RN/05/26
28 October 2005

Research
Note

A Revised Overview of the Identifier-Locator Network Protocol (ILNP)

R. Atkinson

Telephone: +44 (20)7679-7214
Fax: +44 (0)20 7679 1397
Electronic Mail: R.Atkinson@cs.ucl.ac.uk
URL: http://www.cs.ucl.ac.uk/

Abstract

This document provides an overview of the Identifier-Locator Network Protocol (ILNP), using the Internet Protocol, Version 6 (IPv6) as a reference point. The key difference between ILNP and IP is that the IP Address is replaced by two distinct namespaces. The new Identifier namespace is used only for the identity of a given ILNP node. The new Locator namespace is used only for routing packets to a given ILNP node. ILNP has a new, much more crisp, definition for an Address. The document focuses on describing the areas in which ILNP is different from IPv6 and the reasons for those differences.

Keywords

identifier, locator, network, protocol, mobility, multi-homing, ILNP

Department of Computer Science
University College London
Gower Street
London WC1E 6BT, UK

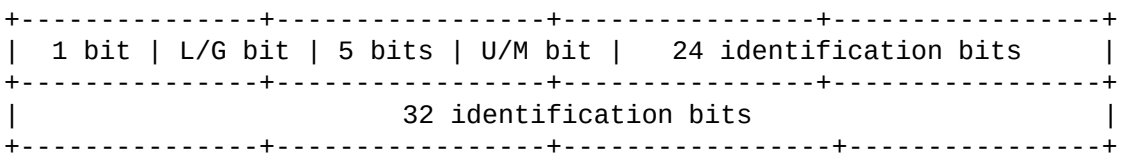

Figure 1: IEEE EUI-64 Format

1 Introduction

This document provides an overview of the Identifier-Locator Network Protocol (ILNP). This experimental protocol has been developed at UCL as part of research into new networking technologies and naming in distributed systems. ILNP is intended to provide the same basic capabilities as IP. Unlike IP, ILNP is designed to support multi-homed sites and mobile nodes as first-order capabilities of the network protocol. Further, the design of ILNP should eliminate the need for much of the entropy, in the form of covered more-specific prefixes, that exists in the current Internet routing table.

Because many of the features of this protocol are similar to or equivalent to the Internet Protocol version 4 (IPv4) [Pos81a] or the Internet Protocol version 6 (IPv6) [DH98], the document will use IPv6 as a baseline protocol reference. This permits editorial focus on describing the differences between this protocol and IPv6. For any topic that is not discussed here, it is very likely that there is no significant difference between IPv6 and ILNP in that area.

It is probably not realistic to believe that the current IPv4 Internet will be replaced soon by any other protocol, whether IPv6, the proposed Host Identity Protocol, or this proposal. However, it is easily possible that a particular private network might replace IPv4 with ILNP or IPv6 if there were compelling advantages to doing so. In such a case, it is likely that a Network Protocol Translator (NPT) could be built to enable users of the private network to communicate with nodes connected to the IPv4 Internet. Such a feature already is commonly requested of IPv6 router implementers, for example. Two likely candidates for initial deployment of ILNP might be a private mobile phone network or a military tactical network.

1.1 Key Concepts

At present, the IP Address is used both to indicate location for routing packets and also for identity of networked nodes.[Sho78, Coh78a, Coh78b] The overloaded semantics of the IP Address create architectural issues and limitations.[Sal93, Chi99] ILNP is designed to split the Address into a Locator, used only for routing, and an Identifier, used only for node identity.[1] Hence, the key protocol difference from IPv6 is that ILNP splits the 128-bit IPv6 address into a 64-bit Locator and a 64-bit Identifier.[2]

The 64-bit Identifier is an *IEEE Extended Unique Identifier (EUI-64)*.[IEE01] The EUI-64 specification provides a non-unique numberspace by having a scope bit that indicates whether a given identifier has local-scope or global-scope. One can form an anonymous identifier where the scope bit has been set to local. Alternatively, the EUI-64 identifier will have the scope bit set to global and will be formed from an IEEE 1394 (”Firewire”) Media Access Control (MAC) Address or from an IEEE 802 (LAN) MAC Address.[IEE95, IEE02, IEE99] The EUI-64 also has a bit reserved to indicate whether an Identifier name a single node (i.e. unicast) or a set of nodes (i.e. multicast). In normal operation, the EUI-64 Identifier probably will be composed from an IEEE 802 MAC Address. The Identifier has no topological significance and is treated as an opaque object.

The Locator is topologically significant and is used to route traffic to a single sub-network containing one or many hosts. Only the Locator is used within the routing system. For multicast traffic, the Locator indicates the location of a Rendevous Point (RP) that is aware of that multicast group, while the Identifier indicates the multicast group. Transport layer protocols bind their session state to the Identifier and do not use the Locator in any way. Applications generally use Fully Qualified Domain Names (FQDNs) and may use Identifiers. The new Application Programming Interfaces (APIs) designed for ILNP are oriented around FQDNs and generally eliminate the need for applications to be aware of location information such as an IP Address.

[1] This idea of splitting the address is not itself new, but until very recently no significant exploration of the idea had occurred.

[2] Mike O’Dell first proposed this split in the late 1990s, but at that time the IETF IPv6 Working Group dismissed the idea as being infeasible. His proposal was incomplete, not fully describing the new system or how security issues would be addressed. This research seeks to demonstrate its feasibility and discuss how the security issues are mitigated.

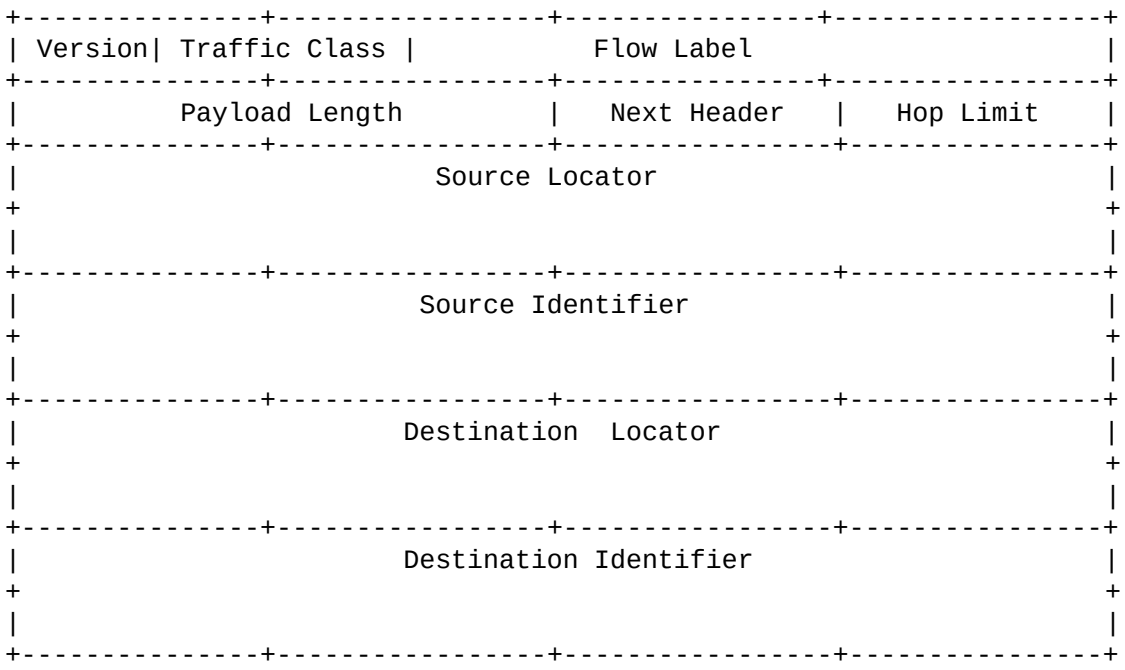

Figure 2: ILNP Header Format

2 ILNP Packet Formats

ILNP has a header format very similar to the IPv6 header format. The primary difference is the split of the Address into separate Locator and Identifier fields. Each field is the same size and in the same location as for the IPv6 header, except for the split of each 128-bit address into a 64-bit Locator and a 64-bit Identifier. In Figure 1, each horizontal row represents 32-bits of header, with a vertical mark every 8 bits. The Version field is 4 bits. The Traffic Class field is 8 bits and is equivalent to the IPv4 Type of Service field or the IPv6 Traffic Class field. The Flow Label field is 20 bits.

ILNP also adds a Destination Option that carries a session-specific nonce that is used to increase forgery resistance for packets that are not cryptographically authenticated. This option contains an Option Type, an Option Data Length of 6, and Option Data. The Option Data contains a 48-bit random nonce. The nonce must be unpredictable and should be cryptographically random.[ECS94]. The initiator of a session selects the initial nonce value for that session and sends it to the responder using this Destination Option. Anytime either party changes its Locator (or any member of its set of Locators), an ICMP New Locator message is sent that includes this option with the initial nonce value repeated. While this nonce can not prevent all forgeries, it is an effective measure against off-path forgery attempts, resulting in security equivalent to normal IPv4 or IPv6 packets without cryptography. To have full protections against forged packets with IPv4, IPv6, or ILNP, one must use some form of cryptographic security (e.g. Authentication Header).

ILNP adds a new ICMP message type, known as the *ICMP New Locator* message. This control message contains the new Locator(s) and optionally contains any deprecated Locator(s). This message has several purposes. It is central to ILNP mobility. It also permits a multi-homed node to tell a correspondent to use a different path, for example when one path fails or in order to balance load among several paths.

3 Identifiers and Locators

The critical difference in ILNP, as compared with either version of IP, is the separation of location from identity. It is believed that this crisp separation is an improvement in several dimensions. ILNP provides support for mobile nodes as a native property of the system, rather than requiring extra protocols and special-case mechanisms of Mobile IPv4 or Mobile IPv6. The new system eliminates the need for additional more-specific prefixes in the default-free zone of the inter-domain routing table; such special-case prefixes are currently about 20-25% of the total prefixes. The addition of a new Identifier namespace is also helpful for other purposes, such as security and distributed computing. The redefinition of the transport-layer checksum coverage (e.g. TCP pseudo-header calculation) removes the location information, so transport-protocols should work through a Network Address Translator (NAT) without special

handling. The new networking APIs discourage applications from using any location information, thus facilitating the development of applications that will not break when their traffic traverses a NAT.[TS00, SE01, HS01, Sen02]

A network-layer Identifier names a node, or, only in the special case of a multicast Identifier, a set of nodes. A single node in the network may have multiple Identifiers at the same time. An Identifier might be locally generated, in which case the local bit of the EUI-64 will be set.[3] An Identifier might represent a multicast group, in which case the multicast bit of the EUI-64 will be set. In normal operation, a node's Identifiers will be derived from its IEEE MAC addresses. Deriving the Identifier this way eliminates the need for a separate Identifier allocation process and also eliminates the need to perform Duplicate Address Detection (DAD) with Neighbour Discovery. It is not required that a given Identifier only be used for packets sent or received from the link layer interface associated with the MAC address used to form that Identifier. Identifiers name nodes, and do not name interfaces. The combination of a Locator and an Identifier names an interface. Thus, the Identifier, which names a node, is a bit different from an IP address, which names an interface on a node. When a node is using multiple Identifiers concurrently, each interface will have multiple names. The new networking application programming interface for use with ILNP replaces the address with the fully-qualified domain name. So an Identifier does not exist as an object in the normal application programming interface for ILNP.

An ILNP network Locator is an unsigned 64 bit quantity that names a single subnetwork. A subnetwork contains one or more nodes that all share the same point of network attachment. The Locator is similar to a 64-bit IPv6 routing prefix. As with an IP routing prefix, the Locator value is always topologically significant. Each active node has at least one valid Locator. A multi-homed node will have more than one valid Locator at the same time. Locators are used only for routing packets. Locators are allocated in the same way that provider-oriented IPv6 routing prefixes are allocated. The new networking application programming interface for use with ILNP replaces the address with the fully-qualified domain name. So a Locator does not exist as an object in the normal application programming interface for ILNP. Also, Locator information is not included in the transport-layer state with ILNP.

Each ILNP session has a nonce value included in its network-layer session state. A new network-layer option is used to send a nonce value among the nodes legitimately participating in a given network-layer session. This nonce mechanism limits the potential attackers of the session to the set of nodes that are directly along the path(s) used for that IP session by the corresponding nodes. Hence, the security risk of an ILNP session that does not use cryptographic authentication is no worse than of an IP session that does not use cryptographic authentication. With either ILNP or IP, the use of network-layer cryptography (e.g. Authentication Header) is optional and provides much stronger security protections and risk reduction.[Atk95]

3.1 Concept of Operations

With ILNP, the host initiating a session performs the same set of actions as with IP. First the host chooses which remote host(s) to begin communicating with. Once that is done, the Domain Name System (DNS) is used to map from a Fully-Qualified Domain Name to one or more Identifiers and to one or more Locators. However, the initiator's Transport Layer session state will contain only the Identifier in use, not the Locator(s). At the top of the Network Layer, there is a mapping table from each Identifier currently in use to a set of one or more Locators corresponding to that Identifier. This mapping table has entries for the Fully-Qualified Domain Name, Identifier(s), and current Locator(s) of each remote endpoint for each current IP session. The initiator then sends an initial packet (e.g. TCP SYN, UDP) to the responder at one of the Locators provided by the DNS.

The enhanced DNS supports mapping from a Fully Qualified Domain Name (FQDN) to a set of Identifiers and to a set of Locators. It also supports a reverse mapping function that provides a service equivalent to the current DNS PTR record lookup.

Unlike with IP, this initial packet will contain a Nonce Destination Option that sends a cryptographically random nonce value to the responder.[ECS94] The initial packet will also contain a FQDN Destination Option that provides the initiator's domain name to the responder. The responder will process that packet normally (e.g. sending a TCP SYN back if the initial packet was a TCP SYN), but will also save the nonce and FQDN information in network-layer session state for that session.[4] The responder will also initialise one or more entries into the Domain-Name/Identificr/Locator mapping table inside the responder's networking subsystem. The transport-layer session state created on the responder will contain the Identifier of the initiator, and usually the Domain Name of the initiator, but not the Locator (or Address). If the responder sends a reply packet for that session (e.g. TCP SYN), then the first reply packet will send the same nonce value back to the sender in a Nonce Destination Option. Normally, whenever

[3] It is possible that a locally generated identifier might be anonymous.

[4] The FQDN information will initially be marked as untrusted. It will be marked as trusted only after the remote party has cryptographically authenticated using a public key associated with that FQDN that can be obtained and authenticated via DNS Security mechanisms.

the session is initiated by the local node, the transport-layer state will include the FQDN of the remote end of each session, in addition to the Identifier(s) of that FQDN. Caching the remote node's FQDN in this manner helps ensure that one node is not able to steal another node's sessions.

After the transport-layer session is established, the responder will usually seek to determine the fully-qualified domain name of the initiator and authenticate the binding of that FQDN and the Identifier. This can be done by using the new reverse lookup schema (using the PTRL and PTRI records described below) and also with the ICMP FQDN messages. This information is kept cached inside a special mapping table at the top of the network-layer implementation. It is important that an implementation carefully marks authenticate information as authentic and not yet authenticated information as not yet authenticated.

Anytime that a control packet (e.g. ICMP) is sent from one node to one or more nodes within a given IP session, the Nonce Destination Option is included, for the security reasons described above. Ordinary data packets do not need to include the Nonce Destination Option and would not do so in normal operation.

3.2 Multi-homing Considerations

With ILNP, all Locator prefixes are delegated from a regional registry, such as RIPE or APNIC, to network service providers. In turn, each network service provider will delegate a Locator prefix to each end site or end user that is directly connected to that network service provider. In short, all ILNP Locators are provider-oriented, which facilitates maximal Locator prefix aggregation. To provide a concrete example, RIPE might delegate a particular 32-bit Locator prefix to the Joint Academic Network (JANET), which in turn might delegate a 48-bit Locator prefix from that space to University College London (UCL), which in turn might delegate a particular 64-bit Locator to a particular physical ILNP sub-network.

Hence, a multi-homed site will have a distinct Locator prefix delegated from each of its upstream network service providers. Each host within that multi-homed site may have a Locator from each of these upstream network service providers. When all of those providers and their connections to the site are working properly, each host within the site may use any of the valid Locators in communications. We define a *multi-homed node* to be a node that has more than one valid Locator at a that moment. This generalises the definition of multi-homing. A node might be multi-homed one moment and not multi-homed another moment.

If one of those providers, or its connection to the site, cease working properly, the host within the site will (1) update the set of Locators in its DNS entry to remove the Locator that no longer works properly and (2) send an ICMP Locator Update control message containing each currently valid Locator to each of its correspondents. The DNS update can be handled securely by using the Secure DNS Dynamic Update mechanism that has already been defined.[Wel00b] Updating the DNS is important for new remote correspondents who have not yet performed the DNS query to determine the valid set of Locators for this Fully Qualified Domain Name. Sending the ICMP New Locator messages is important for existing correspondents. The ICMP messages will include the revised set of valid Locators and a Nonce Destination Option containing the correct nonce value for that session. The remote correspondent will receive that ICMP message and then check whether the received Nonce value is correct. If the nonce value is correct, then the remote correspondent updates the network-layer session state for that ILNP session, updating the set of Locator values for that identifier in the internal mapping table as directed by the ICMP Locator Update message that was received. For stronger protection and greater risk reduction, the Authentication Header can be used to provide cryptographic protections to such ICMP messages. If AH is in use for a session, this will be noted in the network-layer session state for that session, in order to prevent a forged packet from being able to alter the network state and attack the communicating nodes and their communications session.

This same mechanism can be used to migrate individual ILNP flows from one path to another path if desired, for example as a form of load sharing among several upstream links or among several network interfaces to the multi-homed node. As with the currently deployed IP-based Internet, packets often will travel along asymmetric paths between a pair of nodes.

3.3 Mobility Considerations

In many cases, mobility is best solved using lower-layer (e.g. link-specific) techniques. For example, if a laptop were connected to the network via a mobile phone data link, it need not bother with changing its network-layer location so long as it remained at the same link-layer location (e.g. mobile phone number, wireless Ethernet subnetwork). However, there are cases where lower-layer techniques are not sufficient. In these cases, network-layer mobility mechanisms are very useful. This section describes how network-layer mobility works in ILNP.

Mobility in ILNP works in a manner very similar to multi-homing, except that a mobile host might not be multi-homed. A multi-homed host is one that has more than one valid Locator at the same instant in time. A mobile node is not required to have a *Home Locator*, unlike Mobile IPv4 or Mobile IPv6. If a mobile node happens to have a *Home Locator*, that Locator normally should be the first **L** record listed for that mobile node's domain name.

When a mobile node changes its Location, either to add a newly valid Locator, to delete a newly invalid Locator, or both, the mobile node (1) registers its presence with other ILNP nodes on that link, (2) updates its DNS entry at its' authoritative DNS server to reflect the new Locator list, and (3) sends an ICMP Locator Update control message to each of its current correspondents (as determined from the IP-layer session state). The ICMP Locator Update message contains all currently valid Locators for the originating node. That ICMP message always contains a valid Nonce Destination Option and optionally may also use the Authentication Header to provide cryptographic authentication. The recipient of such a message first validates the nonce (and if present the AH information), then if the message is authentic the recipient updates its internal network-layer session state information to reflect this latest information. As part of normal registration with a local link, the node first will perform Neighbour Discovery operations (or link-specific equivalent processes) and then register itself with the authoritative (reverse) DNS Server for that link.

3.4 Packet Forwarding

Normal IP-style longest-prefix match forwarding is used with ILNP unicast packets. The matching is done on the Destination Locator. The primary difference is that the Locator specifies a single sub-network, not a single host. When the packet reaches an ILNP router with an attached interface having a Locator that matches the Destination Locator, the router uses the Destination Identifier to determine the destination node. Typically, this last-hop router will maintain an internal table that maps each directly-connected node's Identifier to a layer-2 address and layer-2 protocol. [5] Neighbour Discovery provides a standard method for maintaining such mappings in the last-hope router when the interface is using LAN media. Using such a table, the last-hop router will forward the packet to the destination node using some link layer technology (e.g. Ethernet, Wireless Ethernet).

For multicast traffic, the Destination Locator specifies a sub-network having a rendezvous point for the multicast group specified in the Destination Identity field. In the special case of source-specific multicasting, the Source Locator and the Destination Locator will be identical at the originating node. A given multicast group must have at least one rendezvous point and may have more than one rendezvous point. A Destination Identity specifies a single multicast group.[6] The Destination Locator may be rewritten in transit, so any session state should be bound to the Destination Identifier for the group, not to the Locator. The set of routers at the Destination Locator sub-network communicate among themselves so that each is aware of which router handles each given Destination Identifier multicast group.

4 Network Subsystem

This section discusses the changes to the set of networking protocols used in the new Identifier/Locator system. In doing so, it further clarifies ways that the new system varies from the IP-based systems. Suggestions on preferred implementation strategies are also presented.

In a traditional UNIX implementation, the DNS resolver is an application-layer library outside the kernel, the BSD Sockets API or the X/Open TLI API is used by applications to request and receive network services across the kernel boundary, and the transport-layer protocols (and protocols below the transport-layer) are inside the kernel. While this specification does not mandate a particular manner of implementing the new schema, the schema does make certain implementation approaches easier and other approaches more difficult.

In the preferred implementation approach, the DNS Resolver becomes a kernel service and other portions of the DNS protocol (e.g. Secure Dynamic DNS Update) also move into the kernel. The historic implementation of the DNS protocol in application space appears largely to be an artifact of DNS having been implemented separately from and later than the rest of the BSD networking protocols.[LMKQ89, MBKQ96] This new approach helps ensure that applications do not need to be aware of lower-layer protocol information, such as the location of the correspondent node. This approach also avoids the need for a kernel-space to application-space call-back to perform essential DNS operations. The kernel also contains an internal mapping table from each Identifier to the set of Locators associated with that Identifier. This table is initialised by the DNS resolver service (which is now part of the kernel) and is updated from time to time by the new *ICMP Locator Update* messages. The table also includes the current nonce

[5] The layer-2 address used here is not necessarily the same as the Destination Identifier for that session. For example, the Destination Identifier might have been formed from some other interface's MAC Address.

[6] This supports 2^{23} concurrent multicast groups with global-scope plus an equal number with link-local scope.

value for that session and, where known, the fully-qualified domain name for each Identifier in that session. Keeping the FQDN in that same table helps distinguish sessions and nodes in the rare case where more than one node is using a given Identifier.[7]

In the new schema, the transport-protocol implementations are only aware of the Identifier(s) associated with a network session; they are not aware of the location of any remote endpoints or of the local system. This is central to successful operation because of the very different approaches being taken to node mobility and to multihoming. It also represents good software engineering practice by hiding details that the transport-layer protocols do not need to be aware of. Also, implementers should consider moving the Dynamic Host Configuration Protocol (DHCP) client service into kernel space, though the DHCP server should remain in application space. If the DHCP client service is moved inside the kernel, sysctl(2) or similar mechanisms will be needed so that the system administrator can configure the DCHP client appropriately.

4.1 Domain Name System Changes

The existing **A**, **A6**, **AAAA**, and **PTR** resource records of the Domain Name System (DNS) are not used with an ILNP deployment. Each IP address resource record type is obviated by the new **L** record that contains a 64-bit Locator associated with a Domain Name and by the new **I** record that contains a 64-bit Identifier associated with a Domain Name. A single Domain Name must have at least one **L** record and at least one **I** record and may have multiple of each of those two DNS resource record types. The **I** record will normally be long-lived, while the **L** record might change frequently (i.e. each time the sub-network attachment points of a node change).

The current DNS has a special ”pointer” (PTR) record that is used to determine the fully-qualified domain name (FQDN) associated with a given IP address. In the new architecture, the original PTR record is no longer used and is replaced by two separate records. The PTRL record is used to determine the authoritative DNS server for a given Locator. The PTRI record is used to determine the fully-qualified domain name associated with a given Identifier at a given Locator. Because Identifiers do not have any topological structure, it is not practical to perform a flat search directly for an Identifier. Instead, the node seeking to perform the reverse lookup first performs a PTRL lookup to locate the authoritative DNS server for the current location of the target node. Once that has been determined, the node seeking to perform the reverse lookup then performs a PTRI lookup to determine the FQDN, if any, associated with that Identifier at that location. The result of a PTRI lookup includes a flag that indicates to the recipient whether or not the target node has cryptographically authenticated its FQDN identity to the authoritative DNS server for that Location.

Most of the impact of these changes to pointer lookups is inside the DNS resolver. For example, only the resolver needs to retain any state across the two lookups (i.e. first PTRL, then PTRI). This additional state is probably not significant. For example, in the current DNS system, a node’s resolver needs to be able to first perform a CNAME lookup on a target domain name and then perform an address record (A, A6, AAAA) on the name resulting from the CNAME lookup.

In the new system, the DNS resolver service and the Secure Dynamic DNS Update client service are typically implemented inside the kernel for efficiency and to avoid the need for the kernel to make an up-call into an ordinary user space application. This does tend to increase the size of the kernel, but it might also help improve performance of DNS resolution by reducing the number of kernel-space/user-space context switches needed to resolve a name into an address (e.g. in the case of a CNAME record for the target domain name). This change also should help increase the assurance of the trusted computing base provided by the system,, since in most operating systems kernel-space functions are better protected from unprivileged tampering than user-space functions. [8]

To aid the use of DNS to authenticate identity and the implicit FQDN:Identifier:Locator bindings, two new ICMP message types are defined. The ICMP FQDN Request message asks a remote node to supply its own Fully Qualified Domain Name(s). If there is a current session between the two nodes, the reply will normally be sent using the FQDN Destination Option with some packet being sent as part of that active session. The FQDN Destination Option in turn requires that the Nonce Destination Option be present. The ICMP FQDN Response message may used to reply to the ICMP FQDN Request message if there is no current session between the two nodes and may contain one or more FQDNs.[9] The FQDN value received via either the FQDN Destination Option or via the ICMP FQDN

[7] There are two cases where an Identifier might not be unique. First, it is possible that an Ethernet interface manufacturer would have a manufacturing error causing more than one device to have a single Ethernet MAC Address. Second, a malicious adversary might be trying to impersonate another node. The second case is more likely than the first. Keeping the FQDN in the table and authenticating all the information before putting it into the table helps mitigate this issue.

[8] Microsoft Windows might be an exception to this.

[9] Depending on local security policy, the recipient of an ICMP FQDN Request from a source that does not have a current ILNP session with the target may ignore the request.

Reply message is not trusted by the recipient, but is instead treated merely as a hint. The validity of the hint can be verified using the Domain Name System (DNS) to look up I, L, PTRI, and PTRL records for the hint and comparing that information with the received ILNP packet header and any locally cached information at the recipient. A host that does not have any FQDNs will only be reachable via link-local addressing and only from nodes on the directly attached sub-network.

To assist in bootstraping key management there are also two new certificate messages.[10] The ICMP Certificate Request message is used by an initiator to request the public-key certificate of the responder. The initiator MAY also gratuitously supply its own public-key certificate to the responder inside the ICMP Certificate Request message. The ICMP Certificate Response message is used by a responder to provide its public-key certificate to the initiator. These messages are not themselves sensitive, though any certificates supplied via this mechanism will need to undergo normal certificate validation checks against known valid certificate information before such certificates could themselves be trusted. Such validation checks should include the certificate being signed by a trusted party, should include checks against certificate revocation lists, et cetera.

Some computers that currently use the Internet do not have a stable Fully Qualified Domain Name (FQDN). For example, a computer that only connects to the Internet via a dialup telephone modem connection probably does not have a permanent domain name today. With ILNP, such a system would not be required to obtain a permanent domain name, provided the network service provider created suitable DNS configuration a priori and supplied that information to the dialup computer (e.g. via PPP). However, such a system could not take advantage of the multi-homing or mobility features of ILNP without having its own, relatively permanent, domain name. Unlike IP addresses, there is no perceived shortage of domain names, so this does not represent an undue burden on either operators or end users. It is worth noting that a growing number of residential computers have always on connections through Digital Subscriber Line (DSL) modems or through cable modems. Such always connected computer systems often do have a relatively permanent domain name, though it is equally often a relatively unfriendly and non-mobile name within the namespace of the network service provider (e.g. c-12345.fairfax.va.cox.net). At present, most network service providers do not enable Secure Dynamic DNS Update.

At present, DNS caching is used to reduce the volume of DNS traffic and to reduce the latency of replies to DNS queries. Absent Dynamic DNS Update, one might have a relatively long cache time (e.g. hours or days). In the presence of Dynamic DNS or of ILNP, some information might not be cachable. In the case of the existing Dynamic DNS, A, AAAA, and A6 records could not have such a long cache lifetime. In the case of ILNP, L records also could not have such a long cache lifetime for similar reasons. The current DNS does not have per-resource-record granularity of cache time. It would be worthwhile to fix this for Dynamic DNS reasons. Such a fix would also help ILNP deployments.

5 Neighbour Discovery Changes

IPv6 Neighbour Discovery [NNS98] can largely be reused for ILNP, only a few modifications are required. In ILNP, the Locator is the equivalent of the IP routing prefix. So, with ILNP the Prefix Information option is always sent with a prefix-length of 64 and the autonomous address-configuration flag is always 1, indicating that autonomous address-configuration is always in use with ILNP. [11]

In the common case where each Identifier associated with a node has been formed from an IEEE MAC address of a hardware interface on that node, the node does not need to perform Duplicate Address Detection. In the highly unlikely event of a MAC address collision, for example due to manufacturing error with the Ethernet chipset, layer-2 communications problems consequently will develop. In turn, the absence of a working layer-2 communications channel obviates concerns about upper-layer Identifier conflicts. This change from IPv6 greatly reduces the normal-case handoff latency for a mobile ILNP node, without adverse affects. Further, Neighbour Discovery with ILNP omits the complex congestion avoidance methods defined in more recent versions of the IPv6 Neighbour Discovery specification. Instead, ILNP will rely on link-specific congestion management methods. For example, both wired Ethernet and wireless Ethernet already have link-layer congestion management mechanisms.[IEE02][IEE99] Operational experience with IPv6 Neighbour Discovery has shown that this recent addition to the specification creates problems and questions whether its absence in earlier versions of the specification was a problem.

To reduce security risks, a first-hop ILNP router should check the Source Identifier field of all outbound packets to be sure that the Source Identifier used actually exists behind the router interface that the ILNP packet arrives on. This is

[10] This concept comes from Sun's proposed Certificate Discovery Protocol (CDP).

[11] From an architectural perspective, classical Neighbour Discovery should only be used on link technologies having a link-layer broadcast facility (e.g. radio links, Ethernet, Wireless Ethernet). Other link technologies should have a different method of network-layer/link-layer address resolution.

easily performed by examining the Neighbour Discovery cache for that router interface. Because the first-hop ILNP router is a router, it will know which other nodes on the link, if any, are also valid IP routers that are thereby entitled to forward packets originating elsewhere.

When a node registers with the local subnetwork, the node also registers its Fully-Qualified Domain Name (FQDN) and Identifier with the authoritative DNS server for that location. The authoritative DNS server for that location needs this information so that it can properly respond to PTRI enquiries.

ICMP Redirects still exist in the new system. However, the only legitimate use is within the local link if multiple first-hop routers exist. If some node sends a packet to router A incorrectly, then router A might send a redirect message to the node indicating the node should send such packets towards router B. This message can be authenticated cryptographically if desired.

6 Transport Protocol Changes

Each transport-layer protocol is changed to bind the session state to the Identifiers and the transport-layer port numbers, and to omit the Locators from the session state. For example, this means the TCP pseudo-header binds to the Identifier, not to the Locator. Mapping from a given Identifier to its current set of Locators is performed at the top of the network-layer, not inside the transport-layer. The mapping from the fully-qualified domain-name used in the new networking API to the Identifier is performed outside the transport-layer protocol implementation. Because the DNS Resolver is now inside the kernel, that mapping is undertaken in the kernel just above the transport-layer protocol implementation. The transport-layer session state includes the Domain Name and Identifier(s) of each session. Only the Identifier is included in the transport-layer pseudo-header calculations.

If a valid Locator for a given node becomes invalid, for example because the node moved, then it is possible for packets to be lost during the time between the move occurring and the mobile node's ICMP Locator Update message being received and processed by the correspondent node. This packet loss will be proportional to the handoff latency of the mobile node's move and also to the frequency of packet transmission from the correspondent node to the mobile node. Since ILNP and IP are each connectionless datagram protocols that do not provide reliability guarantees, this is potential temporary packet loss is no worse than in the current Internet. Any mechanism to mitigate this issue for IP probably also could be used with ILNP. Attempting to preclude any packet loss at the network layer would violate the service definition and also would violate the end-to-end principle.[SRC84]

At present, the Transmission Control Protocol (TCP) assumes there is only one path between the sender and the receiver, so TCP infers congestion state from various properties of a given TCP session.[Pos81b] By contrast, the Stream Control Transmission Protocol (SCTP) supports transport-layer multi-homing.[OY02] With ILNP, there might be multiple unrelated or partially related paths between the sender and receiver of a given TCP session, so this assumption frequently might be invalid. In this way, TCP over ILNP has some similarity with the existing SCTP over IP combination. Implementation-specific optimisations are often possible for this situation. For example, if the network-layer detects a location change for the remote node of a given session, the network-layer might provide a hint to the transport-layer that the cached congestion state is likely invalid.

The Address List feature of SCTP is designed to permit session multi-homing through transport-layer mechanisms, even though one might not have network-layer multi-homing support. As such, this feature of SCTP is not required or useful in the context of an ILNP deployment. One could get the equivalent capability by using TCP over ILNP.

The architectural improvements provided by ILNP also present opportunities to create new and improved transport-layer protocols. While ILNP can work fine with existing transport-layer protocols, new and improved transport-layer protocols might provide significant further improvements in the user experience. Such work is beyond the scope of the current research project, but is noted here as an opportunity for future work.

7 Application Changes

The BSD Sockets API for networked applications was developed before the development and widespread deployment of the Domain Name System (DNS). Hence, existing networked applications are responsible for name to address translations within the application – and so necessarily include that information as part of the application state. Good software engineering indicates that it would be better if the address information could be hidden from applications that do not absolutely need to know that information. In this regard, the historic BSD and TLI networking interfaces are deficient.

So with the new system, there is a new networking API, one that uses Domain Names rather than IP addresses. This new API can be used for IPv4, IPv6, or ILNP network sessions without requiring any modification to the

application source code. The translation from a fully-qualified domain name (FQDN) into Identifiers and Locators is performed by the kernel software that implements the networking API. That software calls the DNS Resolver, which is now inside the kernel, to determine the appropriate Identifier(s) and Locator(s), passes the Identifier(s) down to the transport-layer and passes the initial Identifier::Locator mappings to the network-layer to be cached there.

Some specialised applications, for example traceroute(8), need to have access to a lower-level API. The raw sockets API still exists in the new system to meet those needs.

8 Differences from Current IETF Work

This section describes in summary how ILNP is different from the current Mobile IPv4 standards, the current Mobile IPv6 standards, current IPv4 multi-homing practices, and current IETF work to alter IPv6 multi-homing.

8.1 Differences from Mobile IPv4

Mobile IPv4 uses a complex architecture involving at least 3 cooperating nodes. The *Mobile Node* is the system trying to communicate with other Internet nodes. Each mobile node has a *Home Agent* that provides forwarding of packets addressed to the Mobile Node whenever the Mobile Node is not connected to the Home Agent's subnetwork. The *Foreign Agent* is on the same subnet as the Mobile Node and provides packet forwarding for the Mobile Node whenever the Mobile Node is not on the Home Agent's subnetwork.

In a typical communications session, some correspondent that desires to communicate with the Mobile Node sends packets addressed to the Mobile Node's permanent (or home) IP address. If the Mobile Node is at home, it receives the packets normally and responds normally. However, if the Mobile Node is not at home, the Home Agent receives the packets addressed to the Mobile Node and forwards them via some tunnelling protocol to the Mobile Node at its (presumably temporary) foreign location. Packets from the Mobile Node when it is not at home are sent via the Foreign Agent at the remote location. Because the Foreign Agent and Home Agent are not the same node at the same time, packets suffer from triangle routing. Packets inbound to the Mobile Node travel via its Home Agent, whilst packets outbound from the Mobile Node travel directly to their destination. Whenever the Mobile Node moves, it must send control messages to the Home Agent informing the Home Agent of the new foreign address of the Mobile Node.

Any architecture that requires a Home Agent creates an issue with triangle routing, whereby packets in one direction travel first from Correspondent to Home Agent, who in turn tunnels the packet to the Mobile Node. The Mobile Node often needs to tunnel from its care-of address to the Correspondent Node in order to pass unicast Reverse-Path Forwarding (uRPF) security checks against packet forgery. Each instance of tunnelling packets increases latency. Sending the packets from Correspondent to Mobile Node via the Home Agent also increases latency. Further, because there is no good mechanism for an application on the Mobile Node to realise whether the node is remote or at home, there is also no good way for the application to perform source address selection. This means, for example, that a DNS query and reply will be tunnelled via the Home Agent rather than being sent directly from the Mobile Node. This increased DNS latency means that it will take longer to setup new sessions than would normally be the case, whenever DNS resolution is needed as part of session creation.

This scheme is quite complex and has numerous security issues (e.g. how to authenticate change of address control messages between the Mobile Node and Correspondent Node or between Mobile Node and Home Agent). Further, unlike ILNP mobility, tunnelling of certain packets between co-operating nodes is required. As tunnelling of packets normally causes packet fragmentation, this can create performance problems in addition to the obvious increase in required network bandwidth for a given amount of user data. For these and other reasons, Mobile IPv4 is not widely implemented or widely deployed in today's Internet.

8.2 Differences from Mobile IPv6

Mobile IPv6 has a complex architecture similar to Mobile IPv4. There are some differences from Mobile IPv4. This discussion will focus on the areas in which ILNP is different from the unique aspects of Mobile IPv6.

Mobile IPv6 extends the overloading of the IP Address by using some IP Addresses primarily for identification and other IP Addresses primarily for routing. However, both kinds of IP Address share a single namespace. With Mobile IPv6, each mobile node has a single permanent address which is always used for identification and is sometimes (i.e. only when actually at "home") used for routing packets. Additionally, each mobile node that is not at "home" has a second IP address that is used only for routing packets. With Mobile IPv6, correspondents normally send traffic to

the ”home address” and an agent forwards them along to the current location, but replies to the correspondents travel directly. This creates a situation with ’triangle routing’ which is necessarily inefficient and has numerous potential issues with commonly deployed site security measures (e.g. NAT, firewalls, etc.).

One of the issues with Mobile IPv6 is that IPv6 Neighbour Discovery requires that a node first appearing on a link perform Duplicate Address Detection (DAD). While the DAD process works fine with a non-mobile node, since it only need be performed once, at system boot time, DAD can be an issue for mobile nodes, particularly mobile nodes on wireless networks, which is perhaps the most interesting mobile scenario for NATO.[Mac03] In this last situation, the time required to perform DAD on arrival at each link can significantly increase the delay in the mobile node becoming reachable at its new location. In low bandwidth scenarios. wired or wireless, frequent use of DAD can consume significant amounts of link capacity. If a mobile IPv6 node unilaterally sends its binding update before DAD has completed, this has little benefit and can actually delay the mobile node from becoming reachable at the new location.[Mac03] Eliminating the need for DAD could significantly improve throughput and reachability of mobile nodes, but this is not permitted by IPv6 because identifier conflicts are not unlikely with IPv6. [TN98, ND01].

Further, IPv6 Neighbour Discovery (ND) is used on all kinds of link layers, including link layers that do not provide congestion management. So the designers require that IPv6 ND include a *random delay* of up to 1 second before responding to a ND query. This further increases latency and delays mobile node handoffs. ILNP avoids this by using different layer-2 address resolution methods on different link layers. When used over Ethernet, ILNP uses ND, but ILNP ND does not have that *random delay* before responding, because Ethernet has link-layer congestion management in the form of CSMA/CD.[IEE02, IEE99] When used over a point-to-point link, ILNP uses the PPP control protocols to determine the end point addresses[McG92, Sim94]. For point-to-point links, congestion management is straight-forward due to the nature of the link. For the majority of deployments, these ILNP changes greatly reduce the handoff latency and increase the effective throughput when communicating with mobile nodes.

With Mobile IPv6, handoff latency increases in *Route Optimisation* mode. Bandwidth overhead increases due to tunnelling from Home Agent to Mobile Node, from Mobile Node to Correspondent Node, and in Route Optimisation mode also from Correspondent Node to Mobile Node. With Mobile IPv6, the Return Routeability check means that if an assigned IPv6 home address is used for identification, then a change in that IPv6 home address will cause all existing sessions to be lost. ILNP does not have such a tradeoff, since all traffic is routed optimally always and no traffic needs to be tunnelled to a mobile node. The one issue with ILNP in this regard is that if both parties to a unicast ILNP session are mobile, and they both undertake a network-layer move at exactly the same time, then the *ICMP Locator Update* messages will be lost. In this very unlikely case, the nodes will need to timeout and then resend the *ICMP Locator Update* and/or perform a DNS resolution to determine each correspondent’s current Locator(s). Also, one should recall that link-layer mobility support, for example bridging with wireless LANs,[IEE99] means that not all physical movements are network-layer movements.

Further, the current work on Routing Optimisation work for Mobile IPv6 requires periodic (e.g. each 7 minutes) binding update messages to confirm its current location, even if the Mobile Node has not moved location. [NAA $^{+}$05]

Also, increasing the rate of Router Advertisement messages can decrease the handoff latency, at the cost of more than 20Kbps of background link bandwidth when the RA inter-arrival time is 30 to 70ms.[Mac03] It would be desirable to have a network-layer protocol that took these issues into account in the base protocol design, so that special mobility extensions would not need to be added later on. That might mitigate the operational issues that exist in Mobile IPv6 today and prevent Mobile IPv6 from being suitable for real-time traffic (e.g. packet voice).

8.3 Differences from current IP Multi-homing

At present, IPv4 multi-homing and IPv6 multi-homing each require that each multi-homed subnet advertise its more-specific prefix in the global IP routing table. This causes the global IP routing table to be much larger than it would be if one could only advertise the aggregated prefixes. In short, the current approach to multi-homing increases the entropy in the global IP routing table. This larger IP routing table has operational impacts on network service providers, both increased memory requirements in core routers and also increased convergence times for inter-domain routing changes. Because of the complexity of advertising these more-specific prefixes, many network service providers discourage multi-homing, either implicitly or explicitly by charging additionally for that service.

Users desire the network to be highly available. Multi-homing is a good mechanism for greatly increasing network availability, because it provides additional redundancy between the end site and the inter-provider core of the Internet. Therefore, improving the ease of deploying multi-homing and reducing the cost of multi-homing to the default-free-zone of the routing system are both significant benefits as compared with the current Internet Protocols.

8.4 Differences from current IETF work on IPv6 Multi-homing

Because this desirability is widely understood and the issues with current multi-homing methods are also widely understood, the IETF just began to re-examine how IPv6 multi-homing should be specified. This current IETF work to improve IPv6 multi-homing support, in the recently initiated IPv6 Shim Working Group, reuses the basic concepts of Mobile IPv6 to support hosts that are connected to more than one service provider concurrently. One IP address is used for identity, while either that address or other addresses might be used for routing packets. A shim protocol layer is inserted between IP and the transport-layer protocol above to translate the different kinds of addresses and ensure that the transport-layer only is aware of the address that is being used as an identifier. The ILNP protocol defined here provides a crisper and clearer separation of the Identifier and Locator roles, by having different objects and different namespaces for each role. This also means that none of the Locator namespace is wasted for use as an Identifier, and vice versa.

Huston has done a solid analysis of the architectural issues around several approaches to IPv6 multi-homing.[Hus05] He has noted that if one uses a provider-supplied IP address as one's identifier, then a change of transit provider will necessitate a change in identifier. Also, he has identified several issues with the interaction of the new proposed approach to IPv6 multi-homing and the Mobile IPv6 specifications. For example, MIPv6 has a Return Route-ability feature where the validity of the identity is checked by perodically sending packets to that address. With the new approach to IPv6 multi-homing, the identity could still be valid even though that address is not reachable.

8.5 Differences from Host Identity Protocol (HIP)

The Internet Research Task Force (IRTF) has an active Research Group looking into the *Host Identity Protocol (HIP)*. Both this work on ILNP and the separate work on HIP have their roots in the now concluded IRTF Namespace Research Group.[12] The HIP architects believe, as we do, that the improvements in the namespace design of the network architecture merits creating new Application Programming Interfaces (APIs) to replace both the traditional BSD Sockets interface and the X/Open Transport Layer Interface.

With HIP, the existing IP address (128 bits) is used as the Locator. With ILNP, only the routing prefix portion (64 bits) of the current IP address is used as the Locator.[MN05]

With HIP, one uses a *Host Identifier* that is actually the public-key associated with the host. This Host Identifier (i.e. the host's public key) does not have a fixed size. The Host Identifier is never used in protocol headers or on the wire. So, HIP also has a 128 bit *Host Identity Tag (HIT)* that is a cryptographic hash of the corresponding *Host Identifier* and can be used in protocols or on the wire. HIP also has a *Local Scope Identifier (LSI)* that can be used with existing APIs and existing protocols that only support a 32-bit value. ILNP Identifiers have different properties than HIP Identifiers. For example, the HIT is used in place of the IP address in the TCP or UDP pseudo-header checksum calculation.[MN05]

With HIP, an essential concept is to use cryptographically generated addresses. While this makes it easier to bind the Identifier to the Address, it also means that if the cryptographic key is compromised, then the identity must be abandoned upon discovery of the key compromise. While the HIP Architecture briefly mentions that non-cryptographic identifiers might be possible, the use of non-cryptographic identifiers is later dismissed because the document authors believe that non-cryptographic identifiers could not be authenticated properly.[MN05] None of the HIP specifications support the use of a non-cryptographic identifier. With ILNP, the Identifier is just a layer-2 MAC Address that belongs to the node. While this requires a bit more effort to authenticate the binding between the Identity, the Locator, and the node's Domain Name, it has the advantage that compromise of a cryptographic key only requires rekeying, and does not require changing the node's Identity, Locator, or Address. So the ILNP approach can be fully authenticated cryptographically when that is desired, yet it does not require that the identifier itself be cryptographically-generated and also does not require the use of cryptographic authentication all of the time. The author believes that the disadvantages of using cryptographically generated addresses outweigh the advantages. Further, by using an EUI-64 where the scope bit is set to local-scope and using a 63-bit cryptographic hash of the HIP Host Identifier, one could create a HIT using ILNP. This implies that it might be possible to also support the HIP with the ILNP, though a detailed analysis of this issue is beyond the scope of this paper.

Further, with HIP there is no non-cryptographic mechanism for authentication.[MN05] All communications must be cryptographically authenticated, regardless of the threat environment. While this is a very secure approach, it would be better to also have a light-weight non-cryptographic authentication mechanism, as ILNP does. Such a mechanism would eliminate significant computational burden (e.g. key exchange, signature verification) for sessions where strong authentication is not necessary.

[12] R. Moscowitz, originator of the HIP concept, was a member of the IRTF NSRG, as was the current author.

As with ILNP, transport-layer state no longer includes the location information, it only contains host identifiers. For the same reasons outlined in Section 2, this de-coupling of location from identity facilitates both scalable multi-homing and also mobility. Similarly, NAT is not a problem with HIP, just as it is not with ILNP. Unlike ILNP, the HIP proponents have not sorted out how HIP could be used with IP multicasting or application-layer multicasting.

HIP's use of IP-layer security (IPsec) disallows secure tunnelling of network-layer packets. Only host-to-host connections with IP-layer security (IPsec) are permitted with HIP. For example, transport-mode ESP is supported by HIP, but tunnel-mode ESP is not supported by HIP. [MN05]

9 Security Considerations

Unlike the current Internet, the Domain Name System is required for ordinary operation. This is not a substantial difference from the present for most users; experience at network service providers has consistently demonstrated that nearly all current users are unable to distinguish "network down" faults from "DNS error" or "DNS down" faults. This does mean that security and availability are even more important for the DNS in the new system. It also means that the new system is a bit more brittle than the previous system. This also is not a surprise as experience in computing has generally shown that more secure systems are more brittle than less secure systems. Fortunately, IETF work on DNS Security has progressed sufficiently to demonstrate that it is possible to secure the Domain Name System. DNS Security specifications have been published by the IETF.[AAL$^+$05a, AAL$^+$05c, AAL$^+$05b] There are also openly published specifications for Secure DNS Dynamic Update.[Wel00b, Wel00a]

In the previously mentioned case where more than one node claims a given Identifier value, the local node has no way to know whether more than one node is attempting to use a given Identifier at the same time. So, to prevent confusion (and identity forgery attacks), additional checks need to be made at session-establishment time and whenever a control message is received for a given session. If there is a current network-layer session using a given Identifier, no new session will be permitted using that Identifier unless the session initiation is successfully authenticated. In low-threat environments, this authentication might consist of verifying the Nonce currently associated with that Identifier is present in the Nonce Destination Option of the session establishment packets. In higher threat environments, one can authenticate cryptographically using the Authentication Header (AH). The modified Encapsulating Security Payload (ESP) is not suitable for authentication in this environment because it is unable to authenticate any network-layer options. So in this environment, ESP must not be used without encryption or for any purpose other than confidentiality. Any party to an ILNP session may require that cryptographic authentication be used for the session.

As a practical matter, most ILNP nodes will have and use public-key certificates that bind a public-key to their Fully Qualified Domain Name (FQDN) and are signed by a trusted party. That is, most nodes will probably use a Public Key Infrastructure (PKI) of some sort. While initial deployment of a PKI can be difficult, there are several large-scale PKI deployments today demonstrating that it is possible. For example, the US military has issued new identity cards that are smart cards containing electronic storage. Each of those new military identity cards contains a public key certificate and a private key for that individual. These are now universally used by both uniformed and civilian employees of the US Department of Defense. The certificates are used for electronic mail, for web site access, etc. Several other organisations have deployed and regularly use an internal PKI. With the recent enhancements to DNS Security, it should be possible to use any such PKI in combination with DNS Security.

If a control message is received that requests a Locator Update in the local node's tables for an Identifier with a current session with the local node, then that Locator Update must be authenticated before being accepted and processed. In a low risk environment, this will at least require validating that the Nonce received in a Destination Option with the control packet is correct and currently valid for the existing network session with that Identifier. In such a low-risk environment, there is a risk of packet replay or packet forgery from any party who knows the session nonce. In practice, knowledge of the session nonce is limited to the set of nodes along the path from originator to responder (and the reverse path), inclusive. In a higher risk environment, authenticating the Locator Update message will involve cryptographic authentication (e.g. using the Authentication Header) and including some form of packet replay detection/prevention. These checks help prevent one node from interfering with another node's current sessions with the local node.

For the case where a new session is being established and the remote system's Identifier is not associated with any current network session, the local node will normally validate the received Locator and received Identifier values by performing a DNS resolver call upon the Domain-Name provided by the remote system in the FQDN Destination Option. If the received Locator is not among the Locator(s) authenticated via the DNS call, the local node may decline to create the proposed new session. If the received Identifier is not among the Identifier(s) authenticated via that DNS call, the local node must decline to create the proposed new session.

In the new system, the networking protocol implementation is performing a larger set of security checks than the current TCP/IP protocol implementation performs. Should any of those checks fail, the error should be logged using system-specific auditing facilities (e.g. BSD System Logging) so that the security office responsible for the system can be aware of the errors and possible attacks on the node.

Most network service providers deploy ingress IP address filtering to prevent packets with forged IP addresses from entering their networks.[FS98] In ILNP, the equivalent function is performed by ingress Locator filtering. on the Source Locator of ILNP packets. The new multi-homing architecture used with ILNP makes it much easier to deploy and manage such address filters. In turn, this can significantly reduce the risk of distributed denial-of-service attacks that are based on network-layer packets with forged source Locators, as compared with the equivalent attacks with forged source IP addresses.

The use of Identifiers enables firewalls to have access control rules that are based on identity, rather than address or location. This might permit a corporate IT security manager to give the CEO's laptop more privileges than a network-capable ID badge reader, for example. Research into this area is beyond the scope of the current project, but might be considered as follow-on work.

In order to close or minimise covert channels, it is important that systems ensure that fields that are supposed to contain zero bits (e.g. fields that exist for alignment reasons) actually do contain all zero bits. In any reasonably complex system, covert channels will exist. The goal here is merely to reduce the bandwidth of such covert channels to some acceptable level of risk.

10 Conclusions

This document has provided an overview of the proposed Identifier-Locator Network Protocol (ILNP). This description of the new protocol has focused on identifying how and where the new network protocol is different from the Internet Protocol version 6 (IPv6), with some discussion of differences from IPv4. Several networking components that are currently implemented in user space in common operating systems will typically move into the operating system kernel in the new design. The new protocol integrates native support for mobility and multi-homing. Further, it removes significant existing network state (e.g. duplicate routing prefix entries needed for multi-homing) from the core of the network and thereby improves the scalability and performance of core routers within the network.

References

[AAL+05a] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, Internet Engineering Task Force, March 2005.

[AAL+05b] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, Internet Engineering Task Force, March 2005.

[AAL+05c] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, Internet Engineering Task Force, March 2005.

[Atk95] R. Atkinson. IP Authentication Header. RFC 1826, Internet Engineering Task Force, August 1995.

[Chi99] J.N. Chiappa. Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture. Unpublished draft, NameSpace Research Group (NSRG), Internet Research Task Force, 1999.

[Coh78a] D. Cohen. On Names, Addresses, and Routings. Internet Experiment Note (IEN) 23, ARPA Network Working Group, January 1978.

[Coh78b] D. Cohen. On Names, Addresses, and Routings (II). Internet Experiment Note (IEN) 31, ARPA Network Working Group, April 1978.

[DH98] S. E. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, Internet Engineering Task Force, December 1998.

[ECS94] D. Eastlake, S. D. Crocker, and J. Schiller. Randomness Recommendations for Security. RFC 1750, Internet Engineering Task Force, December 1994.

[FS98] P. Ferguson and D. Senie. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2267, Internet Engineering Task Force, January 1998.

[HS01] M. Holdrege and P. Srisuresh. Protocol Complications with the IP Network Address Translator. RFC 3027, Internet Engineering Task Force, January 2001.

[Hus05] G. Huston. Architectural Approaches to Multi-homing for IPv6. RFC 4177, IETF, September 2005.

[IEE95] IEEE. Standard for High Performance Serial Bus - Firewire. Standard 1394, IEEE, 1995.

[IEE99] IEEE. Standard for Local and Metropolitan Area Networks: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Standard 802.11, IEEE, March 1999.

[IEE01] IEEE. Standard for Local and Metropolitan Area Networks: Overview and Architecture. Standard 802, IEEE, December 2001.

[IEE02] IEEE. Standard for Local and Metropolitan Area Networks: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. Standard 802.3, IEEE, January 2002.

[LMKQ89] Samuel J. Leffler, Marshall Kirk McKusick, Michael J. Karels, and John S. Quarterman. *"The Design and Implementation of the 4.3 BSD UNIX Operating System"*. Addison-Wesley, Menlo Park, CA, 1989.

[Mac03] J.P. Macker. Interoperable Networks for Security Communications, Phase 1, Task 6. Final Report INSC-TASK6, North Atlantic Treaty Organisation (NATO), December 2003.

[MBKQ96] Marshall Kirk McKusick, Keith Bostic, Michael J. Karels, and John S. Quarterman. *"The Design and Implementation of the 4.4 BSD UNIX Operating System"*. Addison-Wesley, Menlo Park, CA, 1996.

[McG92] G. McGregor. The PPP Internet Protocol Control Protocol (IPCP). RFC 1332, Internet Engineering Task Force, May 1992.

[MN05] R. Moscowitz and P. Nikander. Host Identity Protocol Architecture. Internet-Draft draft-ietf-hip-arch-03.txt, IETF, August 2005.

[NAA+05] P. Nikander, J. Arkko, T. Aura, G. Montenegro, and E. Nordmark. Mobile IPv6 Route Optimisation Security Design Background. Internet-Draft draft-ietf-mipv6-ro-sec-03.txt, IETF, June 2005.

[ND01] T. Narten and R. Draves. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 3041, Internet Engineering Task Force, January 2001.

[NNS98] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). RFC 2461, Internet Engineering Task Force, December 1998.

[OY02] L. Ong and J. Yoakum. An introduction to the stream control transmission protocol (SCTP). RFC 3286, Internet Engineering Task Force, May 2002.

[Pos81a] J. B. Postel. Internet Protocol. RFC 791, Internet Engineering Task Force, September 1981.

[Pos81b] J. B. Postel. Transmission Control Protocol. RFC 793, Internet Engineering Task Force, September 1981.

[Sal93] J. H. Saltzer. On the Naming and Binding of Network Destinations. RFC 1498, Internet Engineering Task Force, August 1993.

[SE01] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022, Internet Engineering Task Force, January 2001.

[Sen02] D. Senie. Network Address Translator (NAT)-Friendly Application Design Guidelines. RFC 3235, Internet Engineering Task Force, January 2002.

[Sho78] J.F. Shoch. Inter-Network Naming, Addressing, and Routing. Internet Experiment Note 19, ARPA Network Working Group, January 1978.

[Sim94] W. Simpson. The Point-to-Point Protocol (PPP). RFC 1661, Internet Engineering Task Force, July 1994.

[SRC84] Jerome H. Saltzer, David P. Reed, and David D. Clark. End-To-End Arguments in System Design. *ACM Transactions on Computer Systems*, 2(4):277–288, November 1984.

[TN98] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 2462, Internet Engineering Task Force, December 1998.

[TS00] G. Tsirtsis and P. Srisuresh. Network Address Translation - Protocol Translation (NAT-PT). RFC 2766, Internet Engineering Task Force, February 2000.

[Wel00a] B. Wellington. Domain name system security (DNSSEC) signing authority. RFC 3008, Internet Engineering Task Force, November 2000.

[Wel00b] B. Wellington. Secure Domain Name System (DNS) Dynamic Update. RFC 3007, Internet Engineering Task Force, November 2000.