Mobility as an Integrated Service Through the Use of Naming

R. Atkinson
Extreme Networks
Santa Clara, CA
95051 USA
rja@extremenetworks.com

S. Bhatti
University of St Andrews
School of Computer Science
St Andrews, Scotland
KY16 9SX
saleem@cs.st-and.ac.uk

S. Hailes
University College London
Dept of Computer Science
London, England
WC1E 6BT
s.hailes@cs.ucl.ac.uk

ABSTRACT

As Mobile IP is deployed, so the requirements for its deployment evolve, reflecting the actual use of IP networks today. This includes the ability to use Mobile IP with IPsec, NATs and multi-homed networks. Furthermore, new requirements arise as people start to use IP in scenarios where the whole network is mobile (e.g. military networks), and where edge-networks may not be IP-enabled (e.g. sensor networks), but there is a requirement to interoperate across an IP network. In all these cases, rather than engineering retro-fits, creating an increasingly complex network landscape with possible unforeseen feature interactions and dependencies, we would prefer an integrated architectural solution. We present, from our ongoing work, a solution that would seem to meet all these needs, through a modified use of naming and addressing. Our proposal is incrementally deployable and existing core network routers & routing protocols need not change.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications; C.2.2 [**Network Protocols**]: Protocol architecture

General Terms

Design

Keywords

Naming, Addressing, Routing, Identifier, Locator, Mobility

1. INTRODUCTION

At present, the Mobile IP standards have limited deployment. While link-layer mobility (e.g. IEEE 802.11 wireless) can be useful for many situations, it can not solve all of the mobility challenges that users face today. The rise of various middleboxes, such as Network Address Translators (NATs) and firewall devices, over the past decade makes Mobile IP even more challenging to deploy.

Existing approaches to mobility have significant operational issues. This section outlines the current approaches to node mobility. For mobility, the IETF has created two different specifications, Mobile IPv4 [11] and Mobile IPv6 [5]. Mobile IPv6 is a direct derivative of Mobile IPv4. Neither of these specifications is widely deployed as yet. Both are complex. Both have numerous operational challenges. Various additions to each have been proposed in numerous Internet-Drafts.

1.1 Mobile IPv4

The basic principle of Mobile IPv4 is that packets from Correspondent Nodes always travel to the Mobile Node's conceptual home address, H, located at the IP network that forms the node's Home Network (HN). Then, if the Mobile Node (MN) is not connected directly, a Home Agent (HA) located on that last-hop IP subnetwork will accept the packets addressed to the Mobile Node and forward them to the Mobile Node's current location, at its Care-of-Address (CoA), using IP-in-IP tunnelling of the original packet. However, packets from the Mobile Node to the Correspondent Node (CN) travel directly, using normal routing, (except when the CN is itself mobile, in which case the return packets travel back to the CN via the Home Agent acting for the CN). This packet forwarding path forms a triangle with vertices at the CN, the HA, and the MN. Each Mobile Node requires at least one trustworthy Home Agent to forward traffic on its behalf. The presence of this triangle routing may increase the latency for packet travel from the Correspondent Node to the Mobile Node. Additionally, the path asymmetry may perturb some protocol behaviour at higher layers, e.g. TCP's "ACK clocking" behaviour for rate control.

Figure 1 shows the network handoff diagram for a network-layer handoff using Mobile IPv4. Since IPv4 does not support stateless auto-configuration, DHCP is included. After movement is detected, there is time period of around 3 round-trip times (RTTs) before data can flow from the correspondent node to the mobile node.

1.2 Mobile IPv6

Mobile IPv4 and Mobile IPv6 are based on the same underlying concepts, but the implementation details are somewhat different. First, the similarities with Mobile IPv4 will be discussed, and then the differences.

With Mobile IPv6, each Mobile Node (MN) has a permanent IPv6 address, which is called its *Home Address*. This is used as a stable identifier for the Mobile Node. For example, TCP session state is bound to the Mobile Node's Home Address. So, regardless of where the Mobile Node is connected to the network, transport-layer protocols (e.g. TCP, UDP) and application protocols name the node using its Home Address. As with Mobile IP, a Correspondent Node that wishes to communicate with the Mobile Node will send packets to the Mobile Node's Home Address. Then, a *Home Agent (HA)* located on the same IP subnetwork as that Home Ad-

MobiArch '07, August 27–31, 2007, Kyoto, Japan.
Copyright 2007 ACM 978-1-59593-784-1 ...$5.00.

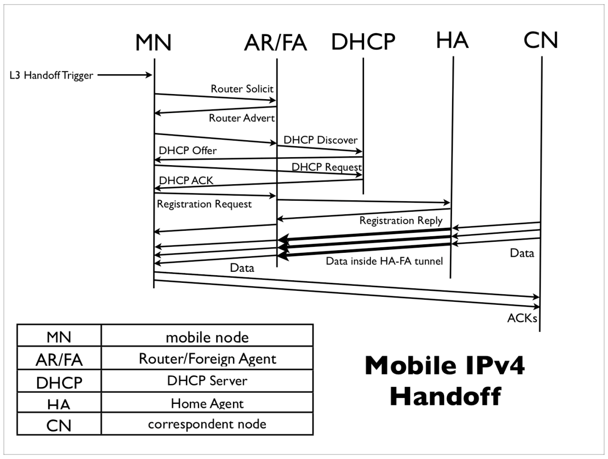

Figure 1: Mobile IPv4 handoff time-sequence diagram

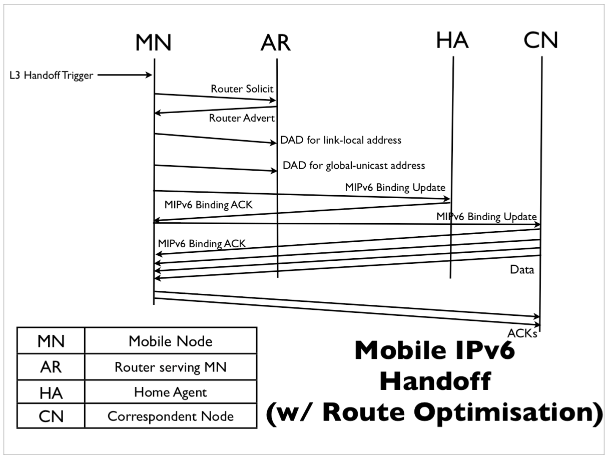

Figure 2: Mobile IPv6 handoff time-sequence diagram

dress will forward traffic to the Mobile Node at its current location. The Mobile Node's current location is indicated by its *Care of Address*, which is used as a locator. Traffic forwarded between the Home Agent and the Care of Address is sent via an IP-in-IP tunnel. The Home Agent also responds to IPv6 Neighbor Discovery protocol messages, including Duplicate Address Detection (DAD), that are intended for the Mobile Node and are present on the Home Address's subnetwork whenever the MN is absent. Duplicate Address Detection significantly slows the network-layer handoff time, which has caused the IETF to explore ways to optimise DAD.[14, 7, 8] Mobile IPv6 introduces a new Mobility Header which is used to carry various mobility-related control messages between the Mobile Node and the Home Agent. These control messages permit the Mobile Node to inform the Home Agent of any changes to its current location, including when the Mobile Node comes home to its Home Address.

Unlike Mobile IPv4, packets from the Mobile Node are tunnelled back to the Home Agent, decapsulated from the tunnel by the Home Agent, and then are forwarded along to the ultimate destination. This IPv6 tunnelling incurs a fixed 40 byte overhead per packet tunnelled. So the "triangle routing" issue of Mobile IPv4 does not exist in the same form with Mobile IPv6. This difference helps ensure that traffic from the Mobile Node will not be dropped due to ingress IP address filtering.[3] Unfortunately, this tunnelling is computationally expensive, increases latency, and causes packet fragmentation.

In order to eliminate some of this tunnelling and also to generally reduce packet latency, Mobile IPv6 has an optional mechanism to provide *Route Optimisation*. With this mechanism, the Mobile Node informs the Correspondent Node of its actual location within the network by exchanging *binding update (BU)* messages. This optimisation reduces the chance that packets will need to be fragmented, and generally reduces the round-trip time, but the additional overhead of the Home Address Option or Routing Header means that some packets will still need to be fragmented prior to transmission and reassembled upon receipt.

Figure 2 shows the packet time-sequence diagram for a network-layer handoff using Mobile IPv6. Since IPv6 includes stateless auto-configuration, DHCP is omitted. After movement is detected, there are 3 round-trips plus Duplicate Address Detection (DAD) delays for 2 different addresses (link-local and global unicast) required before data can flow from the correspondent node to the mobile node.

2. MOBILITY THROUGH NAMING

This section discusses our claims that ILNP could be used to provide a suitable solution for the key problem space of enabling mobile hosts and mobile networks.

Instead of proposing new additions or changes to the existing Mobile IP protocols, we seek a new approach to mobility based on the principle of separating the names for Identity from Location. At present, the deployed Internet uses the same name, an IP address, sometimes to *identify* end nodes (e.g. by TCP or UDP), and other times to *locate* an end node (e.g. by routing protocols). Various issues arise from this semantic overloading. We seek to solve them by breaking the IP address into two separate parts. The first is called an *Identifier* because it is only used as an end node identifier (e.g. with TCP or UDP). The second is called a *Locator* because it is only used by the network-layer to send packets to the subnetwork where the node is located. While this can be undertaken with either IPv4 or IPv6, the exposition is simpler if one considers IPv6.

2.1 Separating Location from Identity

The Identifier-Locator Network Protocol (ILNP) [1] proposes to split the IP Address into two distinct components. The first component is the Locator. A Locator names a single subnetwork and is topologically significant. The second component is the Identifier. An Identifier names a single node and is not topologically significant. This split is believed to enable an improved network architecture, particularly with respect to mobility and multi-homing. In the new architecture, the set of Identifiers used by a node can be very long-lived, but the set of Locators could be very short-lived. As a node moves from one point of network attachment to another, the Identifier(s) typically are constant, but the Locator(s) change with each move to a different subnetwork. With ILNP, upper-layer protocols (e.g. TCP, FTP) include the Identifier, I, in their session state, but never include the Locator in their session state. We note that separating Location from Identity to support mobility is not new; certainly the concept has been proposed in NIMROD, in the GSE proposal for IPv6, and in HIP.[2][12][10][9]

We call our approach ILNPv6 as it is derived from IPv6. We use 128-bit addresses as in IPv6, but use the high-order 64 bits of the address (also called the Routing Prefix) as our Locator, L. A Locator names a sub-network, not an end node. The low-order 64-bits of the IPv6 address become our Identifier, I. Identifier values are extremely likely to be globally unique, because of the way that they are created, and always must be unique within the scope of a given Locator. By default, Identifier values are in IEEE EUI-64 format and are formed from one or more IEEE MAC address(es) associated with the node[1]. When created in this way, Identifiers are normally globally unique.[2] However, ILNP only requires a very high probability that the Identifier is globally unique.[1] If there were two nodes on the global Internet that happened to accidentally be using the same Identifier, another node could communicate with both of those nodes sequentially, but could not communicate with both at the same time. Note that the Identifier does not name an interface and is not tied to the interface from which it takes its value; a MAC address simply provides a convenient way to create an Identifier very likely to be unique within the scope of a given Locator.

2.2 ILNPv6

ILNPv6 can be implemented as a set of backwards-compatible extensions to IPv6. ILNPv6 uses the same packet header format as IPv6, except that each 128-bit IPv6 Address field is split into separate 64-bit Locator and 64-bit Identifier fields. So the Source IPv6 Address field is split into a Source Locator field and Source Identifier field, while the Destination IPv6 Address field is split into a Destination Locator field and a Destination Identifier field. With ILNP, the Identifier field is always formed from an IEEE MAC address associated with the node. This might be any MAC address that belongs to the node, because the Identifier names a node rather than naming an interface of a node.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |   Next Hdr    |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                        Source Locator                         +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                      Source Identifier                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                      Destination Locator                      +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                    Destination Identifier                     +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: ILNPv6 packet header

With ILNPv6, upper-layer protocols (e.g. UDP, TCP) are modified to only use the 64-bit Identifier values. For example, the TCP pseudo-header checksum for ILNPv6 only uses the 64-bit Identifiers, never the 64-bit Locators. The Locator values are only used by the network-layer and only for forwarding packets between source and destination. This decoupling means that changes to Locator values are invisible to the transport-layer and other upper protocol layers. In turn, this facilitates a simpler approach to mobility and multi-homing. This also means, as a side effect, that NAT, which would only affect the Locators, has no impact on the transport-layer or other upper-layer protocols when ILNP is in use.

IPv6 Neighbor Discovery is used unchanged in ILNPv6, so last-hop routers need not change. Also, since the Locator is effectively the IPv6 routing prefix, it is clear that core routers and routing protocols need not change. A mobile node may discover its Locator value much the same way it might discover its IP routing prefix today, through Router Advertisements and Router Solicitations.

Also, we propose that a new, more abstract, networking API be provided. Unlike the current BSD Sockets paradigm, this new interface does not use addresses or the sockaddr data structure. The new API uses domain-names instead of addresses (and service-names instead of port numbers). While current applications have to call separately into the DNS Resolver library to translate DNS information into numeric values that can be placed into a BSD sockaddr data structure to be given to the BSD Sockets interface, in the future, applications can just provide the domain name information to create, open, or respond to a network session. In this, we follow the example of the Java programming language, which already offers a more abstracted networking API option to application writers. We believe that this new API will prove easier to use, making it easier and faster to create new networking applications. The new API, by providing data hiding of lower protocol layer details, encourages application authors not to delve into the lower protocol layer details when designing an application layer protocol.

2.3 Domain Name System Usage with ILNP

ILNP requires the creation of two new Domain Name System (DNS) Resource Records, an I record and an L record. The I record is used to hold the Identifier(s) associated with a domain-name, while the L record is used to hold the Locator(s) associated with that same domain-name. Normally, if one requests either the I or L records for a given domain-name, then all I and all L records associated with that domain-name are returned.

As an optimisation, three other DNS resource records are also added. The PTRL and PTRI records are used together to perform a reverse lookup. When one performs a PTRL lookup on a given Locator value, the fully-qualified domain name of the authoritative DNS server for that subnetwork is returned. In turn, if one then sends a PTRI lookup request with some Identifier value to that authoritative DNS server, then the fully-qualified domain name (FQDN) of the node on that subnetwork with that Identifier is returned. (The LP record is discussed in Section 2.5 below.)

With the currently deployed IPv4 Internet, scalability of DNS depends upon the ability of lower-level (i.e. closer to the end user) DNS resolvers to cache (1) the NS records used to indicate zone delegation [3] and (2) also the A records of the upper-level (i.e. closer to the root) DNS servers associated with those NS records. In turn, this depends upon that small set of NS and A records having moderately long lifetimes. Many top-level DNS servers, for example F-Root, use BGP anycasting with replicated servers, rather than BGP site-multihoming.[4] The BGP anycast techniques work unchanged with ILNP, so upper-level DNS servers can continue to have moderately long lifetimes for their A (or AAAA or L & I) records. Also, DNS resolvers will continue to cache learned DNS resource records for the configured lifetimes of each learned DNS resource record. So the deployed DNS will continue to scale as well with ILNP as it does for the current IPv4 Internet.

In practice, the currently deployed IPv4 Internet is heavily dependent upon the proper operation of the Domain Name System.

[1] Identifier values can be formed in other ways but we defer that discussion because of lack of space.

[2] The exception here would be if a single MAC address were accidentally burned into two different systems due to manufacturing error.

[3] For example, the delegation by a root server of `.COM` to some set of authoritative DNS servers.

Internet Service Providers have understood for more than a decade that users are unable to distinguish *Network not working* from *DNS not working*. For example, while in theory one might use a raw IPv4 (or IPv6) address to connect to some web server, it is very likely that the web page referenced will in turn reference domain names embedded in URLs for external objects required to load that single web page properly in one's web browser. As a second example, many deployed network Access Control Lists (ACLs) require that DNS reverse (i.e. PTR) lookups of the originator's IP address resolve properly to a fully-qualified domain name. Such DNS resolution will fail, of course, if the DNS itself is down. In turn, this will cause the ACL rule to drop packets from the originator. In the particular case of IPv6, users have found it awkward to use raw IPv6 addresses because of the increased address size (compared with IPv4) and hexadecimal format. For ILNP servers or clients that are relatively stationary, one could hand-configure the L and I values for that node into one's application and ILNP would continue to work. Hence, the Internet's dependence upon proper operation of the DNS continues to exist with ILNP, but is neither new nor made worse by ILNP in practice.

Similarly, with the currently deployed IPv4 Internet, even if one has an IP address for a given node, attempts to use that IP address to reach the node might not succeed in all cases. There are a wide range of faults that might cause such connection attempts to fail. In the simplest case, one might have the wrong IP address for that node.[4] Alternately, the node itself might be down. Also, the node might be a laptop computer that is up but not connected to the global Internet at that moment, or there might be a routing fault somewhere along the path from the originator to the destination. With the current Internet, it is often impossible to know which of these causes is preventing the connection attempt from succeeding. So the inability to know which fault type applies to a given connection failure is neither new nor made worse by ILNP.

2.4 Mobile Hosts with ILNP

With ILNP, mobility support is a native property of the network protocol, rather than an add-on protocol. In fact, with ILNP, mobility and multi-homing are supported by a common set of mechanisms. When a mobile node changes its location, its Locator will change. At that point, the mobile node sends ICMP control messages – Locator Update (LU) messages – to all existing correspondents informing of the node's new Locator(s). These LU messages are authenticated to prevent forgery attacks, either using a lightweight non-cryptographic method that prevents off-path attacks or using more comprehensive cryptographic authentication. Additionally, the mobile node updates the set of L records in its DNS entry by using Secure Dynamic DNS Update.[13] If the direct ICMP messages are not delivered to an existing correspondent for any reason, then that correspondent can learn the updated Locator(s) by making a DNS query. New correspondents will discover the current Locator(s) through the DNS as part of the normal session initiation process. So with ILNP there is no routing table impact due to mobility and we eliminate the protocol complexity of the current Mobile IP techniques.

Neither a Home Agent nor a Foreign Agent is needed for ILNP, unlike both Mobile IPv4 and Mobile IPv6, since ILNP nodes support IETF Secure Dynamic DNS Update. Secure Dynamic DNS Update does require a packet exchange, but this packet exchange need not be initiated or completed before the Mobile Node updates its existing correspondents with the Mobile Node's new location using a Locator Update (analogous to the IPv6 binding update). Microsoft Windows XP clients and servers support Secure Dynamic DNS Update, so it is widely available.[6] Further, Secure Dynamic DNS Update is useful in the current Internet and already is deployed in some places. Another potential issue with ILNP is its reliance on DNS Security to authenticate domain-name, Identifier, and Locator mappings. However, ILNP uses the existing DNS Security mechanism designed for the IP Internet. The deployed IP Internet has significant vulnerabilities if DNS Security is not in use, so we need DNS Security for the existing IP Internet, i.e. independent of the question of use within ILNP.

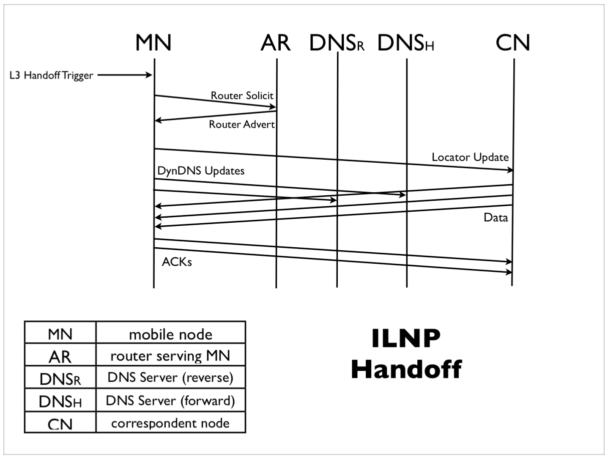

Figure 4: ILNP mobility handoff time-sequence diagram

Duplicate Address Detection (DAD) is not required for ILNP because the Identifier is formed from an IEEE identifier already present within the node, unlike Mobile IPv6. The IEEE EUI-64 value is very probably globally unique. However, link-layer communications will fail first should more than one node on the same link try to use the same MAC address. So ILNP does not need to consider that case. The absence of DAD reduces the network-layer handoff latency.

ILNP never requires packet tunneling and always uses optimal routing through the normal routing mechanisms, unlike both Mobile IPv4 and Mobile IPv6. The elimination of tunnelling might significantly improve performance, both due to the optimised routing of packets and the absence of packet fragmentation/reassembly due to tunnelling overhead.

Figure 3 shows the packet time-sequence diagram for a network-layer handoff using ILNP. Since ILNP supports stateless auto-configuration, DHCP is omitted. After movement is detected, only 1 RTT and 1 Locator Update are required before data can flow from the correspondent node to the mobile node. ILNP can provide much lower network-layer handoff latency than either version of Mobile IP.

2.5 Mobile Networks and Multi-homing

An increasing consideration with IP is that of multi-homing (independent of Mobile IP). Additionally, there is a growing interest in mobile networks, i.e. from the IETF NEMO WG charter[5]:

"The NEMO Working Group is concerned with managing the mobility of an entire network, which changes, as a unit, its point of

[4] For example, because DHCP with dynamic IP addressing is in use and the node's IP address varies temporally or because of renumbering or because of an updating fault when using the existing IETF standard Secure Dynamic DNS Update procedure.

[5] http://www.ietf.org/html.charters/nemo-charter.html

attachment to the Internet and thus its reachability in the topology. The mobile network includes one or more mobile routers (MRs) which connect it to the global Internet.
A mobile network is assumed to be a leaf network, i.e. it will not carry transit traffic. However, it could be multihomed, either with a single MR that has multiple attachments to the internet, or by using multiple MRs that attach the mobile network to the Internet."

With ILNPv6, multi-homing, node mobility and network mobility are essentially handled by the same mechanism: through the change in the value of the Locator, L. When a mobile node moves to another IP sub-network, it will change its value of L, discovering a suitable value locally from Router Advertisements. An ILNP node may hold and use more than one value of L concurrently if it is multi-homed, whether through a single router that happens to be multi-homed, or through multiple routers, each offering a different value for L. With ILNP, a mobile network can be seen as a special case of multi-homing: values of L can be changed as site connectivity changes.

ILNP adds an additional DNS resource record to enable simple and scalable mobile networks. We introduce the use of a fully qualified domain name (FQDN) to name a mobile network, introducing an extra level of indirection in naming a node. A node that is connected to a mobile network may use a *LP* (Locator Pointer) record in the place of an L record. Where a L record would provide a 64-bit Locator associated with the node, a LP record provides the FQDN of the mobile network that the node is connected to. So if one does an L record lookup on a node's domain-name, the I records, L records (if any), and LP records are all returned. The correspondent then performs an L record lookup in the FQDN found in the LP record to learn the actual numeric Locator value. The LP record is a performance optimisation; one could use individual L records, at the cost of numerous DNS updates being required when a mobile network moves.

Of course, whilst we have described this mechanism for use in mobile networks, it is also useful for fixed networks using ILNP. In all cases, it acts to reduce the volume of the data in a DNS server for a site and also to improve manageability of the DNS data.

3. OTHER BENEFITS

With our discussion above, ILNP could be considered *yet another IP mobility proposal*. So, in this section, we try to address our claims that ILNPv6 would have additional benefits in the network architecture. Note that these benefits are useful in general, and not just for mobile IP networks. It is also important to note that these are all native capabilities that naturally arise from the enhanced naming scheme, rather than being bolted-on with special-purpose mechanisms.

3.1 NAT For Free

As we have noted above, with the use of IP addresses today, as the Transport protocol state is tied to the whole of the IP address, operation of localised addressing using Network Address Translation (NAT) causes problems for end-to-end communication. The NAT box must perform a large amount of work per packet, as well as hold some application state in some cases. With ILNPv6, as only the Identifier is used in Transport session state, and is not used for routing, the Locator value can be changed by the NAT as required, without affecting the end-to-end connection state. So NAT and mobile nodes, or even mobile networks, can work together easily using ILNP.

3.2 IP Security Without the Fuss

IETF IP Security always needed a node identifier that was not an IP address, but no such namespace exists in the current architecture. So, IP Security Associations (SAs) were bound to IP Addresses instead. With ILNP, IPsec Security Associations (SAs) are bound to the Identifier values only, not to the Locator values. Further, the IP Authentication Header omits the Locator fields in its calculations. These changes do not introduce new attacks. Together this means that AH and ESP now will work through a NAT device without requiring any special "NAT Traversal" support. IPsec SAs retain end-to-end state through use of the Identifier value, and so will work with NAT, mobile nodes, and/or mobile networks. We are continuing to explore the security ramifications of ILNP.

3.3 Mobile and Network Realms

There is growing interest in using IP for providing communication mechanisms between non-IP edge networks. Whilst tunnelling is always possible, and may be the most appropriate and desirable mechanism in some applications, a more lightweight and general approach for communication between different network realms may be desirable, especially for some mobile applications.

We note from our discussion above that while the Locator has a well-defined semantic for the IP network layer, the Identifier is opaque and its value is not important. Similarly, the Transport layer state is not tied to the Locator, only to the Identifier and does not make use of the Locator value. So, it should be possible, in principle, to run TCP and UDP across non-IP protocols at the edges of the network and still allow end-to-end communication across an IP core, given a suitable network layer gateway at the boundary between the IP and non-IP network realms.

For example, there is much interest in use of wireless and mobile sensor/actuator networks and such edge networks that may not use IP. However, an Identifier value could be used in sensor/actuator devices, especially one formed in the IEEE EUI-64 syntax, in order to allow state to be maintained across network realms.

Also, MANETs may benefit for inter-MANET communication, or communication with non-MANET nodes through the use of a naming approach based on ILNP.

3.4 Incremental Deployment

We believe that ILNPv6 can be incrementally deployed. As we have explained above, as the most-significant 64-bits of the ILNP address, the Locator, coincide with the IPv6 routing prefix, the core routers and routing protocols do not have to change.

Of course, end-system networking software will need to change. The network layer operation will be modified to recognise the I:L split in the IP address, and also to keep state for current I:L bindings. Neighbor Discovery should not have to change, however. It should be possible to have mixed concurrent operation of ILNPv6 and IPv6, on a per-session basis, with ILNPv6-enabled nodes.

We believe that a version of ILNP based on IPv4 is also possible. Engineering would not be as elegant as for ILNPv6, using the current IPv4 address as the value for L, and requiring an IP option header to carry the I value. However, given the discussion above on Network Realms, at this point, we believe protocols above ILNP should be able to interoperate through gateways.

The most disruptive change to existing infrastructure is likely to be the increased reliance on DNS. Although we believe that for small scale deployment and testing, a modified hosts file could be used, for operational deployment, the DNS has to be upgraded. Even then, no wholesale upgrade to all DNS servers and libraries is required: only those serving ILNP nodes need the new records, (L, I, PTRI, PTRL, LP), and then only those serving mobile nodes need the DynDNS and DNSsec support.

3.5 No Free Lunch

The main potential issue is felt with the new use of naming. Network interfaces no longer have a globally routable name, as they have with IPv4 and IPv6 addresses. This is likely to affect some network management applications most. For example, this might require changes to SNMP MIBs and SNMP applications.

In some existing applications, notably FTP, the IP address is (mis)used as an application-layer name. NATs already have to deal with this. Such applications will have to be modified. Indeed, applications will be forced into specifying application-specific namespaces, rather than just using a namspace that is derived from the use of IP addresses. We encourage applications to use domain names instead, however (see discussion above on a new API). Of course, some network management applications might still need to use I and L values directly; this is not precluded.

Reliance on the Domain Name System is more explicit with ILNP, which is a natural consequence of the focus on the use of names and dynamic name bindings between objects to implement the mobility and multi-homing functions. However, the potential issues from DNS faults already exist with the deployed IPv4 Internet and existing IETF DNS standards. Apart from the five new record types proposed (i.e. I, L, PTRI, PTRL, LP), the Secure Dynamic DNS Update and DNS Security mechanisms are already defined and are being deployed independently of ILNP: ILNP simply leverages their existing standardisation and growing availability.

4. CONCLUSIONS

Whilst our work is ongoing, and we have not completed a prototype, we take the position that given the right naming architecture, mobility becomes a natural capability of the network. Without requiring special purpose mobility infrastructure that might be hard to implement and deploy, expensive to operate and maintain, and without the need for mobile-specific engineering enhancements, it is possible to offer mobility as a first class function integrated in the network architecture.

In the next 12 months, we intend to build an initial prototype of ILNP in BSD Unix and test it between St Andrews and London using the UK Joint Academic Network (JANET).

5. REFERENCES

[1] R. Atkinson, M. Lad, S. Bhatti, and S. Hailes. A Proposal for Coalition Networking in Dynamic Operational Environments. In *Proceedings of IEEE Military Communications Conference*, Washington, DC, USA, Oct. 2006. IEEE.

[2] I. Castineyra, N. Chiappa, and M. Steenstrup. The Nimrod Routing Architecture. RFC 1992, Internet Engineering Task Force, Aug. 1996.

[3] 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, Jan. 1998.

[4] S. Gibbard. Geographic Implications of DNS Infrastructure. *Internet Protocol Journal*, 10(1):12 – 24, Mar. 2007.

[5] D. B. Johnson, C. E. Perkins, and J. Arkko. Mobility support in IPv6. RFC 3775, Internet Engineering Task Force, June 2004.

[6] C. Liu and P.Albitz. *DNS and BIND, 5th Edition*. O'Reilly and Associates, Sebastopol, CA, USA, May 2006.

[7] J. Macker. Interoperable Networks for Secure Communications, Task 6, Phase 1. Final Report INSC-TASK6, North Atlantic Treaty Organisation (NATO), Dec. 2003.

[8] N. Moore. Optimistic Duplicate Address Detection for IPv6. RFC 4429, Internet Engineering Task Force, Apr. 2006.

[9] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. RFC 4423, Internet Engineering Task Force, May 2006.

[10] M. O'Dell. GSE - An Alternate Addressing Architecture for IPv6. Internet-Draft draft-ipng-gseaddr-00.txt, IETF, Feb. 1997.

[11] C. Perkins. IP Mobility Support for IPv4. RFC 3344, Internet Engineering Task Force, Aug. 2002.

[12] R. Ramanathan. Mobility Support for Nimrod: Challenges and Solution Approaches. RFC 2103, Internet Engineering Task Force, Feb. 1997.

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

[14] J. Weston. Interoperable Networks for Secure Communications, Task 3. Final Report INSC-TASK3, North Atlantic Treaty Organisation (NATO), July 2006.