitions, and values derived from IP address values in SNMP object names.

Further experimentation in this area is planned to validate these details.

10.6 Interworking between IP and ILNP

A related topic is interworking: for example, how would an IPv6 node communicate with an ILNPv6 node? Currently, we make the assumption that ILNP nodes "drop down" to using IP when communicating with a non-ILNP capable node, i.e. there is no interworking as such. In the future, it may be beneficial to define interworking scenarios that do not rely on having ILNP nodes fall back to IP, for example by the use of suitable protocol translation gateways or middleboxes.

For now, a simplified summary of the process for interaction between ILNP hosts and non-ILNP hosts is as follows:

a) For a host initiating communication using DNS, the resolution of the FQDN for the remote host will return at least one NID record and at least one of an L32 record (for ILNPv4) or an L64 record (for ILNPv6). Then the host knows that the remote host supports ILNP.

b) When a host has I and L values for a remote host, the initial packet to initiate communication MUST contain a Nonce Header [ILNP-v4OPTS] [ILNP-NONCEv6] which indicates to the remote host that this packet is attempting to set-up an ILNP session.

c) When a receiving host sees a Nonce Header, if it DOES support ILNP it will proceed to set-up an ILNP session.

d) When a receiving host sees a Nonce Header, if it DOES NOT support ILNP it will reject the packet and this will be indicated to the sender through an ICMP message [ILNP-ICMPv6] [ILNP-ICMPv4]. Upon receiving the ICMP messages, the sender will re-initiate communication using standard IPv4 or IPv6.

Many observers in the community expect IPv4 to remain in place for a long time even though IPv6 has been available for over a decade. With a similar anticipation, it is likely that in the future there will be a mixed environment of both IP and ILNP hosts. Until there is a better understanding of the deployment and usage scenarios that will develop, it is not clear what interworking scenarios would be useful to define and focus on between IP and ILNP.

11. SECURITY CONSIDERATIONS

There are numerous security considerations for ILNP from an engineering viewpoint. Overall, ILNP and its capabilities are no less secure than IP and equivalent IP capabilities. In some cases, ILNP has the potential to be more secure, or offer security capability in a more harmonised manner, for example with ILNP's use of IPsec in conjunction with multi-homing and mobility. [ILNP-ARCH] describes several security considerations that apply to ILNP and is included here by reference.

ILNP offers an enhanced version of IP Security (IPsec). The details of IP Security for ILNP were described separately above. All ILNP implementations MUST support the use of the IP Authentication Header (AH) for ILNP and also the IP Encapsulating Security Payload (ESP) for ILNP, but deployment and use of IPsec for ILNP remains a matter for local operational security policy.

11.1 Authenticating ICMP Messages

Separate documents propose a new IPv4 Option [ILNP-v4OPTS] and a new IPv6 Destination Option [ILNP-NONCEv6]. Each of these options can be used to carry a ILNP Nonce value end-to-end between communicating nodes. That nonce provides protection against off-path attacks on an ILNP session. These ILNP Nonce options are used ONLY for ILNP and not for IP. The nonce values are exchanged in the initial packets of an ILNP session by including them in those initial/handshake packets.

ALL ICMP Locator Update messages MUST include an ILNP Nonce option and also MUST include the correct ILNP Nonce value for the claimed sender and intended recipient of that ICMP Locator Update message. There are no exceptions to this rule. ICMP Locator Update messages MAY be protected by IP Security (IPsec), but still MUST include an ILNP Nonce option and the ILNP Nonce option still MUST include the correct ILNP Nonce value.

When a node has an active ILNP session, and that node changes its

Locator set, it SHOULD include the apropriate ILNP Nonce Option in the first few data packets sent using a new Locator value, so that the recipient can validate the received data packets as valid (despite having an unexpected Source Locator value).

Any ILNP Locator Update messages received without an ILNP Nonce option MUST be discarded as forgeries.

Any ILNP Locator Update messages received with an ILNP Nonce option, but do NOT have the correct ILNP Nonce value inside the ILNP Nonce option, MUST be discarded as forgeries.

When the claimed sender of an ICMP message is known to be a current ILNP correspondent of the recipient (e.g. has a valid, non-expired, ILCC entry), then any ICMP error messages from that claimed sender MUST include the ILNP Nonce option and MUST include the correct ILNP Nonce value (i.e. correct for that sender recipient pair) in that ILNP Nonce option.

When the claimed sender of an ICMP error message is known to be a current ILNP correspondent of the recipient (e.g. has a valid, non-expired, ILCC entry), then any ICMP error messages from that claimed sender that are received without an ILNP Nonce option MUST be discarded as forgeries.

When the claimed sender of an ICMP error message is known to be a current ILNP correspondent of the recipient (e.g. has a valid, non-expired, ILCC entry), then any ICMP error messages from that claimed sender that contain an ILNP Nonce option, but do NOT have the correct ILNP Nonce value inside the ILNP Nonce option, MUST be discarded as forgeries.

ICMP messages (not including ICMP Locator Update messages) with a claimed sender that is NOT known to be a current ILNP correspondent of the recipient (e.g. does not have a valid, non-expired, ILCC entry) MAY include the ILNP Nonce option, but in this case the ILNP Nonce option is ignored by the recipient upon receipt, since the recipient has no way to authenticated the received ILNP Nonce value.

Received ICMP messages (not including ICMP Locator Update messages) with a claimed sender that is NOT known to be a current ILNP correspondent of the recipient (e.g. does not have a valid, non-expired, ILCC entry) do NOT require the ILNP Nonce option, because the security risks are no different than for deployed IPv4 and IPv6 -- provided that the received ICMP message is not an ICMP Locator Update message. Such ICMP messages (e.g. Destination Unreachable, Packet Too Big) might legitimately

originate in an intermediate system along the path of an ILNP session. That intermediate system might not be ILNP capable. Even if ILNP capable itself, that intermediate system might not know which packets it forwards are part of ILNP sessions.

When ILNP is in use, IP Security for ILNP also MAY be used to protect stronger protections for ICMP packets associated with an ILNP session. Even in this case, the ILNP Nonce option also MUST be present and MUST contain the correct ILNP Nonce value. This simplifies packet processing, and also enables rapid discard of any forged packets from an off-path attacker that lack either the ILNP Nonce option or the correct ILNP Nonce value -- without requiring computationally-expensive IPsec processing. Received ICMP messages that are protected by ILNP IP Security, but fail the recipient's IP Security checks, MUST be dropped as forgeries. If a deployment chooses to use ILNP IPsec ESP to protect its ICMP messages and is NOT also using ILNP IPsec AH with those messages, then the ILNP Nonce option MUST be placed in the ILNP packet after the ILNP IPsec ESP header, rather than before the ILNP IPsec ESP header, to ensure that the Nonce option is protected in transit.

Receipt of any ICMP message that is dropped or discarded as a forgery SHOULD cause the details of the received forged ICMP packet (e.g. Source and Destination Locators / Source and Destination Identifiers / Source and Destination IP addresses, ICMP message type, receiving interface, receive date, receive time) to be logged in the receiving system's security logs. Implementations MAY rate-limit such logging in order to reduce operational risk of denial-of-service attacks on the system logging functions. The details of system logging are implementation-specific.

11.2 Forged Identifier Attacks

The ILNP Communication Cache (ILCC) contains two unidirectional nonce values (one used in control messages sent by this node, a different one used to authenticate messages from the other node) for each active or recent ILNP session. The ILCC also contains the currently valid set of Locators and set of Identifiers for each correspondent node.

If a received ILNP packet contains valid Identifier values and a valid Destination Locator, but contains a Source Locator value that is not present in the ILCC, the packet MUST be dropped as an invalid packet and a security event SHOULD be logged, UNLESS the packet also contains a Nonce Destination Option with the correct

value used for packets from the node with that Source Identifier to this node. This prevents an off-path attacker from stealing an existing ILNP session.

12. PRIVACY CONSIDERATIONS

There are no additional privacy issues created by ILNP compared to IP. Please see Section 10 of [ILNP-ARCH] for more detailed discussion of Privacy Considerations.

ILNPv6 supports use of the IPv6 Privacy Extensions for Stateless Address Auto-configuration in IPv6 [RFC4941] to enable identity privacy (see also Section 2).

Location Privacy can be provided by locator re-writing techniques as described in Section 7 of [ILNP-ADV].

A description of various possibilities for obtaining both identity privacy and location privacy with ILNP can be found in [BAK11].

13. OPERATIONAL CONSIDERATIONS

This section covers various operational considerations relating to ILNP, including potential session liveness and reachability considerations and Key Management considerations. Again, the situation is similar to IP, but it is useful to explain the issues in relation to ILNP nevertheless.

13.1 Session Liveness and Reachability

For bi-directional flows, such as a TCP/ILNP session, each node knows whether the current path in use is working by the reception of data packets, acknowledgements, or both. Therefore, as with TCP/IP, TCP/ILNP does not need special path probes. UDP/ILNP sessions with acknowledgements work similarly, and also do not need special path probes.

In the deployed Internet, the sending node for a UDP/IP session without acknowledgements does not know for certain that all packets are received by the intended receiving node. Such UDP/ILNP sessions have the same properties as UDP/IP sessions in this respect. The receiver(s) of such an UDP/ILNP session SHOULD send a gratuitous IP packet containing an ILNP Nonce option to the sender, in order to enable the receiver to subsequently send ICMP Locator Updates if appropriate [ILNP-NONCEv6]. In this case,

UDP/ILNP sessions fare better than UDP/IP sessions, still without using network path probes.

A mobile (or multi-homed) node may change its connectivity more quickly than DNS can be updated. This situation is unlikely, particularly given the widespread use of link-layer mobility mechanisms (e.g. GSM, IEEE 802 bridging) in combination with network-layer mobility. However, the situation is equivalent to the situation where a traditional IP node is moving faster than the Mobile IPv4 or Mobile IPv6 agents/servers can be updated with the mobile node's new location. So the issue is not new in any way to ILNP. In all cases, Mobile IPv4 and Mobile IPv6 and ILNP, a node moving that quickly might be temporarily unreachable until it remains at a given network-layer location (e.g. IP subnetwork, ILNP Locator value) long enough for the location update mechanisms (for Mobile IPv4, for Mobile IPv6, or ILNP) to catch up.

Another potential issue for IP is what is sometimes called "Path Liveness" or, in the case of ILNP, "Locator Liveness". This refers to the question of whether an IP packet with a particular destination Locator value will be able to reach the intended destination network or not, given that some otherwise valid paths might be unusable by the sending node (e.g. due to security policy or other administrative choice). In fact, this issue has existed in the IPv4 Internet for decades.

For example, an IPv4 server might have multiple valid IP addresses, each advertised to the world via an DNS A record. However, at a given moment in time, it is possible that a given sending node might not be able to use a given (otherwise valid) destination IPv4 address in an IP packet to reach that IPv4 server.

Indeed, for ILNPv6, as the ILNP packet reuses the IPv6 packet header and uses IPv6 routing prefixes as Locator values, such liveness considerations are no worse than they are for IPv6 today. For example, for IPv6, if a host, H, performs a DNS lookup for an FQDN for remote host F, and receives a AAAA RR with IPv6 address F_A, this does not mean necessarily that H can reach F on its F_A using its current connectivity, i.e. an IPv6 path may not be available from H to F at that point in time.

So we see that using an Identifier/Locator Split architecture does not create this issue, nor does it make this issue worse than it is with the deployed IPv4 Internet.

In ILNP, the same conceptual approach described in [RFC5534]

(Locator Pair Exploration for SHIM6) can be reused. Alternatively, an ILNP node can reuse the existing IPv4 methods for determining whether a given path to the target destination is currently usable, for which existing methods leverage transport-layer session state information that the communicating end systems are already keeping for transport-layer protocol reasons.

Lastly, it is important to note that the ICMP Locator Update mechanism described in [ILNP-ICMPv6] [ILNP-ICMPv4] is a performance optimisation, significantly shortening the network-layer handoff time if/when a correspondent changes location. Architecturally, using ICMP is no different from using UDP, of course.

13.2 Key Management Considerations

ILNP potentially has advantages over either form of Mobile IP with respect to key management, given that ILNP is using Secure Dynamic DNS Update -- which capability is much more widely available today in deployed desktop and server environments (e.g. Microsoft Windows, MacOS X, Linux, other UNIX), as well as being widely available today in deployed DNS server software (e.g. Microsoft and the freely available BIND) and appliances [LA06], than the Security enhancements needed by either Mobile IPv4 or Mobile IPv6.

IETF work in progress is addressing use of DNS to support key management for entities having DNS Fully-Qualified Domain Names.

13.3 Point-to-Point Router Links

As a special case, for the operational reasons described in [RFC6164], ILNPv6 deployments MAY continue to use classic IPv6 with a /127 routing prefix on router to router point-to-point links (e.g. SONET/SDH). Because an ILNPv6 packet and an IPv6 packet are indistinguishable for forwarding purposes to a transit router, this should not create any operational difficulty for ILNPv6 traffic travelling over such links.

14. REFERRALS & APPLICATION PROGRAMMING INTERFACES

This section is concerned with support for using existing ("legacy") applications over ILNP, including both referrals and Application Programming Interfaces (APIs).

ILNP does NOT require well-behaved applications be modified to use a new networking API, nor does it require applications be

modified to use extensions to an existing API. Existing well-behaved IP applications should work over ILNP without modification using existing networking APIs.

14.1 BSD Sockets APIs

The existing BSD Sockets API can continue to be used with ILNP underneath the API. That API can be implemented in a manner that hides the underlying protocol changes from the applications. For example, the combination of a Locator and an Identifier can be used with the API in the place of an IPv6 address.

So it is believed that existing IP address referrals can continue to work properly in most cases. For a rapidly moving target node, referrals might break in at least some cases. The potential for referral breakage is necessarily dependent upon the specific application and implementation being considered.

It is suggested, however, that a new, optional, more abstract, C language API be created so that new applications may avoid delving into low-level details of the underlying network protocols. Such an API would be useful today, even with the existing IPv4 and IPv6 Internet, whether or not ILNP were ever widely deployed.

14.2 Java (and other) APIs

Most existing Java APIs already use abstracted network programming interfaces, for example in the java.Net.URL class. Because these APIs already hide the low-level network-protocol details from the applications, the applications using these APIs (and the APIs themselves) don't need any modification to work equally well with IPv4, IPv6, ILNP, and probably also HIP.

Other programming languages, such as C++, python and ruby, also provide higher-level APIs that abstract away from sockets, even though sockets may be used beneath those APIs.

14.3 Referrals in the Future

The approach proposed in [ID-Referral] appears to be very suitable for use with ILNP, in addition to being suitable for use with the deployed Internet. Protocols using that approach would not need modification to have their referrals work well with IPv4, IPv6, ILNP, and probably also other network protocols

(e.g. HIP).

A sensible approach to referrals is to use Fully-Qualified Domain Names (FQDNs), as is commonly done today with web URLs. This approach is highly portable across different network protocols, even with both the IPv4 Internet or the IPv6 Internet.

15. IANA CONSIDERATIONS

There are no IANA considerations.

(The RFC Editor is requested to remove this section prior to publication.)

16. REFERENCES

16.1 Normative References

[IEEE-EUI] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", http://standards.ieee.org/regauth/oui/tutorials/EUI64.html, IEEE, Piscataway, NJ, USA, March 1997.

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

[RFC3007] B. Wellington, "Secure Domain Name System Dynamic Update", RFC3007, November 2000.

[RFC3177] IAB and IESG, "IAB/IESG Recommendations on IPv6 Address Allocations to Sites", RFC3177, September 2001.

[RFC4219] R. Hinden & S. Deering, "IP Version 6 Addressing Architecture", RFC4219, February 2006.

[RFC4862] S. Thomson, T. Narten & T. Jimnei, "IPv6 Stateless Address Autoconfiguration", RFC4862, Sep 2007

[RFC6177] T. Narten, G. Huston, & L. Roberts, "IPv6 Address Assignment to End Sites", RFC6177, March 2011.

[ILNP-ARCH] R.J. Atkinson & S.N. Bhatti, "ILNP Architectural Description", draft-irtf-rrg-ilnp-arch, 10 July 2012.

[ILNP-ARP] R.J. Atkinson & S.N. Bhatti, "ARP Extension for ILNPv4", draft-irtf-rrg-ilnp-arp, 10 July 2012.

[ILNP-DNS] R.J. Atkinson, S.N. Bhatti, & S Rose, "DNS Resource Records for ILNP", draft-irtf-rrg-ilnp-dns, 10 July 2012.

[ILNP-ICMPv4] R.J. Atkinson & S.N. Bhatti, "ICMPv4 Locator Update message" draft-irtf-rrg-ilnp-icmpv4, 10 July 2012.

[ILNP-ICMPv6] R.J. Atkinson & S.N. Bhatti, "ICMPv6 Locator Update message" draft-irtf-rrg-ilnp-icmpv6, 10 July 2012.

[ILNP-NONCEv6] R.J. Atkinson & S.N. Bhatti, "IPv6 Nonce Destination Option for ILNPv6", draft-irtf-rrg-ilnp-noncev6, 10 July 2012.

[ILNP-v4OPTS] R.J. Atkinson & S.N. Bhatti, "IPv4 Options for ILNP", draft-irtf-rrg-ilnp-v4opts, 10 July 2012.

16.2 Informative References

[BA11] S. Bhatti & R. Atkinson, "Reducing DNS Caching", Proceedings of IEEE Global Internet Symposium (GI2011), Shanghai, P.R. China. 15 April 2011.

[BAK11] S.N. Bhatti, R. Atkinson, J. Klemets, "Integrating Challenged Networks", Proceedings of IEEE Military Communications Conference (MILCOM), IEEE, Baltimore, MD, USA. Nov 2011.

[LA06] Cricket Liu and Paul Albitz, "DNS and Bind", 5th Edition, O'Reilly & Associates, Sebastopol, CA, USA. 2006. ISBN 0-596-10057-4.

[PHG02] A. Pappas, S. Hailes, & R. Giaffreda, "Mobile Host Location Tracking through DNS", Proceedings of IEEE London Communications Symposium, IEEE, September 2002, London, England, UK.

[SBK02] Alex C. Snoeren, Hari Balakrishnan, & M. Frans

Kaashoek, "Reconsidering Internet Mobility", Proceedings of 8th Workshop on Hot Topics in Operating Systems, IEEE, Elmau, Germany, May 2001.

[ID-Referral] B. Carpenter and others, "A Generic Referral Object for Internet Entities", draft-carpenter-behave-referral-object-01, 20 October 2009.

[RFC3972] T. Aura, "Cryptographically Generated Addresses (CGAs)", RFC3972, March 2005.

[RFC4291] R. Hinden & S. Deering, "IP version 6 Addressing Architecture", RFC4291, February 2006.

[RFC4581] M. Bagnulo & J. Arkko, "Cryptographically Generated Addresses Extension Field Format", RFC4581, October 2006.

[RFC4941] T. Narten, R. Draves, & S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC4941, Sep 2007.

[RFC4982] M. Bagnulo & J. Arkko, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses", RFC4982, July 2007.

[RFC5534] J. Arkko & I. van Beijnum, "Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming", RFC5534, June 2009.

[RFC6164] M. Kohno and others, "Using 127-bit IPv6 Prefixes on Inter-Router Links", RFC6164, April 2011.

[ILNP-ADV] R. Atkinson & S. N. Bhatti, "Optional Advanced Deployment Scenarios for ILNP", draft-irtf-rrg-ilnp-adv, July 2012.

ACKNOWLEDGEMENTS

Steve Blake, Stephane Bortzmeyer, Mohamed Boucadair, Noel Chiappa, Wes George, Steve Hailes, Joel Halpern, Mark Handley, Volker Hilt, Paul Jakma, Dae-Young Kim, Tony Li, Yakov Rehkter, Bruce Simpson, Robin Whittle and John Wroclawski (in alphabetical order) provided review and feedback on earlier versions of this document. Steve Blake provided an especially thorough review of an early version of the entire ILNP document set, which was extremely helpful. We also wish to thank the anonymous reviewers

of the various ILNP papers for their feedback.

Roy Arends provided expert guidance on technical and procedural aspects of DNS issues.

RFC EDITOR NOTE

This section is to be removed prior to publication.

Please note that this document is written in British English, so British English spelling is used throughout. This is consistent with existing practice in several other RFCs, for example RFC-5887.

This document tries to be very careful with history, in the interest of correctly crediting ideas to their earliest identifiable author(s). So in several places the first published RFC about a topic is cited rather than the most recent published RFC about that topic.

Author's Address

RJ Atkinson
Consultant
San Jose, CA
95125 USA

Email: rja.lists@gmail.com

SN Bhatti
School of Computer Science
University of St Andrews
North Haugh, St Andrews
Fife, Scotland
KY16 9SX, UK

Email: saleem@cs.st-andrews.ac.uk

Expires: 10 JAN 2013