Evolving the Internet Architecture Through Naming

Randall Atkinson, *Senior Member, IEEE,* Saleem Bhatti, *Member, IEEE,* and Stephen Hailes, *Member, IEEE*

***Abstract*—Challenges face the Internet Architecture in order to scale to a greater number of users while providing a suite of increasingly essential functionality, such as multi-homing, traffic engineering, mobility, localised addressing and end-to-end packet-level security. Such functions have been designed and implemented mainly in isolation and retrofitted to the original Internet architecture. The resulting engineering complexity has caused some to think of 'clean slate' designs for the long-term future. Meanwhile, we take the position that an evolutionary approach is possible for a practical and scaleable interim solution, giving much of the functionality required, being backwards compatible with the currently deployed architecture, with incremental deployment capability, and which can reduce the current routing state overhead for the core network. By enhancing the way we use naming in the Internet Architecture, it is possible to provide a harmonised approach to multi-homing, traffic engineering, mobility, localised addressing and end-to-end packet-level security, including specific improvement to the scalability of inter-domain routing, and have these functions co-exist harmoniously with reduced engineering complexity. A set of proposed enhancements to the current Internet Architecture, based on naming, are described and analysed, both in terms of architectural changes and engineering practicalities.**

***Index Terms*—Architecture, Communication system routing, Internet, Internetworking, Networks**

I. Introduction

INTERNET users wish to use multi-homing and traffic engineering. At present, there is particular concern about their impacts on the scalability of Internet routing [4]. Site multi-homing is increasingly deployed and significantly increases both entropy and size of the inter-domain routing table. Use of more-specific IP routing prefixes to enable traffic engineering is growing and also increases both entropy and size of the inter-domain routing table. Prior work indicates that site multi-homing is the dominant source of recent entropy increases and routing table growth [5]. So there is substantial current interest in alternative networking and routing approaches.

Separately, business mergers/divestitures and non-aggregatable address block assignments tend to increase the entropy and size of the inter-domain routing table. For these last two items, a candidate solution approach is to have improved capability to change the upstream Internet provider for a site without having to renumber all nodes in an end site.

Manuscript received 4 November 2009; revised 14 April 2010. This paper contains revised and updated material from the authors' papers in the *IEEE Military Communications Conference (MILCOM)* of 2008 and 2009. [1]–[3]

R. Atkinson is a consultant.

S. Bhatti is with the University of St Andrews (e-mail: saleem@cs.st-andrews.ac.uk).

S. Hailes is with University College London (UCL).

Digital Object Identifier 10.1109/JSAC.2010.101009.

TABLE I
TERMINOLOGY USED IN THIS PAPER

Term	DNS Record	Definition
Address	AAAA, A	Name used both for locating and identifying a network entity.
Locator	L64	Name that locates, topologically, a subnetwork.
Identifier	ID	Name that identifies a node, within the scope of a given locator, but could be globally unique.

TABLE II
USE OF NAMES IN ILNP AND IP

Protocol layer	ILNPv6	IP (v4 and v6)
Application	FQDN	FQDN, IP address
Transport	Identifier, *ID* (+ port)	IP address (+ port)
Network	Locator, *L*64	IP address
Link	MAC address	MAC address

A. Naming Problems in IP today

In our discussion below, we use the term *name* in a very general sense, to refer to any label that is attached to a network object. A summary is given in Table I.

An IP Address has two different functions – as a *locator* for naming an IP (sub)network, and as an *identifier* for naming a node [6]. The IP Address bits are used in applications, in transport protocols (e.g. within the TCP pseudo-header checksum), and in the network layer to route packets to their destination(s) – see the final column of Table II.

What is not visible in the final column of Table II is that in operational use, an IP address is bound to a specific interface or sub-network point of attachment (SNPA) on a host. So, all IP address bits are used across several different layers, from application layer right down to the physical interface. This means, for example, that a Transport layer communication end-point is bound directly to a SNPA on a host. This use of the IP address entangles the functionality across the layers, hindering the application of the end-to-end argument [7] for certain network functionality. Specifically, the impacts of this current semantic overloading of the address include:

Localised Addressing: Site border use of Network Address Translation (NAT) to enable private addressing introduces a discontinuity in the end-to-end state which must be managed by dynamic state mappings. This makes harmonised use of multi-homing, mobility, traffic engineering, and end-to-end security difficult and complex.

Multi-homing and Traffic Engineering: At present, multi-homing and some traffic engineering mechanisms each require that additional routing state be kept in most or all backbone routers. Since IP routing uses longest-prefix match to select the preferred route to a destination, these functions require additional, more-specific IP routing prefixes to be advertised to all backbone routers, causing the size of the backbone

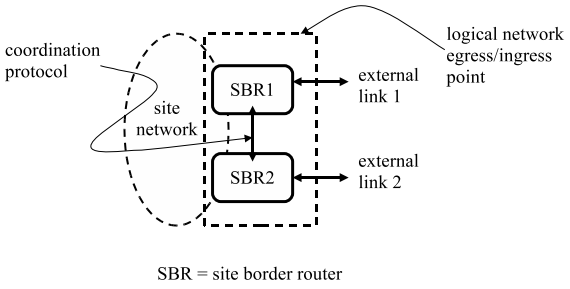

Fig. 1. An example site network, with two site border routers (SBRs), each providing access to an independent external link. The coordination protocol is not discussed in this paper, but existing protocols could be adapted easily.

routing tables to increase geometrically, raising scalability concerns. The concerns are such that the IAB asked the Internet Research Task Force's (IRTF's) Routing Research Group (RRG) to investigate better approaches to these issues [4].

End-to-End Security: At present, IPsec Security Associations [8] include both source and destination IP addresses, and so are tied to a SNPA. This means that if a node moves, or a network moves, then the existing IPsec Security Associations become invalid. This exacerbates existing concerns about the scalability of key management for IPsec devices.

Support for Mobility: Mobile IPv4 (MIPv4), Mobile IPv6 (MIPv6), and Network Mobility (NEMO), require that extra IP addresses, known as Care of Addresses (CoAs), be used with a special-purpose router, known as the Home Agent, using IP-in-IP tunnelling to forward packets sent by a correspondent via the mobile node's Home Address (HoA). The CoA acts as a locator and changes as the host or network moves, and the HoA acts as an identifier. The IETF is currently undertaking the unification of MIP and NEMO (with IKEv2), but will retain the basic approach, using HoAs and CoAs.

B. Network Scenario

In operational networks, care must be taken regarding the physical structure and connectivity of the network, and how this relates to proposed new architectures. Our discussion considers the scenario depicted in Figure 1. Here, a site network has connectivity provided by two site-border router (SBRs). The site network might be multi-homed, might wish to use its two links for traffic-engineering, or perhaps even a mobile network with two radio links.

C. Key Concepts

Our work investigates a network architecture that crisply separates identifiers and locators [9]. We call our approach the *Identifier-Locator Network Protocol (ILNP)*. ILNP is an architecture and can be engineered for use with either IPv4 or IPv6 [10]. While the idea of an *Identifier*/*Locator* split is not new, our particular approach is new and is specified in more detail than preceding proposals [11]–[14]. Due to space limitations, we hereafter focus on ILNP for IPv6 (*ILNPv6*).

At present, the IP Address is used both to indicate location for routing packets and also for identity of networked nodes [15]–[17]. The overloaded semantics of the IP Address create architectural issues and limitations [11], [18], [19]. ILNPv6 is designed to split the Address into a *Locator (L64)*, used only for routing, and an *Identifier (ID)*, used only for node identity. As shown in the second column of Table II, transport protocols (e.g. TCP, UDP) include only the Identifier in the pseudo-header calculation, while the Locator is used only in the network layer and for Neighbour Discovery. As recommended in RFC1958 [20], we believe applications should use fully-qualified domain names (FQDNs), rather than lower-layer names (e.g. IP addresses), as much as possible.

The decoupling of location from identity means that *Locator rewriting* can enable many functions efficiently for whole sites, though it is also possible for each function to be implemented individually by each host. The binding of a ID:L64 pair to a SNPA remains dynamic, through the use of normal IPv6 Neighbour Discovery. With ILNP6, the layers of the protocol stack are disentangled, and end session state remains invariant, as we shall show later.

We are open-minded about the potential of other new namespaces (e.g. in the Application Layer) in evolving the Internet Architecture. However, our focus in this paper is upon the network layer and transport layer.

II. Overview of ILNP

This section presents a brief overview of our proposed enhancements to the Internet Architecture, and also specifically to IPv6. We use the term *Identifier-Locator Network Protocol v6 (ILNPv6)* to refer to our proposal, as it can be engineered as enhancements to IPv6[1] [10], [21].

We replace the concept of the IPv6 *address* with the concepts of an *Identifier (ID)* combined with a *Locator (L64)*. The *Locator* names an IPv6 (sub)network: this is used only in routing, and not by the upper layers (e.g. not by TCP or UDP). The *Identifier* is only used for node identity (e.g. by TCP in the TCP pseudo-header checksum) and is not used by the network layer. However, the L64:ID pair is used for Neighbour Discovery (ND). A summary of the difference between the use of names in IP and the use in ILNP is given in Table II.

A. Identifiers & Locators

The 64-bit Identifier is an *IEEE Extended Unique Identifier (EUI-64)* [22], as shown in Figure 2 and as used for IPv6 addresses [23]. The key difference is that in ILNPv6, the Identifier is not an *interface* identifier, but a *node* identifier. The default ILNPv6 Identifier value will have the scope bit (L/G bit) set to global and, for convenience, will be formed from an IEEE 1394 ("Firewire") Media Access Control (MAC) Address or from an IEEE 802 (LAN) MAC Address [24]–[26] taken from the the host. Alternatively, the EUI-64 specification provides a non-unique number space by having a scope bit that indicates whether a given identifier has local-scope or global-scope. For example, one can form anonymous identifiers for privacy as in RFC4941 [27], or use cryptographically generated identifier values such as the 64-bit CGA values as per RFC3972 [28]. There is also a bit reserved (the U/M bit) to indicate whether an Identifier names

[1]Limited space precludes a full discussion, but more details can be found at http://ilnp.cs.st-andrews.ac.uk/.

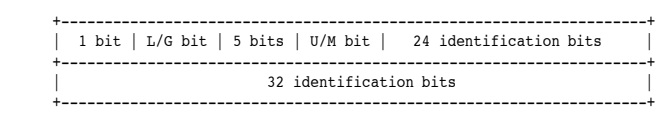

Fig. 2. IEEE EUI-64 Format as used for IPv6 [28] and ILNPv6.

a single node (i.e. unicast) or a set of nodes (i.e. multicast). The Identifier has no topological significance and is treated as an opaque object at the network layer and below.

The Locator is topologically significant and is used to route traffic to a single sub-network containing one or many hosts. The routing system only uses the Locator. For multicast traffic, the Locator indicates the location of a Rendezvous 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. Existing applications that use the BSD Sockets API, but that do not use the IP address bits in applications should continue to work when ILNPv6 is used instead of IPv6.

B. ILNPv6 is an Enhancement of IPv6

The IPv6 packet header and the ILNPv6 packet header are deliberately similar. Essentially, the IPv6 address is broken into two separate components, a Locator (L64) and an Identifier (ID). Significantly, the IPv6 *Interface Identifier* is replaced by an ILNPv6 *Node Identifier* (ID), with slightly different semantics but the same syntax, as shown in Figure 3.

The Locator (L64) is an unsigned 64-bit value carried in the upper portion of the IPv6 address: it is simply a renaming of the IPv6 address prefix used for routing, and retains the same syntax and semantics. The (Node) Identifier (ID) is an unsigned 64-bit value carried in the lower portion of the IPv6 address. The ID value names a *node*, not a *network interface*. An end-system may use multiple ID values and multiple L64 values simultaneously. However, for the duration of a given transport layer session, its ID value should remain constant. This is consistent with the IPv6 Addressing Architecture [23].

Strictly, the ID value needs to be unique only within the scope of the L64 value with which it is used. However, an ID value that is likely to be globally unique (e.g. by use of an EUI-64 value) is extremely useful: for example, it allows us to dispense with IPv6 Duplicate Address Detection (DAD), greatly reducing the network-layer handoff time for a mobile node. An ILNPv6 node discovers its address prefix (L64 value) through the normal IPv6 Router Advertisement mechanisms.

So, ILNPv6 has a header format very similar to the IPv6 header format. Each field is the same size and in the same location as for the IPv6 header, except for the split of each 128-bit IPv6 address into a 64-bit Locator and a 64-bit Identifier.

C. Routing

The ILNPv6 Locator is consistent with the IPv6 Addressing Architecture [23], specifically with section 2.5.4, which states that the sum of bits in the global routing prefix and the subnet identifier is 64 bits. At present, IPv6 address allocation practices provide sites with IPv6 address blocks that are 48-bits long, so there are 16 bits left for internal sub-networks.

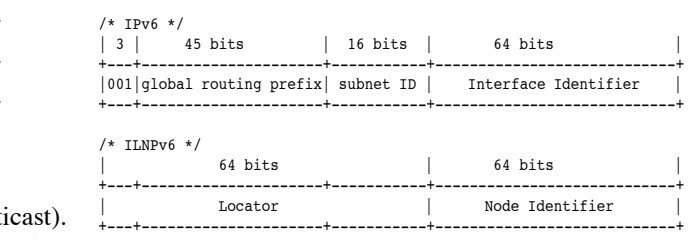

Fig. 3. IPv6 address format (from RFC3587 [29]) as used in ILNPv6

Unicast routing with ILNPv6 is identical to unicast routing with IPv6. The ILNPv6 Locator is equivalent to the full 64-bit IPv6 routing prefix. This means that no changes to deployed IPv6 routers are required before one begins deploying ILNPv6 in the Internet. Similarly, Neighbour Discovery uses the 128-bit combination of the Locator and the Identifier, so ND also does not need to be changed. However, when the Identifier is formed from an IEEE MAC address, Duplicate Address Detection (DAD) is not required, which enables a performance optimisation. *In short, already deployed IPv6 routers will support ILNPv6 without any changes.*

ILNP multicasting is conceptually simple. The Locator field contains the value of a *Rendezvous Point* (i.e. Locator naming a subnetwork with a router joined to the multicast tree) for the destination multicast group. If the originating node does not know of any Rendezvous Point, the Locator field is set to zero by the originating node and can be filled in by an ILNPv6 router. The Identifier field, which has the EUI-64 multicast bit set, names the specific multicast group. While ILNPv6 multicasting is slightly different from IPv6 multicast, backwards compatibility can be maintained by examining the high-order octet of the Locator field. If those bits are all 1, then the packet is an IPv6 multicast packet [23].

D. End-system Session State Invariance

Denoting the TCP, IP, and ILNP session state with the *tagged tuple* notation below, let us consider a TCP connection, with the end-system state represented as the tuples:

$$\langle TCP : a, p, b, q \rangle \tag{1}$$

$$\langle IP : w, x \rangle \tag{2}$$

$$\langle ILNP : y, z \rangle \tag{3}$$

where a and b are, respectively, the local and remote node names, and p and q are, respectively, the local and remote TCP port numbers. If the TCP tuple (1), which is the end-system state, remains *invariant* during the operation of functions such as multi-homing, private addressing, mobility, traffic engineering and end-to-end security, then it is clear that the end-to-end protocol is not affected and the operation of those functions are invisible to the transport layer and the application.

For IPv6 or IPv4, in tuple (1) a and b are, respectively, the local IP address and remote IP address. Further, for tuple (2) w is equal to a, and x is equal to b. So changes to IP addresses in use cause end-system state to vary.

For ILNPv6, in tuple (1), a and b are the local Identifier and remote Identifier, respectively, while in tuple (3) y and z are, respectively, the local and remote Locator values. So, changes in Locator values do not affect the end-to-end protocol state

at the transport layer or above. An internal cache at the top of the network-layer within an ILNPv6 implementation will track current ID:L64 bindings for existing ILNPv6 sessions.

We will use the network scenario of Figure 1 with a TCP example to show how the functions of private addressing, multi-homing, mobility, end-to-end security, and traffic engineering can be provided in an integrated fashion while maintaining this session state invariance.

E. DNS Enhancements

ILNPv6 needs several new DNS resource records. The *ID* resource record contains the unsigned 64-bit Identifier associated with a domain name. The *L64* record contains an unsigned 64-bit Locator associated with a domain name. A given domain name might have multiple *ID* and multiple *L64* records. The combination of a given *L64* record and an associated *ID* record is equivalent to the current IPv6 address. Reverse lookups can be done as with IPv6 today.

The IETF Secure Dynamic DNS Update standard [30] permits a mobile node or multi-homed node to update its *L64* records when the node moves or its upstream connectivity changes (e.g. due to a link fault). Separately, the DNS enhancements for ILNPv6 do not change the fundamental operation of the Domain Name System (DNS). So the DNS Security (DNSsec) standards [31] can be used unchanged to authenticate these new DNS records, and our proposed enhancements do not create new security risks. Both of these DNS standards reportedly are both interoperable and available today in widely used operating system software (e.g. Windows, MacOS X, Linux, BSD) [32].

Finally, multiple L64 values may be used simultaneously to enable multi-homing, and these would be visible within the DNS. The use of preference values in the *L64* DNS record would indicate the receiving node's preference about which Locator a correspondent should use.

F. IP Security Enhancements

At present IPsec Security Associations (SAs) are bound to full IP addresses at the local and remote sites, a and b, respectively, as a form of end-system identity [8]. So, for tuple (2), IPsec requires that the IPv6 addresses at each end-point of the communication remain fixed. This requirement is not always met today when private addressing (e.g. NAT), multi-homing, or mobility are used with IP. So the IP Security specifications have had to be extended, retrospectively, with special-purpose modifications (e.g. for NAT traversal).

With ILNPv6, however, IPsec SAs are bound only to the Identifier (a and b, in tuple (1)), never to the Locator. This makes it easy for the IPsec Security Association – and the related secure communications channel – to remain intact and operational even if the end-points move.

G. Locator Re-writing and Private Addressing

To support private addressing, IP provides three well-known IP networks in a process known as Network Address Translation (NAT) [33]. NAT functions reside at the site border router (SBR) of the privately addressed network and re-write addresses and checksums at the IP and TCP layer, translating between the privately used (local) address, A_L, and the globally unique (routable) address, A_G, for that site, and port numbers may also be re-written so that A_G can be shared amongst many nodes in the private network.

Let us consider a TCP connection with the end-system state at the private network as:

$$\langle TCP : A_L, P_L, A_R, P_R \rangle \langle IP : A_L, A_R \rangle \tag{4}$$

where A_L is the local IP address, P_L is the local port number, A_R is the remote IP address and P_R is the remote port number. However, after traversing a NAT, the TCP state at the remote node (correspondent) will be:

$$\langle TCP : A_G, P_G, A_R, P_R \rangle \langle IP : A_G, A_R \rangle \tag{5}$$

where A_G and P_G are, respectively, the address and port number written by the NAT function: the end-system state is different at each end of the connection and the NAT holds the mapping. This can be disruptive to many applications and functions, such as IPsec and mobility.

With ILNPv6, the end-system state is bound only to the Identifier, and only the Locator is used for routing. So, ILNP end-system state of any TCP connection would be:

$$\langle TCP : I_L, P_L, I_R, P_R \rangle \langle ILNP : L_L, L_R \rangle \tag{6}$$

where I_L and I_R are, respectively, the local and remote Identifier values. An ILNPv6 NAT would re-write only Locator values between, say, L_L, the local (private) Locator value, and L_G, the globally unique Locator value, which are only seen at the network layer packet. So, an ILNPv6 NAT is transparent (invisible) to the end-system connections. For example, if L_L is a local (private) Locator value for our end site, L_G is the global Locator value for our end site, and L_R is the remote Locator value, the TCP packet ingress to the ILNPv6 NAT would be as in tuple (6), and after traversing the ILNPv6 NAT would egress as in tuple (7). As the local Locator value, L_L is simply an IPv6 prefix, it can be easily generated following the IETF recommendations for local IPv6 prefixes [34].

$$\langle TCP : I_L, P_L, I_R, P_R \rangle \langle ILNP : L_R, L_G \rangle \tag{7}$$

This maintains the invariance requirement (Section II-D).

While we have examined ILNPv6 Locator re-writing in the context of NATs, ILNPv6 can use Locator re-writing to support site multi-homing [1], traffic engineering [2], mobile hosts [21] and networks [3], in harmony with other functions [10], efficiently, through a (several) SBR(s).

H. Related Work

Nimrod is a well-known earlier effort to develop a new routing architecture [12]. While Nimrod considered an identifier/locator split, it has never been deployed.

ILNPv6 is derived in part from, and was strongly influenced by, Mike O'Dell's *8+8* network architecture, which also was known as *GSE* [13], [14]. ILNPv6 is not identical to GSE, and has somewhat different properties. In particular, ILNPv6 has

fully addressed security issues and has particularly focused on providing scalable mobility capabilities. ILNPv6 was also influenced by the first author's discussions within the IRTF Name Space Research Group (NSRG).

Of course, the publication of the IAB report and the subsequent re-chartering of the IRTF Routing RG has led to several other proposals for evolving the Internet Architecture. Among the other leading proposals within the IRTF Routing RG are *Six/One*, *LISP*, and *APT*. The *Six/One* proposal is perhaps most similar to ILNPv6, using site border routers to rewrite IP addresses so that site multi-homing will not require more-specific IP routing prefixes [35].

Cisco Systems has proposed *LISP* to the IRTF Routing RG. *LISP* uses scoped addressing to decouple the site-interior provider-independent address, called an EID, from the global-scope inter-domain provider-aggregatable address, which is called an RLOC [36]. The *APT* proposal is somewhat similar to *LISP* in that it separates end-site addressing from inter-domain addressing [37], [38]. Some use the term *Map & Encapsulate* to refer to the class of architectures that include both *LISP* and *APT*.

There has been much other recent research into network architectures, both evolutionary and 'clean slate'. The NewArch Project developed the *FARA* architecture, which is perhaps the best known recent 'clean slate' architecture [39]. Space precludes listing every proposal.

The Host Identity Protocol (HIP) has its own separate IRTF HIP RG, and is another example of an experimental evolutionary architecture [40]. HIP pays great attention to security, requiring cryptographic protection on all packets, while in ILNPv6 we have chosen to maintain the existing ready availability of cryptographic services, while letting individual end users or sites choose when and where to use those services.

The *SHIM6* protocol extensions to support IPv6 multi-homing have recently been standardised by the IETF [41]. *SHIM6* provides a host with more than one IPv6 address at the same time. With *SHIM6*, one IPv6 address is used for transport-layer sessions and is used for identity, while a different IPv6 address can be used for routing packets to the node. So SHIM6 provides a form of identifier/locator split. It is too early to know whether SHIM6 will be widely implemented or widely deployed.

III. Multi-homing

There are two kinds of multi-homing: *site multi-homing* and *host multi-homing*. ILNP can support both, in harmony, but as the latter is not widely used, we discuss only the former. Although host multi-homing is not widely used in IPv6, ILNPv6 hosts can use host multi-homing directly, by using two (or more) L64 values simultaneously with the same ID value. Appropriate selection and routing of the L64 values used may have direct benefits for transport protocols that wish to use multi-path transport layer sessions. Site multi-homing is when a given site has multiple upstream connections to different service providers. This, in combination with BGP and IP routing, can provide greater resilience and availability to all of the nodes within that site.

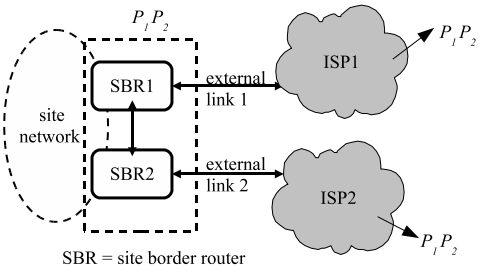

Fig. 4. General multi-homing scenario: our example site network, with two site border routers (SBRs), each provisioned through a separate ISP. In this example, the site network has two routing prefixes, P_1 and P_2, that must each be advertised through each ISP.

A. ILNPv6 Site Multi-homing

Today, site multi-homing is implemented by advertising the site's more-specific IP routing prefix to the entire Internet and relying on the Internet's normal longest-prefix-match route selection algorithm. Unfortunately, this requires that IP routing prefixes to be de-aggregated. So instead of an ISP advertising a single IP routing prefix that covers all of its customers, there are additional more-specific prefixes for each multi-homed site using that ISP.

A common scenario for multi-homing is one that is described in Figure 4. In this example, the site's network is provisioned for external connectivity via two Internet Service Providers, ISP1 and ISP2. The simplest case is that a single routing prefix, P_1, for the site network should be advertised through both ISP1 and ISP2. Even if this prefix is taken from the normal (aggregateable) address allocation of one of the ISPs, it must be advertised separately once it is used for multi-homing. In this example, the site network has elected to use a two routing prefixes P_1 and P_2, and so both prefixes need to be advertised through each of ISP1 and ISP2.

So, the amount of *additional* state introduced into the global routing tables for each multi-homed site is $O(N_I.N_P)$, where N_I is the number of upstream providers, and N_P is the number of prefixes used by the site. This practice is the largest source of entropy in the global routing table today [5], and routing scalability has become a major concern, in large measure due to the current geometric growth in routing entropy [4].

ILNPv6 uses the same mechanism to provide both site multi-homing and host multi-homing. With ILNPv6, the new DNS *L64* or *LP* records advertise the current reachability for a node or site. New correspondents perform a DNS lookup, as at present, to determine how to send packets initially to the target node(s). Whenever a node's currently valid Locator(s) change, the node sends *ICMP Locator Update (LU)* control messages to its existing correspondents. These messages can be authenticated either cryptographically or non-cryptographically, as appropriate for the node's threat environment. The correspondent receives this update, validates it, and then begins using the new Locator(s) to send packets to the original node.

Using Figure 4, consider an IPv6 site network using two routing prefixes, P_1 and P_2. Often, sites prefer the prefixes P_1 and P_2 to be provider independent, as the address prefixes are considered as part of the site's *identity* as well as providing routing information. Each SBR has to advertise both P_1 and P_2

on *both* links, i.e. four *additional* routing entries to advertise.

We can use Locator values L_1 and L_2, respectively on external link 1 and external link 2, taken simply from the upstream provider's Locator space and need not be Provider Independent: the site maintains fixed identity by using Identifier values. As the Locator values are not part of the transport protocol state, we can use both Locator values simultaneously. *So, no additional prefixes need to be advertised.*

With the use of Localised Addressing (Section II-G), using tuple (7) as the TCP packet state from our site network, then packets egressing SBR1 will have the state given in tuple (8) and packets egressing SBR2 will have state as in tuple (9).

$$\langle TCP : I_L, P_L, I_R, P_R \rangle \langle ILNP : L_1, L_R \rangle \quad (8)$$

$$\langle TCP : I_L, P_L, I_R, P_R \rangle \langle ILNP : L_2, L_R \rangle \quad (9)$$

Note that the TCP state at each end of the connection remains the same – the invariance requirement (Section II-D) is maintained. So, ILNPv6 can provide transparent multi-homing to the site-network. Whilst ILNPv6 does not need to use the SBR Locator re-writing to support site multi-homing (each ILNPv6 host could manage its own Locator values), it is a convenient engineering optimisation.

For every multi-homed IP routing prefix converted for use with ILNPv6 multi-homing, the multi-homed site incurs the cost of adding two new DNS entries (an *L64* & an *LP* record) to its DNS servers, plus an *LP* record for each host in the site: the benefit of removing a prefix from the routing tables of thousands of routers accrues globally. So ILNPv6 moves the cost of multi-homing out of the global backbone and localises it within the multi-homed site.

IV. Backwards Compatibility & Incremental Deployment

ILNPv6 is both backwards compatible with IPv6 and incrementally deployable. Ideally, a new version number would be used for ILNPv6 in the IP header. However, as an engineering convenience, to aid deployment, we chose to enhance IPv6, keeping the the ILNPv6 header format nearly identical to the IPv6 header format, so no changes to IPv6 routers or deployed IPv6 backbones is required to deploy ILNPv6. We now describe the mechanisms supporting these properties, and some possible deployment scenarios.

A. Detecting ILNPv6 Capability

An ILNPv6 capable node will have the new ILNPv6-related DNS resource records (e.g. *ID*, *L64*, *LP*) present for its Fully-Qualified Domain Name (FQDN), and could also have (multiple) IPv6-specific *AAAA* record(s). A node initiating a new session, using DNS to discover how to contact a particular correspondent, will either see those new ILNPv6-specific records in the DNS Reply or not. If the initiating node does see ILNPv6-specific records for the correspondent, present as *Additional Data* in the response to an ÂAAA DNS Query, then the initiating node will attempt to use ILNPv6 to create the new session. This means that the initial packet(s) of the session will include a new *ILNPv6 Nonce Option*. If a reply is expected (e.g. TCP is in use or bi-directional UDP), and no reply is received, then the initiating node can fall back to using ordinary IPv6 instead after a suitable time period.

When ILNPv6 is in use, the initial packet(s) of each session will contain the IPv6 Nonce Option. That option will never appear in a packet when IPv6 is in use for the session. If the responder is ILNPv6-capable, it will respond and include its own IPv6 Nonce Destination Option (with its own nonce value) in its reply. If the responder is not ILNPv6-capable, then it will drop the packet because the received Nonce Option is unrecognised.

B. Example Deployment Scenarios

Military networking environments require resilience, high availability, security, and usually require support for both mobile networks and mobile nodes. For security reasons, many military network deployments are either closed or have limited external connectivity. Military networks generally have more centralised equipment selection and configuration control than residential or commercial networks. So a military network might well be an ILNP early adopter.

Other forms of wireless and mobile networking (e.g. smart phones) might find ILNPv6's benefits especially beneficial. At least two emerging smart phone handset manufacturers are also major content providers [42]. Such a vendor could upgrade both the handset and its own content servers simultaneously, gaining nearly immediate user-visible improvements for users accessing its content.

V. Conclusion

We have presented an integrated set of extensions to the current Internet Architecture that provide scalable support for multi-homing, mobility, and end-site traffic management that can greatly improve the scalability of the Internet. Our approach moves as much state as possible out of the routers in the network core into only the specific end-systems that benefit from the functionality that they require. This better aligns costs and benefits to the users needs. It also tends to restore the Internet towards its original model of end-to-end services, a model which has scaled very effectively over the past twenty years. In designing our extensions, we have been carefully to fully address security issues. Our proposed approach is no less secure than the current IP Internet, and in some areas provides security improvements.

By replacing the *IP address* with the *Identifier* and *Locator*, each with crisp semantics, we propose to make naming an architectural tool in enabling integrated capabilities, such as multi-homing, traffic engineering, mobility, private addressing (NAT), and end-to-end security.

From an architectural perspective, ILNPv6 naming semantics disentangle the layers of the protocol stack, so the core network remains unaware of the differences between IPv6 and ILNPv6, and the functions listed above are considered as end-to-end functions.

From an engineering perspective, we propose a protocol based on IPv6, the Identifier Locator Network Protocol (ILNPv6) to provide this capability. ILNPv6 is backwards compatible with the deployed Internet, and is incrementally deployable. ILNPv6 deployment requires no changes to core

routers, although end-hosts looking to use ILNPv6 need upgrades. Where possible, we have re-used existing mechanisms (e.g. DNS Security, Secure Dynamic DNS Update) without change. Our approach does not require any changes in the core of the network, so even a small number of corresponding nodes can upgrade and immediately gain incremental benefits.

ACKNOWLEDGMENTS

The authors would like to thank: Mark Handley (UCL) for helpful discussions and early feedback on multicast routing, and Devan Rehunathan (University of St Andrews) for discussions regarding mobile networks.

REFERENCES

[1] R. Atkinson, S. Bhatti, and S. Hailes, "Harmonised Resilience, Security and Mobility Capability for IP," in *IEEE Military Communications Conference*. San Diego, CA: IEEE, Nov. 2008.

[2] ——, "Site-Controlled Secure Multi-homing and Traffic Engineering for IP," in *IEEE Military Communications Conference*. Boston, MA: IEEE, Oct. 2009.

[3] D. Rehunathan, R. Atkinson, and S. Bhatti, "Enabling Mobile Networks Through Secure Naming," in *IEEE Military Communications Conference*. Boston, MA: IEEE, Oct. 2009.

[4] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," IAB, RFC 4984, Sep. 2007.

[5] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang, "IPv4 Address Allocation and the BGP Routing Table Evolution," *ACM Computer Communications Review*, vol. 35, no. 1, 2005.

[6] B. Carpenter, J. Crowcroft, and Y. Rekhter, "IPv4 Address Behaviour Today," RFC 2101 (Informational), IETF, RFC 2101, Feb. 1997.

[7] J. Saltzer, D. Reed, and D. Clark, "End-to-End Arguments in System Design," *ACM Trans. Comput. Syst.*, vol. 2, no. 4, 1984.

[8] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," IETF, RFC 4301, Dec. 2005.

[9] R. Atkinson and S. Bhatti, "Naming Enhancements for the Internet," in *IEEE London Communications Symposium*. London, UK: IEEE, Sep 2004, pp. 173–176.

[10] R. Atkinson, S. Bhatti, and S. Hailes, "ILNP: mobility, multi-homing, localised addressing and security through naming," *Telecommunication Systems*, vol. 42, no. 3, pp. 273–291, Dec. 2009.

[11] C. Bennett, S. Edge, and A. Hinchley, "Issues in the Interconnection of Datagram Networks," ARPA Network Working Group, Internet Experiment Note (IEN) 1, Jul. 1977.

[12] I. Castineyra, N. Chiappa, and M. Steenstrup, "The Nimrod Routing Architecture," IETF, RFC 1992, Aug. 1996.

[13] M. O'Dell, "8+8 - An Alternate Addressing Architecture for IPv6," IETF, Internet-Draft draft-odell-8+8-00.txt, Oct. 1996. [Online]. Available: http://arneill-py.sacramento.ca.us/ipv6mh/draft-odell-8+8-00.txt

[14] ——, "GSE - An Alternate Addressing Architecture for IPv6," IETF, Internet-Draft draft-ipng-gseaddr-00.txt, Feb. 1997.

[15] J. Shoch, "Inter-Network Naming, Addressing, and Routing," ARPA Network Working Group, Internet Experiment Note (IEN) 19, Jan. 1978.

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

[17] ——, "On Names, Addresses, and Routings (II)," ARPA Network Working Group, Internet Experiment Note (IEN) 31, Apr. 1978.

[18] J. Saltzer, "On the Naming and Binding of Network Destinations," IETF, RFC 1498, Aug. 1993.

[19] J. Chiappa, "Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture," Name Space Research Group (NSRG), IRTF, Unpublished Draft, 1999.

[20] B. Carpenter, "Architectural Principles of the Internet," IETF, RFC 1958, Jun. 1996.

[21] R. Atkinson, S. Bhatti, and S. Hailes, "A Proposal for Unifying Mobility with Multi-Homing, NAT, and Security," in *5th ACM Intl. Workshop on Mobility Management and Wireless Access - MOBIWAC2007*. Chania, Crete, Greece: ACM, Oct. 2007.

[22] IEEE, "Standard for Local and Metropolitan Area Networks: Overview and Architecture," IEEE, Standard 802, Dec. 2001.

[23] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," IETF, RFC 4291, Feb. 2006.

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

[25] ——, "Standard for Local and Metropolitan Area Networks: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method And Physical Layer Specifications," IEEE, Standard 802.3, Jan. 2002.

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

[27] T. Narten, R. Draves, and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," IETF, RFC 4941, Sep. 2007.

[28] T. Aura, "Cryptographically Generated Addresses (CGA)," IETF, RFC 3972, Mar. 2005.

[29] R. Hinden, S. Deering, and E. Nordmark, "IPv6 Global Unicast Address Format," IETF, RFC 3587, Aug. 2003.

[30] B. Wellington, "Secure Domain Name System (DNS) Dynamic Update," IETF, RFC 3007, Nov. 2000.

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

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

[33] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets," IETF, RFC 1918, Feb. 1996.

[34] R. Hinden and B. Haberman, "Unique Local IPv6 Unicast Addresses," IETF, RFC 4193, Oct. 2005.

[35] C. Vogt, "Six/One Router: A Scalable and Backwards-Compatible Solution for Provider-Independent Addressing," in *3rd ACM Intl. Workshop on Mobility in the Evolving Internet Architecture*. Seattle, WA, USA: ACM, Aug. 2008.

[36] D. Meyer, "The Locator Identifier Separation Protocol," *The Internet Protocol Journal*, vol. 11, no. 1, pp. 23–36, March 2008.

[37] D. Jen, M. Meisel, D. Massey, L. Wang, B. Zhang, and L. Zhang, "APT: A Practical Tunneling Architecture for Routing Scalability," UCLA, Technical Report 08004, 2008.

[38] D. Jen, M. Meisel, H. Yan, D. Massey, L. Wang, B. Zhang, and L. Zhang, "Towards A New Internet Routing Architecture: Arguments for Separating Edges from Transit Core," in *ACM HotNets 2008*. Calgary, AB, CAN: ACM, Oct. 2008.

[39] D. Clark, R. Braden, A. Falk, and V. Pingali, "FARA: Reorganizing The Addressing Architecture," *ACM Computer Communications Review*, vol. 33, no. 4, October 2003.

[40] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture," IETF, RFC 4423, May 2006.

[41] E. Nordmark and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6," IETF, RFC 5533, June 2009.

[42] C. Labovitz, S. Iekel-Johnson, D. McPherson *et al.*, "ATLAS Internet Observatory, Annual Report 2009," in *NANOG 47 Meeting*. Ann Arbor, MI: NANOG, Oct. 2009.

Randall Atkinson is a consultant working in Washington, DC, US. He has been active in Internet standards and research for over twenty years. He served two terms on the Internet Architecture Board (IAB), and is probably best known as the author of the original IP Security standards. He took his undergraduate and graduate degrees in Electrical Engineering and Computer Science from the University of Virginia in the US.

Saleem N. Bhatti is a Professor at the School of Computer Science, University of St Andrews, UK. His research interests are in the general area of networked and distributed systems, and especially in network architecture, and the control and management plane of networked systems. His current focus is specifically on hybrid and ad hoc architectures, edge networks, and security-related topics. He holds a B.Eng. (Hons), M.Sc. (Distinction), and Ph.D. degrees all from University College London (UCL), UK.

Stephen Hailes is a Professor in the Department of Computer Science at University College London (UCL), UK. His research interests are in the general area of wireless and mobile systems, distributed systems and security. His current focus includes wireless sensor networks, ad hoc networks, and applications of these technologies to control of autonomous vehicles, biological and veterinary sciences. He holds a B.A. and Ph.D. degrees from the Computer Laboratory, University of Cambridge, UK.