Integrating Challenged Networks

Saleem N. Bhatti
University of St Andrews, UK
saleem@cs.st-andrews.ac.uk

Randall Atkinson
Cheltenham Research, VA, USA

Joakim Klemets
Finland
joakim.klemets@gmail.com

***Abstract*—For a comprehensive information coverage across theatre, it is necessary to integrate many different sources of data which are likely to use protocols specific to a specialised purpose. For example, resource-constrained or challenged networks such as sensor systems and MANET systems, using their own protocols, may be used in conjunction with other Internet Protocol (IP) based communication and need to be integrated into the GIG. While such integration may be possible today, the engineering is complex and the resultant system may be difficult to configure and maintain, as well as being brittle when systems changes or reconfiguration is required. Furthermore, when security and identity issues are considered, the additional overhead for enabling integration within the context of sensor systems and MANETs raises challenging technology issues. Based on our ongoing work, we present a potential solution which organises such systems based on identity and location, but allows integration with Internet-wide communication.**

I. Introduction

Use of sensor networks and mobile ad hoc networks (MANETs) is crucial to future Network Centric Warfare (NCW). Such networks may be referred to as *resource constrained* or *challenged* networks, as they often have constraints (such as battery power and bandwidth). Nevertheless, sensor information may provide input to strategic, tactical, operational or C*I considerations. Meanwhile, MANET networks may be used by mobile units, such as platoons of soldiers or a group of armoured vehicles, in order to provide flexible communication between such units in mobile environments where infrastructure is not present. Sensor system communication and MANET communication share similar properties: many devices organised in an ad hoc manner comprising a capability where each node may potentially act as a source and sink of information, but may also perform a routing function to allow the nodes to form a collaborative communicating network. Another essential capability is that such a network of ad hoc nodes should have secure connectivity to external nodes, allowing integration with the Global Information Grid (GIG).

We have previously presented architectural work that considers whole site mobile networks [1] [2], harmonised secure mobile communication [3], and secure multihoming and traffic engineering capabilities [4]. This is part of our ongoing work to produce a next generation network architecture that provides enhanced capability, whilst remaining backwards compatible with today's IPv6 networks [5] [6]. We now show how the same architecture allows integration of sensor systems and MANETs, and the enabling of the 'Internet of Things' for military applications.

A. Structure of this paper

This paper takes a position that an architectural approach based on naming can allow integration of challenged network environments, as well as provide enhanced capability. In Section II, we present a discussion of the needs of ubiquitous systems and how this aligns with integration of challenged network environments. We then present our ongoing work in Section III and show how a naming approach can be beneficial in Section IV. We present an evaluation of a prototype of our approach in Section V. After listing briefly some related work in Section VI, we conclude in Section VII.

II. Ubiquitous, Pervasive and Mobile

The provision of ad hoc communication capability, especially where anything and everything is connected for communication all the time can be seen through Weiser's vision of Ubiquitous Computing (UbiComp) [7]. Today, we can see this appearing as a requirement, not just as a vision, in the military context, but with applications in domestic life also, e.g. energy-efficient smart buildings relying on a pervasive distribution of sensors. In today's deployed sensor systems, wireless communication is used but mobility is not a widely-held requirement. Increasingly, however, the use of mobile networks of sensors is becoming a military requirement. Again, there are non-military applications also, e.g. wireless body area networks for health monitoring [8]. Overall, there is benefit in moving towards direct support for sensor systems that are mobile.

For MANETs, mobility is already a requirement, and key issues for MANETs have included security and integration with other communication. Indeed, in both sensor systems and MANETs for military use, the lack of integrated support for security and mobility, with onward integration to wider connectivity means that engineering 'glue', such as gateways and proxies or other middle-boxes, need to be used. While such engineering solutions provide valuable capabilities, they also introduce extra management overhead, a single point of failure and a point of attack for adversaries.

Our approach here is to review Weiser's vision of UbiComp. We demonstrate how meeting those needs architecturally with integrated support for mobility, security and multi-homing capability, allows us to realise a clean mechanism of communication, leveraging existing engineering, that is compatible with today's IPv6 networks. One realisation of the UbiComp vision is the viewpoint that is currently being called *The Internet of Things (IoT)*.

A. End-to-End Naming Discontinuity

We consider in Figure 1 an example of a site network, which could be a sensor network deployment or a MANET. In normal operation, all communication passes through G1 and G2, nodes that cooperate/coordinate to realise a gateway function for the site. Note that in networks today, the gateways G1 and G2 could be physically heterogeneous to the other nodes in the network, as they perform a different function. However, we have previously argued that it is beneficial for the resilience of the network that in a mobile scenario for G1 and G2 to be homogenous to other nodes in the site network, so that any node can dynamically take on the gateway capability [1]. Indeed, we take the position that relying on specialist nodes of which there are relatively few would reduce the resilience of the site network. Alternatively, resilience can be increased by allowing any node to take up the roles of G1 and G2 by simplifying the implementation of the functions that gateway nodes undertake. So, it should be possible to have direct communication between *any* node in the site network and a remote correspondent node, even if normal operation might result in a network structure as shown in Figure 1. Such a position may be more appropriate to MANET networks rather than sensor networks: sensor nodes may be extremely simple devices, e.g. 'smart dust' type applications. In such cases, for resilience, it would be required to replicate nodes that could be gateway-capable.

Both sensor networks and MANETs share common characteristics as they are both specific instances of ad hoc networks. As shown in Figure 1, one key characteristic in today's engineering is use of address-based routing: the set of addresses, $\{A_s\}$, used for the nodes in the sensor/MANET network do not have any topological significance, due to the ad hoc and perhaps changing topology of the network. So, the set of addresses $\{A_s\}$ can be considered to be a set of *identities* of the nodes. This overloading of the address values provides a convenient engineering data type, and allows transmitted packets to be bound to (source and destination) nodes easily within the site network. However, this causes additional complexity when nodes require off-site communication due to the transformation functions and additional state required in the gateway nodes to permit such communication. For example, in Figure 1, this might require G1 and G2 to undertake functions to support multi-homing, mobility support and network address translation (NAT) between the set of addresses $\{A_s\}$ and, say, global addresses E_1 and E_2 assigned to external link 1 and external link 2, respectively. Of course, E_1 and E_2 could change value if the site network is mobile.

So, the line through G1 and G2 represents a *naming discontinuity* across which G1 and G2 need to provide suitable transformation functions: end-to-end transparency in communication is lost, which can raise the following security issues in communication direct to nodes within the site:

- *Authentication:* It might not be possible for a correspondent node to authenticate individual nodes or network packets from individual nodes within the site directly.

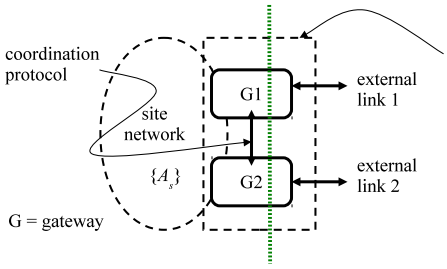

Fig. 1. Example scenario. A site network (the 'site' could be a sensor network deployment or a MANET), which has one or more gateways (G) for communication off-site. The gateway will perform various transformations of the data transmissions within the site network for correct communication off site. The site network uses a set of internal addresses, $\{A_s\}$, for intra-site communication. The dashed line through G1 and G2 represents a logical boundary where a naming discontinuity exists, and across which A_s must be transformed to permit individual nodes to communicate off-site. We assume a suitable coordination protocol exists between G1 and G2, but we do not describe it here, as it could be based on existing commercial capability.

- *Confidentiality:* It might not be possible for a correspondent node to maintain secrecy at the packet level directly to a node within the site network.
- *Site connectivity resilience.* If the functions of G1 and G2 cannot be reproduced by other nodes, failure of G1 or G2 might result in loss of connectivity for the network.

The gateways each need to act as a trusted 'man-in-the-middle'. In some scenarios, such a 'man-in-the-middle' may be desirable, but not in all situations. We have already noted the disadvantages of such a configuration.

B. Enabling Ubiquitous Connectivity

An essential capability for UbiComp systems is networking that can provide ubiquitous connectivity across many different types of underlying physical sub-networks. The natural choice for providing this is the Internet Protocol (IP). However, it was understood early on by Weiser that IP as it exists today is not ideal for this purpose, as an IP address (for both IPv4 and IPv6) names a Sub-Network Point of Attachment (SNPA) on an end-system [7]. Today, Mobile IP (v4 and v6) exists to provide mobility support for individual mobile nodes, and the IETF NEMO WG work exists to provide mobility support for whole networks of nodes, these two separate mechanisms to be merged under the IETF MEXT WG[1]. Also, specific proposals have been made for purpose-built support for ubiquitous communication (e.g. [9]). None are widely deployed for various reasons:

1) *Architectural Stagnation.* They do not change the fundamental observation made by Weiser: the IP address still names an SNPA.
2) *Engineering Complexity.* As the IP address is included in end-system state (e.g. transport protocol state), proxies and tunnels must be used in order to allow end-systems to be mobile. The use of tunnels and proxies introduces additional engineering complexity and may introduce

[1] http://datatracker.ietf.org/wg/mext/charter/

problems for application layer protocols, e.g. tunnels introduce packet overhead and may cause fragmentation of application packets.

3) *Lack of Control.* Integrating functions to provide a richer or more useful mobile connectivity landscape becomes difficult. For example, to allow host mobility, network mobility, end-to-end packet-level security with multi-homed mobile connections is not possible today, as several engineering retro-fits to IP would need to be coalesced into a single, complex engineering solution.
4) *Administrative Dependencies.* New functionality may require extensive support from a network operator, and so deployment and use of such functionality may be constrained due to network engineering. For example, multi-homing today is implemented *in the network*, by use of additional routing state (e.g. extra routing prefix information), even though it is potentially an end-to-end function, and this is causing major global concern about routing scalability [10].

For UbiComp systems, we take the position that an architecture following the end-to-end model [11] would allow provision of capability to enable sensor systems and MANET systems, including appropriate capability for mobility, security and integrated communication. However, our proposed architectural change can be realised through an engineering approach that can work on today's IPv6 infrastructure without requiring any changes to core network devices such as routers and switches.

III. Networking with names

We address directly Weiser's concern on naming and introduce an architectural change to IP, creating a richer namespace, improving network scalability and allowing easy integration of multiple network functions. Our proposal is the *Identifier-Locator Network Protocol (ILNP)*[2] [5] [6]. It allows harmonious integration of mobile hosts and networks, localised addressing, multi-homing, packet-level end-to-end security [12] [3], and supports traffic engineering options and multi-path transport-level flows [4].

Judicious engineering will allow ILNP to be introduced today onto the deployed IPv6 backbone, without requiring changes to existing core routers or routing protocols, by incrementally updating end-systems only. All enhancements to functionality are implemented in end-hosts. The network engineering is simplified, and network scalability is improved. Earlier this year, the IRTF Routing Research Group recommendation proposed that the IETF standardise ILNP [13].

A. ILNP – Architecture

In the right-hand column of Table I, we see a summary of the use of namespaces in IP today (both IPv4 and IPv6). The IP address is used for various purposes in the protocol stack. This entangles the otherwise separate layers. For example, all transport protocol flows are bound to a specific interface on a host (this is always true of transport protocols such as TCP and UDP), impacting mobility and multi-homing. Further complications result as application level flows often use the IP address and a fully qualified domain name (FQDN) synonymously, even though IP addresses are bound to a specific interface. In the worst case, this means that an application-level session that uses an IP address to form its session state becomes bound to a specific interface on a node.

TABLE I
Use of names in ILNP and IP.

Protocol layer	ILNP	IP
Application	FQDN	FQDN, IP address
Transport	Identifier, I	IP address
Network	Locator, L	IP address
(interface)	dynamic binding	IP address

ILNP provides a cleaner namespace for the protocol stack as shown in Table I. Applications can use their own namepsace, but default to using FQDNs. Entities above the network layer should use only an *Identifier (I)*, which has no topological significance and is a globally unique identity for a *node* not an interface. For example, transport protocols bind their end-system state only to the *I* value. The network layer uses topologically-significant *Locator (L)* values only for routing. Of course, some specialist applications, such as network management tools, may still make use of *I* and *L* values directly as required. Significantly, unlike IP, ILNP does not bind a globally routable name to an interface, and the binding between *I* values, *L* values and interfaces is dynamic. Nodes can choose to use multiple *I* and *L* values and multiple interfaces simultaneously and adjust dynamic bindings between them as required, e.g. to implement mobility and multi-homing.

In summary, *end-to-end protocols bind to I values*, which are used above the network layer only. *L* values are, effectively, names of networks (e.g. a network prefix as used today), with dynamic bindings between *I* values, *L* values, and interfaces.

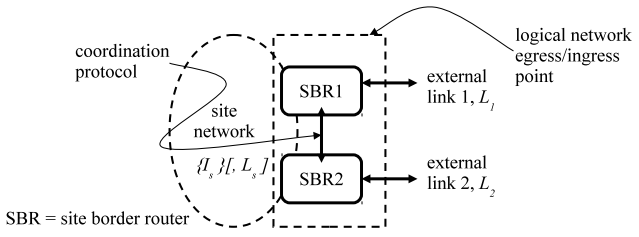

Fig. 2. An example with ILNP. We consider the network level communication only. Our site network now uses for each device an *Identifier* from the set I_s, which have no topological significance. For the purposes of intra-site routing, this is sufficient. Optionally, a local *Locator* value, L_s, for the site could be used, but is not essential. The external links use locator values L_1 and L_2, which topologically consistent with the current connectivity of SBR1 and SBR2, e.g in engineering terms, they are network prefixes as used today.

For example, as in Figure 2, consider a site node with an Identifier, I_1. This value might be used internally within the site for the operation of the routing protocol used, e.g. a sensor network routing protocol, or a MANET routing protocol. Such

[2]http://ilnp.cs.st-andrews.ac.uk/

routing protocols have scope only within the site, so as long as the set of values, $\{I_s\}$, for the site contains unique values, the site routing protocols can use these values directly. When packets are destined off-site, they are transmitted through SBR1 or SBR2, and sent with corresponding locator values. That is, if packets from a node using I_1 are transmitted through external link 1, they can be seen as having address $I_1 : L_1$, or through external link 2 as $I_1 : L_2$. Intuitively, this can be thought of as "identity I_1 is reachable at locator value L_1 or L_2". Note that this is, effectively, multi-homing for the site [4]. The binding between *I* and *L* values is dynamic. So, if some locally scoped locator value, L_s, is being used internally within the site, the SBRs use *locator re-writing* to change this to L_1 or L_2 as required. Note that this is, effectively, network address translation (NAT) for the site, without loss of end-to-end transparency [3]. With ILNP we refer to this as *localised addressing*. Also, as the values of L_1 and L_2 are independent of *I* values, as well as being dynamically bound to *I* values, changing the binding of locator values for a given I value is also easy to do. Note that this is, effectively, site mobility [2].

B. ILNP – Engineering

The engineering of ILNP exploits existing infrastructure. We have defined *ILNPv6*, an instance of ILNP that can be implemented on IPv6, but with the namespace changes described above. *L* and *I* values are encoded into the IPv6 address space – see Figure 3. The top 64 bits retain the same syntax and semantics as in IPv6, that of a routing prefix, naming an IPv6 network. So, its value can be determined using existing mechanisms, such as through IPv6 Router Advertisements. The lower 64 bits are used as a *node* identifier, not an *interface* identifier, and are *not* used for routing in the core network. So, ILNPv6 has no impact on today's IPv6 deployment. The *I* value is used only by the end-host, so only end-system code needs to be updated. The *I* values can be chosen as default for IPv6 (ala RFC4291 [14]), using the format in Figure 4.

We believe ILNPv6 provides an answer to Weiser's *architectural* concern regarding the use of IP, while offering an *engineering* solution that is deployable on IPv6 networks today, enabling challenged networks, including sensor networks, MANETs and IoT, to be integrated into existing infrastructure.

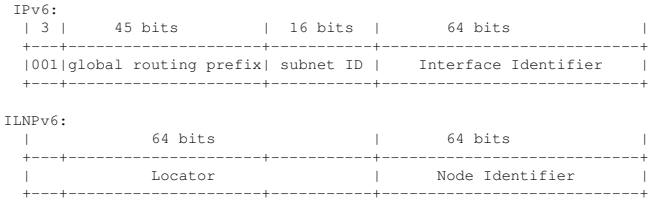

Fig. 3. IPv6 address format (RFC3587 [15]). The ILNPv6 Locator has the same syntax and semantics as that of an IPv6 routing prefix. The ILNPv6 Identifier has the same *syntax* as an IPv6 Interface Identifier, but its *semantics* are that of *node* identity and not an *interface* identity.

C. Security through Naming

As described, *I* values may use default values for IPv6. However, we can also leverage existing standards for additional security by encoding a different *I* value. For example, nodes may use cryptographically generated *I* values for high-assurance applications (ala RFC3972 [16]); or *I* values may be chosen to preserve node privacy (ala RFC4941 [17]). A basic level of location privacy can also be achieved through a simple redirection by a Locator Re-writing Relay (LRR), as shown in Figure 5. Such a mechanism avoids the overhead of tunnels, and problems such as fragmentation due to MTU size issues that can occur due to the use of tunnels. However, if on-path passive monitoring is a threat, then the SBR and the LRR can use appropriate mechanisms that exist today, for example, the use of a secure tunnel implemented with existing technology such as IPsec, L2TP etc.

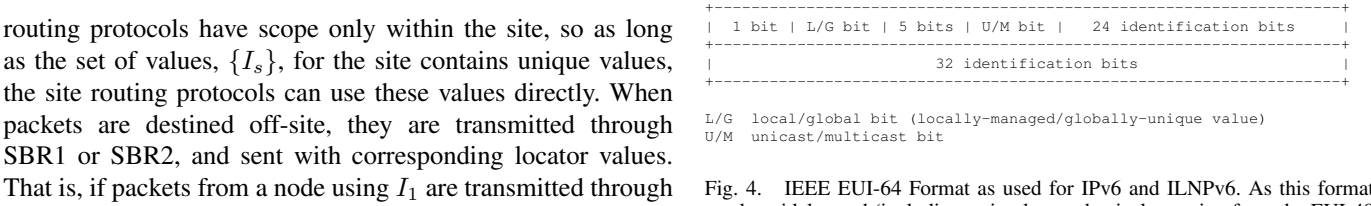

Fig. 4. IEEE EUI-64 Format as used for IPv6 and ILNPv6. As this format can be widely used (including a simple, mechanical mapping from the EUI-48 Format), it is very easy for nodes to form default *I* values using an existing interface as a source for a value, with a very high probability that the value is globally unique. However, by setting the L/G bit to L (local), it is also possible to use an administratively managed namespace for *I* values.

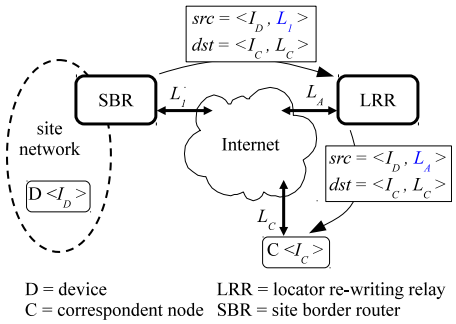

Fig. 5. A Locator Re-writing Relay (LRR) can provide a degree of location privacy. The packet from device D, with identifier I_D, is transmitted by the Site Border Router (SBR) from its current location with locator L_1 to the LRR. The LRR rewrites the value L_1 to L_A before forwarding the packet on to the correspondent, C, which has Identifier I_C. The LRR tracks the session using the pair $\langle I_D, I_C \rangle$. C always sees only L_A as the locator value for D. Incoming sessions to D can always be identified at the LRR by the use of I_D as the destination identifier. Outgoing sessions from D to C require an extension header in the first packet of the session in order to inform the LRR that L_C is the real destination for the packet.

For end-to-end packet level security using ILNP, a full description is given in our previous work [4], and we provide a summary here.

With deployed IP Security (*IPsec*) today, the IPsec Security Associations (SAs) are bound to full IP addresses at the local and remote sites, as a form of end-system identity. So, IPsec requires that the IP addresses at each end-point of the communication remain fixed – not varying over time and not varying between source and destination. However, today, for supporting private network address (e.g. use of NAT), multi-homing, and mobility, IP address values in packets might

change. Since the *High Assurance IP Encryptor (HAIPE)* used to protect existing military IP networks is a US DoD profile of the IETF standard IPsec, these issues directly apply to military IP networks.

These issues forced the IETF to develop a retrospective workaround to enable the IP Encapsulating Security Payload (ESP) to cope with these important capabilities in at least some deployments. Now, IPsec ESP is tunnelled within UDP, increasing the packet overhead and the complexity within IPsec endpoints [18]. Unfortunately, that workaround is not sufficient to resolve those same issues for the IP Authentication Header (AH). So, AH with IP remains unable to traverse NAT devices. However, no such problems exist with ILNP. ILNP Security Associations need only to include the Source Identifier and Destination Identifier, but *not* the Source Locator or Destination Locator. So, ILNP can provide IPsec/HAIPE, integrated with NATs, multi-homing, mobility and traffic engineering without loss of end-to-end transparency [4].

D. Meeting the UbiComp challenge

With respect to the issues listed above in Section II-B, we see that ILNP addresses the points raised as follows.

1) *Remove Architectural Stagnation.* ILNP introduces a fundamental change to the network communication architecture by recognising and solving the issue of the overloading of the use of the IP address and the entanglement of protocol layers: names for SNPAs are no longer part of the naming hierarchy.
2) *Reduced Engineering Complexity.* As ILNPv6 leverages existing IPv6 packet and address syntax, the core network systems and devices will be no more complex than they are today. However, as capability such as multi-homing, localised addressing and mobility can now be implemented by locator rewriting, the normal mechanisms used for such capabilities can be removed, simplifying engineering of the network. For example; to implement multi-homing no longer requires additional routing state [4]; to implement network mobility no longer requires tunnels and agents [2]; localised addressing is a first class capability rather than an engineering retro-fit [3]. Of course, the end-system stack or SBR may now have some additional complexity in managing the dynamic *I* /*L* bindings, but we take the position that this cost is also acceptable as explained below.
3) *Improved End-to-End Control.* By moving the implementation of various capabilities, such as mobility, to become functions of locator rewriting, which is performed at SBRs (or end-systems), control of those capabilities is now with the site network. For example, in order to implement multi-homing or certain types of traffic engineering with ILNP, the site network can can configure appropriate policy and security features at the SBR without the need for additional routing state across the core IP network [6].
4) *Reduced Administrative Dependencies.* Use of certain capabilities that previously required cooperation of the network operators is simplified. For example, for multi-homing, as multiple prefixes are no longer required to be present in the core routing state, multi-homing capability can be managed from the site by military personal, and there is no reliance on network administrators from a commercial network operator.

Overall, we take the position that ILNP can provide the basis on which Weiser's vision can be realised on today's IPv6 network, and is suitable for provisioning integrated capability for challenged networks such as sensor systems, MANETs, and a military-specific 'Internet of Things'.

IV. Impact of Naming Discontinuities

If we consider sensor systems and MANET systems today, they introduce naming discontinuities into the communication architecture. This has significant impacts on engineering. Consider the general diagram shown in Figure 1. Here, our 'site' network could be a sensor network (a sensor site), or a MANET (a MANET site). The gateways, G1 and G2, may need to perform several different functions. The site networks might not use IP directly, so there is need for *information transformations* (or 'mappings') between the site and IP to allow nodes within the site to communicate directly with correspondent nodes in the outside world. Such mapping could occur at several levels:

- *Application-level mapping.* The application level information may be transformed, including syntactic and/or semantic transformations, at the application level. For example, in sensor networks, a query ingress to the network may result in multiple responses from different sensor devices, and these might be collated or summarised in an application-specific manner. Such a transformation may involve the identity of one or more nodes. Such a mapping may not be required for MANET protocols.
- *Protocol-level mapping.* The nature of the communication – the transmission of data at the network level – may not be directly compatible with IP, so a mapping of protocol packet formats may be required. This may also involve the mapping of network level addresses. For example, a MANET might require a protocol mapping to allow communication between a device in the MANET and a node that is exterior to the MANET site: in this case, the network packet format and network addressing may have to be mapped. Sensor systems may also need to perform such mapping to allow communication between individual devices and nodes external to the sensor site, though, in current usage, this model of communication with individual sensors is not widely used.
- *Node identity mapping.* The identification of nodes may need to be mapped from the format used in the site to something that is compatible with IP. Note that this identity is sometimes linked to a network level address and so this function may be entangled with the protocol mapping. Such a mapping may impact both sensor systems and MANETs.

Note that where application level mapping is required, then, currently, the use of a middle-box device, such as an application-level gateway or proxy, is inevitable. However, in each of the three mappings above, we show later that identity information can be preserved for a site node when using ILNP, architecturally. For ILNPv6, from an engineering viewpoint, as the *I* values are all realised as different instances of the widely-used IEEE EUI-64 syntax (see Figure 4), ILNPv6 identities can be easily created in a standard way for many devices.

A. Using Identities

ILNP explicitly supports naming of a node, i.e. node identity. Let us consider three cases of the use of an identity.

Default. For boot-strapping purposes, or where the threat environment does not require strong identity, or where assurance on identity is to be provided above the network layer, a node can form its own identity (*I* value) by using bits from one of its existing communication interfaces. This does *not* mean that the I value is tied to a particular interface, the interface simply act as a convenient source of bits that have a high probability of being globally unique. The L/G bit is set to G.

Cryptographic protection of identity. If the threat environment dictates the strong authentication of node identity, an *I* value can be formed exactly as specified for IPv6 by use of *Cryptographically Generated Addresses (CGAs)* as in RFC3972 [16]. The management of the associated information (e.g. public key used to generate the CGA, the CGA Parameters and Hash Values, etc.) are as defined in RFC3972.

Protection against tracking and interception. If a user of a node is concerned about its communications being tracked and/or intercepted, then a node can use different *I* values over time to perturb such a threat. In order to facilitate this, IPv6, and so ILNPv6, can use *Privacy Extensions for Stateless Address Autoconfiguration* in RFC4941 [17] to assign *I* values. Also, see Section III-C regarding location privacy.

Note that we do not take a position on the utility of the use of RFC3972 or RFC4941 mechanisms with ILNPv6, our intent here is to show simply that they can be leveraged.

Indeed, it is possible for ILNP nodes to use multiple *I* values simultaneously, for different sessions. For example, a node may have three *I* values, I_d, I_c and I_p, to be used, respectively, for each of the three cases listed above. The node could have there separate sessions active simultaneously, one with each *I* value. Once a session is set up using a specific *I* value, that *I* value must remain constant for the duration of the session to maintain end-to-end transparency.

B. Discovery of Identities: Name Resolution

It is expected that applications, especially in the sensor network domain and the MANET domain, are likely to use application-specific mechanisms to discover identities, e.g. by mapping an application-specific name to an identity. For example, ad hoc protocols typically have a *discovery* phase in order to establish routes, through which they will learn the identities of the nodes on the available routes. However, there are also various mechanisms that exist today that could be adopted if a more general solution is required, especially for integration with IP.

The discovery of identities within a site network can be undertaken by utilising a number of existing mechanisms. For example, RFC3972 is actually designed to support RFC3971, *SEcure Neighbor Discovery (SEND)* [19], which allows secure discovery of node identities within a network. Where the threat environment permits, multicast DNS[3] (mDNS, part of ZeroConf[4]) could be used very effectively, as it is open source and widely available in commercial products also.

For wide-area name resolution, for example, a remote correspondent node wishing to find the *I* /*L* values for a node within the site could adopt the normal DNS for use. There are two challenges with DNS. The first is a security challenge. There may be requirements for (at the very least) records to be authenticated so they can be checked by a receiver; and for allowing new *I* and/or *L* values to be written to the DNS securely, to support dynamic changes in *I* values (e.g. if using 'private' *I* values ala RFC4941), and to support changes to *L* values (for mobility). Ongoing deployment of DNSsec [20] and Secure DNS Dynamic Update [21] will allow both these challenges to be resolved. The second challenge is that as *I* and *L* values may be updated asynchronously, and to avoid stale values being used from a DNS cache, some *I* and *L* values may need to have zero caching. Traditionally, it has been considered that caching times for DNS responses need to be large. However, our recent work shows that zero caching at an end site for selected commonly used DNS record types is entirely feasible in an operational DNS deployment [22].

V. Evaluation of a Prototype

As described in Section III-A, ILNP can be seen as an abstract architectural design, and so could be applied at any layer of the protocol stack. Our discussion in Section III-B describes ILNPv6 as a network layer protocol. In order to demonstrate the efficacy of our approach, we have built a proof-of-concept implementation of a subset of ILNPv6 for use within a single lab network.

A. ILNPv6 Prototype – Overlay

Our prototype operates as an overlay network, running on top of UDP/IPv4, which allowed us fast development compared to, say, a full implementation in an OS kernel[5]. The network stack used in the overlay is shown in Table II.

TABLE II
Overlay protocol stack used for prototype.

Protocol layer	Protocol	Comment
Application	Packet transfer	Packets with a counter
Transport	SDP	Simple Datagram Protocol
Network	ILNPv6	ILNPv6 overlay
Link	UDP/IPv4	Unreliable link layer

[3]http://www.multicastdns.org/
[4]http://www.zeroconf.org/
[5]We plan to build an open-source kernel implementation by Q4/2012

The Application protocol is a simple, one-way, unreliable packet stream, each successive packet containing a monotonically-increasing counter value. The Transport protocol, SDP, is a dummy protocol which adds a simple header and allows multiplexing. The Network layer ILNPv6 overlay uses the IPv6 header, but with the address fields interpreted at end-nodes as two 64-bit values, an *L* value and an *I* value. The ILNPv6 packet is carried in multicast UDP/IPv4 packets.

Using IP multicast enables emulation of several different networks as separate collision domains on the same physical network. We use N different multicast addresses locally to emulate N different subnetworks, as shown in Figure 6.

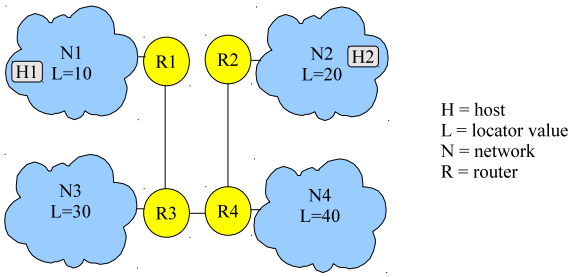

Fig. 6. A logical diagram of the evaluation testbed. Each network, N1-N4, is realised as a separate IPv4 multicast group. Each of the nodes, H1 and H2, and routers, R1-R4, are given a unique *I* value. The *L* values used at each network are as shown. The routers, R1-R4, create the topology required by joining the appropriate IPv4 multicast group.

We use the application and SDP to send a stream of (small) packets for a duration of 5mins, from H1 to H2. We emulate a low-bit rate flow: 64bytes of data transmitted every 8ms (64Kbps), which could be, for example, a compact voice stream on a MANET, or a data stream from an activated sensor. As the flow is in progress, H2 moves moves between N2 and N4 every 30s. This movement is emulated in the overlay by H2 changing its IPv4 multicast group. The new *L* value used by H2 in its new network is transmitted to H1 using an *ICMP Locator Update* message, ILNPv6's own version of the IPv6 Binding Update message [5]. The overlay ran in a closed network with servers connected by a 100Mb/s Ethernet switch.

Our implementation of the core-ILNP functionality was written in approximately 2500 lines of C code.

Although we have used individual hosts H1 and H2, these could easily be SBR1 and SBR2 from Figure 2, and represent a whole network, such as a MANET or mobile sensor network. Also, the hand-offs can clearly be seen as representing mobility, but, by virtue of the semantics of the change in L value, can also be seen as dynamic link provisioning for multi-homing or traffic engineering [4].

B. Results

We have run nine scenarios, with three different emulated packet loss values (0% as a reference, 5% and 10%), and three different emulated (one-way) delay values (0ms as a reference, 2 to 5ms emulating a LAN, and 100 to 120ms emulating a WAN, with LAN and WAN delay values chosen somewhat arbitrarily based on delays measured with *ping* from the network at University of St Andrews). We made 25 runs in each case; all measurements were considered at the application level. The case with zero configured packet loss and zero configured delay provides the baseline performance of the overlay.

TABLE III
MEASURED PACKET LOSS

Emulated Delay	Emulated Packet Loss 0%	5%	10%
None	0.17%	5.6%	12%
LAN	0.61%	6.4%	12%
WAN	0.83%	7.7%	12%

Percent packet loss (mean from 25 runs).

TABLE IV
MEASURED HANDOFF DELAY

Emulated Delay	Emulated Packet Loss 0%	5%	10%
None	22 ms	23 ms	24 ms
LAN	25 ms	26 ms	27 ms
WAN	240 ms	240 ms	240 ms

Handoff Delay (median from 25 runs).

TABLE V
NON-ACKNOWLEDGED LOCATOR UPDATES

Emulated delay	Emulated Packet Loss 0%	5%	10%
None	0.0	0.046	0.14
LAN	0.0	0.076	0.13
WAN	0.0	0.11	0.12

Non-Acknowledged Locator Updates (mean from 25 runs).

Table III shows the Measured Packet Loss for each scenario. While there is a slight increase in Measured Packet Loss as delay increased for low values of Emulated Packet Loss, this effect disappears for high values of Emulated Packet Loss. Table IV shows the median handoff delay for each scenario. The overlay prototype will handoff at 10004 ms even if the Locator Update was not acknowledged, so the median value for handoff delay is more meaningful than the mean value. Note that handoff delay is constant or nearly constant for each Emulated Delay value. Table V shows the mean number of Locator Updates that were sent for which no Locator Update Acknowledgement was received by the mobile node (H2). This could be caused either by the original Locator Update message being lost or by only the Locator Update Acknowledgement being lost. The slight decline in non-acknowledged Locator Updates for cases with 10% Emulated Packet Loss might be caused by slight differences in the timing of the several manual commands required to execute each test run, because the current ILNP overlay prototype requires several different commands to be executed in sequence on different servers.

The *R* statistical analysis software[6] was used to perform a Factorial Anova for 2 Factors, with Replication, for Measured

[6]http://www.r-project.org/

Packet Loss. This showed that only the *Emulated Packet Loss* is significant ($p < 2 \times 10^{-16}$). The *R* statistical analysis software also was used to perform a Factorial Anova for 2 Factors with Replication, for non-acknowledged Locator Updates. This also showed that only *Emulated Packet Loss* by itself is significant ($p < 2 \times 10^{-16}$). So neither measured packet loss nor non-acknowledged Locator Updates are significantly affected by delay, which means that ILNP's Locator Update mechanism should work as well over high delay links (e.g. SATCOM) as over low-delay links (e.g. fibre).

Based on these experiments, in which the mobile node sent only a single Locator Update message to its correspondent, we believe that a mobile node (or mobile network) should send periodic Locator Update messages to a correspondent node until that correspondent's Locator Update Acknowledgement message has been received by the mobile node (or mobile network). However, this is an engineering and implementation issue, rather than an architectural issue, and so can be tuned as required in any real implementation.

VI. Related work

We know of no other work that specifically seeks to integrate sensor network communication, MANET communication with Internet-wide communication, including support capabilities such as multi-homing, mobility and localised addressing. We present here a non-exhaustive list of relevant work which has not been mentioned in our earlier discussion.

The IETF 6lowpan WG[7] has been investigating the provision of IPv6 compatible protocols for use with low power wireless personal area networks. The architecture provides integration with the IP through a gateway node. 6lowpan is aimed at connectivity over networks using 802.15.4 for connectivity. More recently, the IETF ROLL WG[8] has been considering IPv6-compatible routing in similar networks. 6lowpan and ROLL are not considering the wider integration of capabilities as ILNP does.

The Host Identity Protocol (HIP) has both an IRTF RG[9] and an IETF WG[10]. HIP aims to make a clean separation of the locator and identifier roles of the IP address. HIP's approach, however, is to use public keys as the basis for the host identity namespace and to use existing IP addresses as locator values. HIP has work in progress to address mobility and other capability. HIP is from the start potentially more secure than ILNP due its use of public keys for generating host identity values. However, this also means it could be too heavy weight for use in challenged network environments.

The integration of MANETs and the Internet has been considered in various proposals [23]. The IETF MANET WG[11] is focussed on routing protocols that are IP-friendly. Other capabilities (e.g. multi-homing) are not currently being considered.

[7] http://datatracker.ietf.org/wg/6lowpan/charter/

[8] http://datatracker.ietf.org/wg/roll/charter/

[9] http://irtf.org/hiprg

[10] http://datatracker.ietf.org/wg/hip/charter/

[11] http://datatracker.ietf.org/wg/manet/charter/

VII. Conclusion

We have taken the position that an architectural viewpoint for the provision of communication capability for Ubiquitous Computing capability can result in an integrated communication architecture. We present a modification of our ongoing work on the Identifier-Locator Network Protocol (ILNP) and show how this can be used to allow integration of challenged network environments. By using identities to name nodes and locator values (e.g. network prefixes) to name networks, we allow capabilities such as multi-homing, mobility and localised addressing to be integrated with challenged network scenarios. We have built a simple prototype implementation as an overlay network for use in a lab environment, which shows the approach is feasible.

References

[1] R. Atkinson, M. Lad, S. Bhatti, and S. Hailes. A Proposal for Coalition Networking in Dynamic Operational Environments. In *Proc. MILCOM2006*, Oct 2006.

[2] D. Rehunathan, R. Atkinson, and S. Bhatti. Enabling Mobile Networks Through Secure Naming. In *Proc. MILCOM2009*, Oct 2009.

[3] R. Atkinson, S. Bhatti, and S. Hailes. Harmonised Resilience, Security and Mobility Capability for IP. In *Proc. MILCOM2008*, Nov 2008.

[4] R. Atkinson and S. Bhatti. Site-Controlled Secure Multi-homing and Traffic Engineering for IP. In *Proc. MILCOM2009*, Oct 2009.

[5] R. Atkinson, S. Bhatti, and S. Hailes. ILNP: mobility, multi-homing, localised addressing and security through naming. *Telecommunication Systems*, 42(3):273–291, Dec 2009.

[6] R. Atkinson, S. Bhatti, and S. Hailes. Evolving the Internet Architecture Through Naming. *IEEE JSAC*, 28(8):1319–1325, Oct 2010.

[7] M. Weiser. Some computer science issues in ubiquitous computing. *CACM*, 36(7):75–84, Jul 1993.

[8] D. Rehunathan, S. N. Bhatti, O. Chandran, and P. Hui. vNurse: Using virtualisation on mobile phones for remote health monitoring. In *Proc. HealthCom2011 – 13th IEEE Intl. Conf. on e-Health Networking, Applications and Services*, Jun 2011.

[9] I. F. Akyildiz, S. Mohanty, and J. Xie. A ubiquitous mobile communication architecture for next-generation heterogeneous wireless systems. *IEEE Radio Commun.*, 43(6):S29–S36, Jun 2005.

[10] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang. IPv4 address allocation and the BGP routing table evolution. *SIGCOMM Comput. Commun. Rev.*, 35:71–80, January 2005.

[11] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Trans. Comput. Syst.*, 2:277–288, Nov 1984.

[12] R. Atkinson, S. Bhatti, and S. Hailes. A Proposal for Unifying Mobility with Multi-Homing, NAT, and Security. In *Proc. MobiWAC'07 – 5th ACM Intl. Wkshp. on Mobility Mgmt. and W'less Access*, Oct 2007.

[13] T. Li. Recommendation for a Routing Architecture. RFC 6115, IETF, Feb 2011.

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

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

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

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

[18] A. Huttunen, B. Swander, V. Volpe, L. DiBurro, and M. Stenberg. UDP Encapsulation of IPsec ESP Packets. RFC 3948, IETF, Jan 2005.

[19] J. Arkko, J. Kempf, B. Zill, and P. Nikander. SEcure Neighbor Discovery (SEND). RFC 3971, IETF, Mar 2005.

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

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

[22] S. N. Bhatti and R. Atkinson. Reducing DNS Caching. In *Proc. GI2011 – 14th IEEE Global Internet Symposium*, Apr 2011.

[23] S. Ding. A survey on integrating MANETs with the Internet: Challenges and designs. *Comput. Commun.*, 31:3537–3551, Sep 2008.