IEEE MILCOM 2014, 06-08 Oct 2014, Baltimore, MD, USA
(with corrections to the camera-ready copy on IEEExplore)

Fast, Secure Failover for IP

Saleem N. Bhatti & Ditchaphong Phoomikiattisak
School of Computer Science
University of St Andrews
St Andrews, UK
Email: {saleem,dp32}@st-andrews.ac.uk

Randall J. Atkinson
Cheltenham Research
USA

***Abstract*—We describe a mechanism for fast, secure failover for IP. The mechanism is invisible to end-systems: sessions are maintained during failover. Our novel approach is to model the failover as a mobility problem, and use a mobility solution in order to implement change in connectivity. Our system is based on the Identity Locator Network Protocol (ILNP), an Experimental IRTF protocol which is realised as superset of IPv6. Our empirical results from a testbed emulation show that there is almost zero gratuitous loss during failover.**

I. Introduction

Uninterrupted communications for an individual host or an entire site network are essential for mission critical operations. Our focus in this paper is on network-layer interruptions that could be due to faults or due to malicious intent. In such cases, network-layer (i.e., IP) connectivity is perturbed, and a transition to alternative connectivity is required – *failover*. This requirement is made more challenging by the requirement for such failover to comply with the existing security policy for the communication sessions that are in progress. Ideally, the failover should also allow existing communication sessions to experience minimal disruption – *graceful failover*.

Specifically in this paper, we consider failover of communication sessions where IPsec [1]–[3] is in use. IPsec forms the basis for the *High Assurance Internet Protocol Encryptor (HAIPE)* that is used to secure military communications using IP. IETF IPsec was not designed to provide automatic failover, but proprietary solutions do exist. However, proprietary solutions usually result in vendor lock-in and often can not be easily integrated with other important capabilities, such as mobility or multihoming. Router failover is possible by using the IETF Virtual Router Redundancy Protocol (VRRP) [4], but management of IPsec in this context is not well-defined. Additionally, reliance on clusters of routers with the VRRP might lead to an operational landscape that is complex to configure, manage and maintain.

A. Contribution

Our contribution in this paper is to present a network-layer IPsec-compatible failover mechanism, using results from our previous work on mobility [5], [6], which:

- Does not require cooperation of the network provider, so the trust boundary is confined, with no dependencies on third parties.
- Offers an end-to-end (site-to-site) solution: the necessary system state is maintained between the trusted parties.
- Can operate over existing IP networks and can be deployed incrementally.
- Offers graceful failover, with our testbed-based performance analyses showing virtually zero gratuitous loss during failover.

After presenting some related work (Section II), we provide an overview of the key features of ILNP (Section III). We then provide an analyses of the IPsec-compatible, graceful failover mechanism for ILNP describing its relevance to future tactical networks (Section IV). This is followed by a performance analyses of the solution, based on a testbed emulation under different delay/loss conditions (Section V), and some concluding remarks (Section VI).

II. Related Work

In this paper, we treat network-layer (i.e., IP) failover as a mobility problem, because we see that changing the network-layer connectivity is equivalent to network-layer mobility. So, we review current mobility solutions as well as presenting an overview of ILNP.

A. Mobility mechanism as a failover mechanism

We propose a failover solution – *graceful failover* that is based on a mobility model – *soft handoff* – that can be realised with ILNP. We will show that this model is unique compared to other current mobility models (described below) in two key aspects: (i) it is completely an end-to-end model, requiring only update of the end-systems that require the capability, and is completely compatible with use of end-to-end IPsec; and (ii) it does not rely on the use of tunnels, proxies, or other middleboxes, unlike other solutions. While the use of tunnels does add some overhead, the use of middleboxes is potentially a risk for performance and security. A middlebox might become a single point of failure, or a performance bottleneck, and might offer alternative attack vectors for an adversary wishing to perturb the operation of the network. We give a brief overview of the IETF-mandated mobility mechanisms below, all of which, apart from SHIM6, rely on either a middlebox, tunnelling, or both.

Mobility extensions for IP, *Mobile IP (MIP)*, have been developed by the IETF for both IPv4 (MIPv4) [7] and IPv6 (MIPv6) [8]. Both require the use of a proxy – a Home Agent

(HA) – and the use of tunnels from the Home Network of the mobile node to its current location. MIPv6 allows some optimisation by the use of a Binding Update to bypass the Home Agent. However, its handoff model remains a hard handoff model, which typically incurs packet loss during handoff, and so many extensions and modifications have been proposed to MIPv6 in order to deal with this problem.

Mobile IPv6 Fast Handovers (FMIPv6) [9] uses tunnels between the Previous Access Router (PAR) and Next Access Router (NAR) during handoff to reduce gratuitous packet loss during handoff [10]. When the packets are traversing the PAR-NAR tunnel, routing is not optimal and the problems listed above with use of tunnels and proxies are present.

Hierarchical Mobile IPv6 (HMIPv6) Mobility Management (HMIPv6) [11] is an extension of MIPv6 that uses an entity called a Mobile Anchor Point (MAP) to reduce signalling overhead, reduce handoff latency and so reduce gratuitous loss. However, all traffic must go via the MAP.

An alternative to MIPv6 is *Proxy Mobile IPv6 (PMIPv6)* [12]. This has the disadvantages of requiring a proxy – a *Mobile Access Gateway (MAG)* – which tracks movements and registers them with another middlebox – a *Local Mobility Anchor (LMA)* – which is analogous to a HA in MIPv6. Also, traffic between MAG and LMA is tunnelled. To minimise the handoff latency of PMIPv6, a Fast Handover mechanism is proposed [13], applying the concepts of FMIPv6 to PMIPv6.

The Host Identity Protocol (HIP) supports mobility and multihoming [14], [15]. HIP uses public and private key pairs to give strong assurances of identity. The public key is used as a Host Identifier by higher layer protocols (such as TCP) to represent the host identity, whilst an IP address is used for routing. HIP requires use of strong cryptography and support infrastructure, even in IP deployments where the threat environment does not require strong cryptographic protection, e.g., inside a military network already secured using link-layer communications security (COMSEC) and possibly also transmission security (TRANSSEC) mechanisms. For improved performance, it is recommended that a HIP Rendezvous Server (RVS - a middlebox) is used to coordinate communication.

The *Locator Identifier Separation Protocol (LISP)* [16] is a network-based solution. LISP makes extensive use of tunnels and requires new network entities to be deployed and managed. LISP uses the 'map-and-encap' method for mapping IP addresses into a separate routing schema: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). LISP was originally designed for multihoming, but now has extensions proposed to support mobile nodes (LISP-MN) [17].

The *Level 3 Multihoming Shim Protocol for IPv6 (SHIM6)* [18] is a host-based solution that implements Locator-Identifier separation. SHIM6 use an extra 'shim' layer between the network and the transport protocol to perform mapping between identifiers and locators (ILNP does not require additional layers). SHIM6 was designed to enable multihoming. Mobility solutions have been proposed [19] but have high landoff latency: optimisations to improve this have also been proposed [20]. Combined mobility and multhoming is also possible [21]. Overall, SHIM6 has a high signalling overhead.

B. Transport level mechanisms

Another possibility is failover at the transport layer. There has been work on the Stream Control Transport Protocol (STCP) [22] on enabling mobility, by dynamically changing addresses [23]. Modifications to TCP for enabling multi-path TCP (MP-TCP) [24] have also been proposed for enabling mobility [25]. In both cases, the fundamental mechanism proposed is based on multihoming the end-host with multiple IP addresses (and multiple interfaces if required), and dropping one of the addresses to achieve handoff/failover.

Enabling this in the transport layer has the advantages that the transport protocol can apply congestion control across the failover, and the mechanism is end-to-end. However, the applications may need to be re-engineered for operation over a new and specific transport protocol API, and some applications may not be easy to re-engineer. For example, real-time communication uses UDP-based communication as TCP is unsuitable. Also, security concerns were flagged early on in the development process of both SCTP [26] and MP-TCP [27], some of which remain unresolved, with new attacks also being identified [28].

C. Identifier-Locator Network Protocol

The Identifier-Locator Network Protocol ILNP is an experimental protocol from the IRTF, and is described in RFCs 6740-6748 [29]–[37]. While ILNP is a general architectural alternative to the current IP architecture [38], our previous work on ILNP has demonstrated that it can support the following capabilities as first class functions; mobile hosts [39]; mobile networks [40], [41]; end-site control of multihoming and traffic engineering [42]; support for datacentres [37], [43] including for network-layer virtual machine migration based on the mobility model presented in this paper [43]. In ILNP, mobility and multihoming form a duality: the same basic mechanism is used for both at the network layer, allowing failover to naturally be modelled using a mobility mechanism [44], [45]. We present salient details of ILNP in Section III.

III. ILNP

The *Identifier-Locator Network Protocol (ILNP)*[1] is a set of backwards-compatible and incrementally-deployable extensions to IP [29]–[37] that have been recommended by the IRTF Routing Research Group (RRG) Chairs [46].

A. Names and name bindings

ILNP changes the use of IP addresses in the protocol stack, as shown in Table I. Also, instead of having static bindings of IP addresses to interfaces, ILNP uses dynamic bindings between L64 values and interfaces, and between NID values and L64 values. In the table, we see that in ILNP, transport layer protocols only use NID values in their protocol state. The NID values identify the node and are not statically bound to an interface. The L64 value has the same syntax and semantics

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

as an IPv6 routing prefix, and occupies the same place in the IPv6 packet – see Fig. 1. This means that IPv6 routers can route ILNPv6 packets as if they were IPv6 packets. However, code on end-systems needs to be modified in order to interpret the new semantics of the NID value, and manage the dynamic bindings between NID values, L64 values and interfaces.

TABLE I
USE OF NAMES IN IP AND ILNP.

Protocol layer	IPv4 and IPv6	ILNP (ILNPv6)
Application	FQDN, IP address	FQDN or app.-specific
Transport	IP address	Node Identifier (NID)
Network	IP address	Locator (L64)
(interface)	IP address	dynamic binding

As transport protocols in ILNP now only bind to the NID values, there is true end-to-end session state invariance by use of the NID values. The L64 values, and the bindings between L64 values and NID values, can be changed while a transport layer session is in progress without changing the end-to-end state, which is not possible in IP where whole addresses are used for transport layer state. As L64 values are routing prefixes in ILNPv6, by careful management of the NID to L64 bindings in the end-system stack, functionality such as mobility and multihoming is possible in ILNP without additional network support.

As an example of the way names are used in ILNP, we consider a simple example of a TCP connection at a node X with a correspondent node Y. Note that this is an architectural view and not an engineering view. In the tuple expressions (1) and (2) we show the use of names at X, where A are IP addresses, P are port numbers, I are node identifiers and L are Locator values, and subscripts identify nodes X and Y.

$$\langle tcp : P_X, P_Y, A_X, A_Y\rangle\langle ip : A_X, A_Y\rangle\langle if : A_X\rangle \qquad (1)$$

$$\langle tcp : P_X, P_Y, I_X, I_Y\rangle\langle ilnp : L_X, L_Y\rangle\langle if : (L_X)\rangle \qquad (2)$$

We see in expression (1) for IP, that the IP address values for X and Y are used at the network layer and the transport layer, and A_X is bound statically to an interface. Effectively, the TCP connection is bound to the interface and any change in the IP address of X would impact the whole stack. In expression (2) for ILNP, TCP binds only to the I values, and there is a dynamic binding for the interface to the Locater, denoted by (L_X), so that the interface can be selected for forwarding. With ILNP, changes to locator values or interfaces do not impact the TCP connection.

An ILNP node may hold multiple NID values simultaneously, and any one could be used for any communication session, as required. However, once a communication session is in progress using a specific NID value, that NID value cannot be changed.

B. Locator Update (LU) handshake

An ILNP node can change its Locator (L64) value dynamically. When this happens, it signals the change to correspondent nodes to maintain existing sessions. The signalling takes the form of a simple handshake, based on the use of a new ICMPv6 message type, the *Locator Update (LU)*, sent by the node, with a complementary message, the *Locator Update Acknowledgement (LU-ACK)*, returned by the correspondent [32]. (More details in Section IV-B.)

The LU/LU-ACK handshake is always authenticated. All LU/LU-ACK messages must include an additional ICMPv6 header that contains a strongly unpredictable Nonce value [33]. This provides lightweight, non-cryptographic protection against off-path forgery attacks. If the threat environment requires, the LU/LU-ACK messages can also be protected cryptographically using IPsec with ILNP, which will be explained in Section IV-A.

C. ILNPv6

When ILNP architecture is applied to IPv6, the result is called *ILNPv6*. Unlike IPv6, ILNPv6 does not use addresses in terms of architecture: nodes have one (or more) 64-bit *Node Identifier(s), NID)*, and one (or more) 64-bit *Locator(s), L64*. In ILNPv6, the NID and L64 values are encoded into the same packet header fields as is the IPv6 address, as shown in Fig. 1. However, the address field is used in different ways at the end-host. Core network equipment, such as switches and routers, still treat the address field as it was for an IPv6 packet, so ILNPv6 is backwards compatible. However, at end-systems, the IPv6 address field is treated as two separate 64-bit fields.

```
/* IPv6 - RFC4291 + RFC3587 */
|            64 bits                  |         64 bits         |
+-------------------------------------+-------------------------+
|   IPv6 Unicast Routing Prefix       |  Interface Identifier   |
+-------------------------------------+-------------------------+

/* ILNPv6 - RFC6741 */
|            64 bits                  |         64 bits         |
+-------------------------------------+-------------------------+
|             Locator (L64)           |  Node Identifier (NID)  |
+-------------------------------------+-------------------------+
```

Fig. 1. Comparison of IPv6 unicast address format with ILNPv6 unicast address format. The L64 value has the same syntax and semantics as the Iv6 routing prefix. The NID value has the same syntax as the IPv6 Interface Identifier, but has different semantics.

IV. IPSEC FAILOVER FOR ILNPV6

IPsec relies on session-state invariance in order to maintain a secure end-to-end IPsec session. That is, IPsec uses *identifier* values, which it considers to be stable and (semi-)permanent, for an end-system. One of these identifier values is the IP address. The source IP address forms part of the identity of an IPsec session. However, in the event of network-layer failover, IP addresses may need to change (e.g., to mitigate a DDoS attack targeted at a particular IP address).

A. IPsec Security Association state in ILNPv6

In essence, the change required for IPsec to operate with ILNP is simple: instead of using IP addresses in the *IPsec Security Association (SA)*, ILNP IPsec substitutes NID values. NID values are independent of the L64 values, and have dynamic bindings to L64 values. So, it is possible to change

L64 values for IP sessions without any impact on the IPsec SA session state. This is completely analogous to the session state description for TCP shown in tuple expressions (1) and (2): IPsec for ILNP would use I_X and I_Y.

B. Mobility Model for Failover

If we can implement failover by manipulating only the L64 values (routing prefixes) for ILNP nodes, then IPsec sessions using SAs bound to NID values remain stable. This enables IPsec sessions to failover by using new Locator values and retaining existing NID values. So, network-layer connectivity can change without adverse impact on existing IPsec sessions. With ILNP, we implement this using the ILNP mobility model, as summarised in Fig. 2.

ILNP has two models for enabling mobility. In Fig. 2a we see *hard handoff*. In this situation, the mobile node (MN), X, is communicating with a correspondent node (CN), Y, and a session is in progress. When X learns of a new L64 value, by detecting a new IPv6 routing prefix in a IPv6 Router Advertisement (RA), along with other relevant information (e.g. radio signal strength on a given interface) it can trigger a change to the new L64 value. X signals this change to Y with an authenticated *Locator Update (LU)* message, and Y responds with an authenticated LU-ACK. However, with hard handoff, there is lack of synchrony between X and Y, and for a period of time, X is using a different L64 value than the one that Y has a record of.

This model is very similar to the mobility model for Mobile IPv6 (MIPv6), and the LU message is analogous to the MIPv6 Binding Update (BU) message. Both have the drawback that during the handoff, there will be packet loss during the time X and Y have different 'correct' L64 values for X.

Our focus in this paper is the mechanism in Figure 2b, *soft handoff*, which can provide graceful failover. Here, X uses *both* the new L64 value and old L64 value until it gets the LU-ACK from Y. ILNP's clean separation of names allows this to possible, but IP's use of addresses that are bound to interfaces and thereby tied to topology (the IP routing prefix is part of the address) means that this is not possible for classic IP. Our approach minimises packet loss as X is multihomed during handoff – *graceful failover*.

As IPsec sessions are now stable across changes in network-layer connectivity, it remains to examine the performance of the graceful failover, which we present in Section V.

C. Protecting Entire Sites

In our previous work, we have shown that this mobility model is applicable to whole sites and not just individual nodes [37], [42]. So, for convenience, whilst we describe a mechanism in the context of a single node, that node could act as a router for an entire site, using ILNP Locator re-writing for the whole site to provide failover for the whole site network [37], [42], including for edge networks that are non-ILNP [47].

For example, from Figures 2 and 3, if we consider that X and Y are site border routers (SBRs) providing an IPsec tunnel between the sites that they serve, then it is clear to see how a whole site can achieve failover. Even though ILNP is designed to operate end-to-end, more details on how it can operate site-to-site (between ILNP-capable SBRs) can be found in RFC6748 [37], plus our previous work on cross-site virtual machine image migration [43] and integration of non-IP edge networks [47].

D. Protocol overhead

The ILNP mobility mechanism is essentially, an end-to-end mechanism. Whether providing mobility for individual hosts, or for a whole site, it is possible for the individual nodes to manage their own mobility in the manner that is described in this paper and our previous work.

In terms of overhead for the the mobility mechanism, there are two items of signalling that occur:

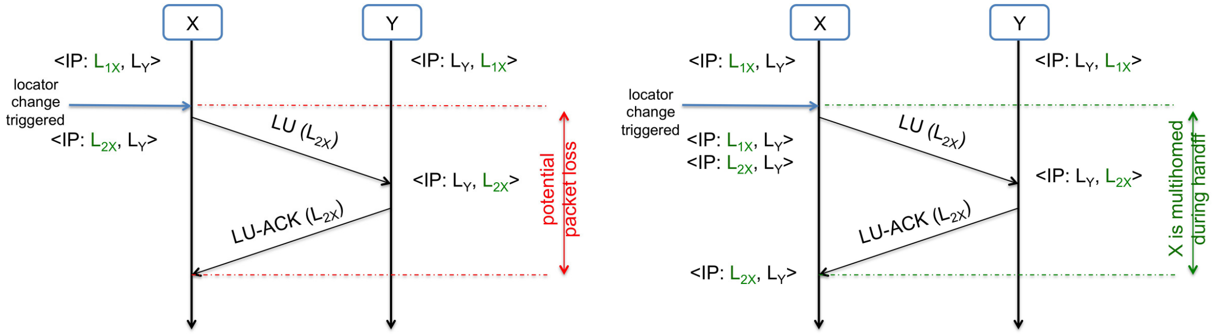
- Locator update (LU) messages: A node that is mobile must send LU messages in order to update Locator state at its correspondent node. One LU exchange (2 packets, LU and LU-ACK) is required as shown in Fig. 2. This is the handoff mechanism. The additional load here is very similar to a MIPv6 Binding Update (BU) exchange.
- DNS updates: If a node expects incoming connections, then remote nodes must have a valid Locator for its current connectivity. So, the node must perform a secure DNS update [48] in order to ensure that a DNS name look-up for that host provides its current Locator value(s). This is the rendezvous mechanism. It is only required for nodes that expect incoming connections: nodes that act as client systems only do not need to perform this operation. The overhead here will be similar to a MIPv6 node updating its Home Agent (HA) with its new Care of Address (CoA). The difference is that the HA will be at the node's Home Network (HN), whereas the DNS records for the node could be held anywhere in the DNS.

For a whole mobile site, as described in Section IV-C, ILNP allows a model where the handoff and rendzevous signalling can be managed by the Site Border Router (SBR) [37], [42]. In this case, the SBR would bear the overhead of handoff and rendezvous exchanges, but offers site-wide optimisation.

For example, if two nodes in a mobile site, X and Y, are both corresponding with host A outside the network, when the site moves (fails over), the whole site gets a new Locator value, L, and so X and Y must use that new value. X and Y would each send LU messages to A to notify it of L, as X and Y have no knowledge that they are both corresponding with A. However, if a Locator re-writing SBR is used, it would need to perform only a single LU exchange with A.

A similar situation applies for rendevous and DNS updates. A whole site 'shares' a Locator DNS record (L64 record), and individual hosts have LP DNS records that point to that single L64 record for the site [31]. So, the SBR needs to update only that single L64 record for the site.

Overall, the overhead and load of the ILNP mobility mechanism (as used for failover) is no worse than for Mobile IP, and where a SBR manages the mobility for the whole site, the overhead is reduced to a minimum.



(a) Connectivity on L_{1X} is dropped as soon as L_{2X} is detected. X signals Y with a Locator Update (LU) and Y responds with LU-ACK. When the LU is sent, X is using L_{2X} for its transmitted packets, but Y expects it to be using L_{1X}.

(b) X maintains connectivity on both L_{1X} and L_{2X} until it receives the LU-ACK from Y. During that time, it is multihomed on both the 'old' prefix (L_{1X}) and the 'new' prefix (L_{2X}), minimising loss of any packets in flight.

Fig. 2. ILNP basic failover (hard handoff, left) and graceful failover (soft handoff, right).

E. Relevance to Future Tactical Networks

Chapin and Chan discuss the requirements for a future architecture for a heterogeneous, survivable, tactical Internet [49], based on a study by staff of the Strategic Technology Office (STO) of DARPA. The security features described in this paper, when implemented by an ILNP site-border router (SBR) [37], are compatible with three of the five architectural components that are discussed by Chapin and Chan: *active border points*, *master control plane* and *remote control points* – we discuss these briefly now.

The ILNP-enabled SBR is an *active border point*. It allows very different subnets to be interconnected via the SBR and the network-layer interconnection is independent of any specific sub-network features, e.g. soft-hand off is not required at the lower layers as it is supported by ILNP. (See also [47], [50].)

The ILNP-enabled SBR is also a good candidate to form part of the *master control plane*. It allows enterprise-wide control, optimisation and management of the site networks, whilst allowing autonomous control locally with respect to operations. (See also [39], [41], [42].)

The ILNP-enabled SBR is a natural *remote control point*, at which to allow application-specific monitoring and control; it is a point at which security policy is implemented, enforced and configured; and it allows capabilities to be evolved through additional ILNP functions. (See also [43]).

Additionally, ILNP can enable the secure, ad hoc coalition of peer networks [51], and support the integration of heterogeneous edge networks [47].

There is also work in progress within the IETF on Distributed Mobility Management (DMM)[2] to update the way Mobile IP operates. DMM will ' ... *focus specifically on managing the use of care-of versus home addresses in an efficient manner for different types of communications.'* The intention is to remove the reliance on a centrally deployed mobility anchor, such as the Home Agent. DMM will be based on existing work: Mobile IPv6 [8], [52], Proxy Mobile IPv6 [12], [53] and NEMO [54]. So, it is likely that the properties of those existing mechanisms will be present, and so it is not yet clear how DMM could be used for future tactical networks with respect to the future architecture discussed here.

V. Failover performance

As noted in Section IV-A, we have end-state invariance for IPsec using the NID values, so the essential item to consider is how well the mechanism performs during graceful failover. This is achieved by an authenticated handshake (LU/LU-ACK, Fig. 2) during which time packet transfer could be perturbed. Our explanation and discussion of performance of the failover mechanism is taken directly from our previous work reported in [5]. *We are concerned particularly with graceful failover as shown in Fig. 2b, which is our novel contribution.*

However, as the L64 values change, for any new sessions, that new L64 value would need to be visible quickly to avoid initiating new sessions to the old L64 value. So, DNS entries for L64 would need to be updated, and our discussion and explanation of this issue is taken directly from our previous work reported in [6].

A. Emulation

In our previous work [5], we have measured mobile handoff performance using the same model for failover as described in this paper. We discuss those results now in the context of failover. The scenario we have used is shown in Fig. 3.

The scenario in Fig. 3 was emulated on a testbed consisting of identical desktop Linux machines in a teaching lab. Experiments were run at times of the day when the lab was not in use to prevent cross traffic interference. X, Y, R1, R2 and R3 were all separate desktop machines. The topology of the network was emulated by using different IPv6 multicast groups to emulate the three different networks as virtual networks on the same physical network. The cloud marked "Emulated Loss and Delay" was another desktop Linux machine running the widely-used *netem*[3] software, which can be used to adjust loss

[2] https://datatracker.ietf.org/wg/dmm/charter/

[3] http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

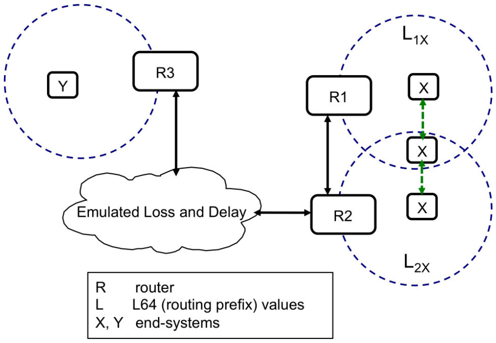


Fig. 3. Experiment topology. Y is a node that stays stable. X is shown changing connectivity between two L64 values, L_{1X} and L_{2X} (the green, dashed arrows. X fails over (changes connectivity) once every 5 seconds.

and delay for packet streams. The conditions that were emulated using netem are shown in Table II – the labels 'LAN', 'MAN' and 'WAN' for the delay are used for convenience. Each delay value was combined with each loss value.

TABLE II
NETWORK CONDITIONS EMULATED USING NETEM.

Delay [ms]	0 / 0 (LAN)	10 / 20 (MAN)	100 / 200 (WAN)
Loss [%]	0 / 0	5 / 10	10 / 20

Two values are shown in each columns as A / B, with A being the one-way value, and B being the round trip value along the end-to-end path. Note that each delay value was combined with each loss value for the experiments, i.e. a total of nine combinations.

We have emulated ILNPv6 handoffs using an overlay on IPv6. The protocol stack for the overlay is depicted in Table III. Using normal IPv6 UDP sockets on Linux, we transmitted ILNPv6 packets, and used a simple connectionless transport protocol (STP) to transmit application-level packet flows. The ILNPv6 overlay layer presented a unicast interface to the STP layer, hiding the use of multicast.

TABLE III
OVERLAY PROTOCOL STACK OF THE PROTOTYPE.

Protocol layer	Protocol	Comment
Application	Datagrams	Packets with numeric ID
Transport	STP	a simple transport protocol
Network	ILNPv6	ILNPv6 Overlay
Link	UDP/IPv6	Unreliable link layer

Our application packet flows were VoIP and ViIP flows, based on traffic characteristics described in previous studies, as shown in Table IV.

We performed 20 runs of each flow described in Table IV for each combination of values given in Table II. Each flow was 65s and there was a failover (handoff) every 5s. Our evaluation metrics are summarised in Table V.

The *loss* is the application-level packet loss, and will be assessed against the emulated loss, so we are concerned with the *gratuitous loss*, i.e. that loss due to the operation of the failover, compared to that which is 'natural' (emulated) in the network. This is a measure of how graceful the handoff is with respect to the application traffic: lower loss is better.

TABLE IV
APPLICATION TRAFFIC EMULATION, FLOWS LASTED 65S.

Description	Data Rate [Kbps]	Pkt Size [bytes]	Ref.
Skype / VoIP	64	300	[55], [56]
YouTube / ViIP	658[a]	1400	[57]

[a] This is slightly more than the 632Kbps reported in [57].

TABLE V
METRICS USED FOR PERFORMANCE EVALUATION.

Metric	Units	Summary	Fig.[a]
loss	%	Application-layer (STP)	4a
f-delay	ms	Time to complete failover	4b
f-overhead	–	# LU/LU-ACK handshakes	4c

[a] The figure showing the related testbed measurements results.

The *f-delay* is the failover delay: how long it takes for the LU/LU-ACK handshake to complete. This is likely to be affected by loss of LU/LU-ACK packets. Here, a good value is close to the 'natural' (emulated) delay of the network, i.e. close to the round-trip delay.

The *f-overhead* is the number of handshakes that are required in order for the failover to complete successfully, and will increase as the 'natural' (emulated) loss of the network increases. The optimal value is 1, as that is the minimum number of handshakes required, a single 2-packet exchange.

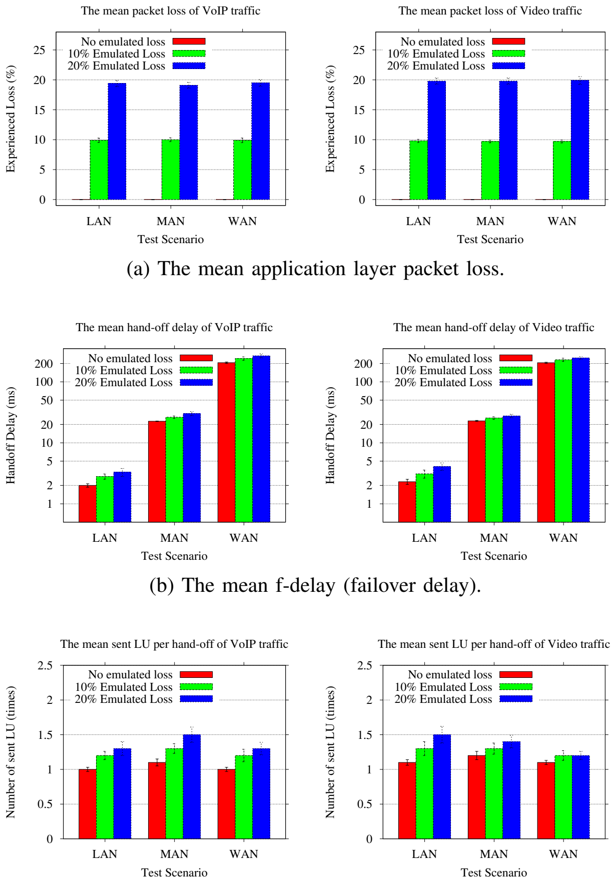
B. Results

In Fig. 4, we show the results of our emulation, with VoIP traffic in the left column and ViIP traffic in the right column.

If we consider first the loss in Fig. 4a, we see that for both VoIP and ViIP flows, the loss seen at the application-layer is the same as that of the emulated network, across all the emulated loss scenarios. That is, *there is no gratuitous loss introduced by the graceful failover mechanism.*

From the evaluation of the time taken for the failover to complete, we find in Fig. 4b that it is very close to the round-trip delay, across all the emulated loss scenarios. That is, *the time taken for the graceful failover to complete is very close to optimal in each case.* It is clear that as the loss increases, the time taken for the failover to complete also increases (as do the error bars), as LU/LU-ACK packets will be lost and will need to be retransmitted. Note that for the LAN scenario, there is zero emulated delay (within netem), so the values shown are due to the delay of the underlying connectivity.

In terms of considering the packet overhead, from Fig. 4c, we see that the value is close to 1 in all cases, and so *the failover has low overhead* and is efficient on network resources. The numbers shown are a mean from 20 runs hence the fractional values. Note also that, for the same loss values, the lower delay scenarios (LAN and MAN) seem to have worse values than for the higher delay scenario (WAN). This may be considered counter-intuitive, but points to a non-optimal implementation of our LU retransmission algorithm when network delay is low: our implementation triggers a timeout too soon and considers the LU lost.



(a) The mean application layer packet loss.

(b) The mean f-delay (failover delay).

(c) The mean f-overhead (number of handshakes).

Fig. 4. Performance of failover during emulations, means from 20 runs. Error bars plotted at 95% confidence, but may not always be visible. From [5].

C. New incoming sessions

ILNPv6 requires new DNS records in order that a DNS name lookup request for a name, N, can resolve to NID and L64 values instead of AAAA values as for IPv6 [31]. However, if the failover results in a change of L64 value, then that would mean the DNS records holding L64 values for name N should be updated when the L64 values change. This can be done securely, is already possible using Secure DNS Dynamic Update [48], and is independent of ILNPv6[4].

However, DNS allows entries to be cached, and so a cached L64 value that is no longer valid could result in erroneous session initiation requests. Therefore, where fast failover is required, time to live (TTL) values for DNS records should be kept low, ideally zero. Our previous empirical study on an operational site DNS deployment shows that it is possible to use TTL values as low as zero on DNS values for A records without incurring significant additional load on DNS [6]. From that study, the data in Table VI shows statistics for A record requests as the TTL of the DNS A records was decreased from 1800s, to 30s and then to 0s. We see that mean, 99th-percentile values and maximum values of request rates remain easily manageable on current hardware platforms.

TABLE VI
QUERY RATES FOR DNS A RECORDS. FROM [6][a].

TTL value	mean	std dev	max	~95%[b]	~99%[c]
all queries					
1800s	3.33	3.47	183	8	14
30s	4.41	4.27	261	10	16
0s	7.49	4.93	369	15	22
internal: queries from end-systems within the site					
1800s	1.31	2.98	176	8	22
30s	1.58	3.57	168	8	24
0s	2.36	3.48	68	8	15
external: queries from end-systems outside the site					
1800s	2.02	1.76	66	5	7
30s	2.82	2.28	259	7	9
0s	5.13	3.40	368	11	14

[a] Each of the TTL values were used for 7 days on an operational network, with the statistics below reported from ~155m DNS queries gathered during a 21 day period.
[b] The value of query rate at which we first see ≥95% of queries.
[c] The value of query rate at which we first see ≥99% of queries.

VI. Conclusion and Future Work

Neither the Internet Protocol (IP) nor IP Security (IPsec) were designed to support failover natively. Today, proprietary solutions exist for IPsec failover, but they require specific vendor support. We have proposed an alternative approach, based on a mobility model supported by the Identifier Locator Network Protocol (ILNP). This model provides failover for IPsec, without the need for network support – the mechanism is end-to-end (or site-to-site). The mechanism uses *network-layer soft handoff*, a mechanism that is unique to the ILNP mobility model and so cannot be supported directly by Mobile IP or any of its extensions.

Our performance evaluations from our previous work, based on flows within an emulated testbed scenario, show that *graceful failover* is possible, with virtually zero gratuitous loss, low failover delay and little signalling overhead.

For the future, we will specify the use of ILNP IPsec in more detail. Then, we intend to implement this in an operating system kernel and experimentally demonstrate this capability in an operational network environment using real failover events. This will allow us to assess in a very practical manner they overall performance and overhead of the ILNP mobility model in failover scenarios.

In terms of related future work, it is clear that the use of an Identifier-Locator paradigm for IP has the potential to introduce new configuration and management requirements for

[4] Two independent, commercial DNS implementations, NSD (from NetLabs - https://www.nlnetlabs.nl/projects/nsd/) and BIND (from ISC - https://www.isc.org/downloads/bind/), support the ILNPv6 DNS records defined in RFC6742 [31].

firewalls. It is possible that there are new opportunities for protection using firewalls with ILNP.

Acknowledgment

Ditchaphong Phoomikiattisak is sponsored by the Thai Government and the University of St Andrews.

References

[1] S. Kent and S. Keo, "Security Architecture for the Internet Protocol," IETF, RFC 4301 (PS), Dec 2005.

[2] S. Kent, "IP Authentication Header," IETF, RFC 4302 (PS), Dec 2005.

[3] ——, "IP Encapsulating Security Payload (ESP)," IETF, RFC 4303 (PS), Dec 2005.

[4] S. Nadas (Ed), "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6," IETF, RFC 5798 (PS), Mar 2010.

[5] D. Phoomikiattisak and S. N. Bhatti, "Network Layer Soft Handoff for IP Mobility," in *PM2HW2N 2013 - 8th ACM Intl. Wkshp. Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks*, Nov 2013.

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

[7] C. Perkins (Ed), "IP Mobility Support for IPv4, Revised," IETF, RFC 5944 (PS), Nov 2010.

[8] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," IETF, RFC 6275 (PS), Jul 2011.

[9] H. Koodli (Ed), "Mobile IPv6 Fast Handovers," IETF, RFC 5568 (PS), Jul 2009.

[10] E. Ivov and T. Noel, "An experimental performance evaluation of the IETF FMIPv6 protocol over IEEE 802.11 WLANs," in *IEEE WCNC 2006*, Apr 2006.

[11] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," IETF, RFC 5380 (PS), Oct 2008.

[12] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," IETF, RFC 5213 (PS), Aug 2008.

[13] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia, "Fast Handovers for Proxy Mobile IPv6," IETF, RFC 5949 (PS), Sep 2010.

[14] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol," IETF, RFC 5201 (Exp), Apr 2008.

[15] P. Nikander, T. H. (Ed)", C. Vogt, and J. Arkko, "End-Host Mobility and Multihoming with the Host Identity Protocol," IETF, RFC 5206 (Exp), Apr 2008.

[16] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)," IETF, RFC 6830 (Exp), Jan 2013.

[17] A. Rodriguez Natal, L. Jakab, M. Portoles, V. Ermagan, P. Natarajan, F. Maino, D. Meyer, and A. Cabellos Aparicio, "LISP-MN: Mobile Networking Through LISP," *Wireless Personal Communications*, vol. 70, no. 1, pp. 253–266, 2013.

[18] E. Nordmark and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6," IETF, RFC 5533 (PS), Jun 2009.

[19] A. Dhraief and N. Montavont, "Toward Mobility and Multihoming Unification - The SHIM6 Protocol: A Case Study," in *IEEE WCNC 2008*, Mar 2008, pp. 2840–2845.

[20] M. Mudassir Feroz and A. Kiani, "SHIM6 Assisted Mobility Scheme, an intelligent approach," in *IEEE CCNC 2013*, Jan 2013, pp. 725–728.

[21] A. Achour, B. Kervella, and G. Pujolle, "SHIM6-based mobility management for multi-homed terminals in heterogeneous environment," in *WOCN 2011 - 8th Intl. Conf. Wireless and Optical Communications Networks*, May 2011, pp. 1–5.

[22] R. Stewart (Ed), "Stream Control Transmission Protocol," IETF, RFC 4690 (PS), Sep 2007.

[23] A. Ezzouhairi, A. Quintero, and S. Pierre, "A New SCTP Mobility Scheme Supporting Vertical Handover," in *IEEE WiMOB 2006*, Jun 2006.

[24] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF, RFC 6824 (Exp), Jan 2013.

[25] C. Raiciu, D. Niculescu, M. Bagnulo, and M. Handley, "Opportunistic Mobility with Multipath TCP," in *ACM MobiArch 2011*, Jun 2011.

[26] R. Stewart, M. Tuexen, and G. Camarillo, "Security Attacks Found Against the Stream Control Transmission Protocol (SCTP) and Current Countermeasures," IETF, RFC 4690 (I), Sep 2007.

[27] M. Bagnulo, "Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses," IETF, RFC 6181 (I), Mar 2011.

[28] M. Z. Shafiq, F. Le, M. Srivatsa, and A. X. Liu, "Cross-path Inference Attacks on Multipath TCP," in *ACM HotNets-XII*, Jul 2013.

[29] R. Atkinson and S. N. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description," IRTF, RFC 6740 (E), Nov 2012.

[30] ——, "Identifier-Locator Network Protocol (ILNP) Engineering Considerations," IRTF, RFC 6741 (E), Nov 2012.

[31] R. Atkinson, S. N. Bhatti, and S. Rose, "DNS Resource Records for the Identifier-Locator Network Protocol (ILNP)," IRTF, RFC 6742 (E), Nov 2012.

[32] R. Atkinson and S. N. Bhatti, "ICMP Locator Update Message for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)," IRTF, RFC 6743 (E), Nov 2012.

[33] ——, "IPv6 Nonce Destination Option for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)," IRTF, RFC 6744 (E), Nov 2012.

[34] ——, "ICMP Locator Update Message for the Identifier-Locator Network Protocol for IPv4 (ILNPv4)," IRTF, RFC 6745 (E), Nov 2012.

[35] ——, "IPv4 Options for the Identifier-Locator Network Protocol (ILNP)," IRTF, RFC 6746 (E), Nov 2012.

[36] ——, "Address Resolution Protocol (ARP) for the Identifier-Locator Network Protocol for IPv4 (ILNPv4)," IRTF, RFC 6747 (E), Nov 2012.

[37] ——, "Optional Advanced Deployment Scenarios for the Identifier-Locator Network Protocol (ILNP)," IRTF, RFC 6748 (E), Nov 2012.

[38] R. Atkinson, S. Bhatti, and S. Hailes, "Evolving the Internet Architecture Through Naming," *IEEE JSAC*, vol. 28, no. 8, pp. 1319–1325, Oct 2010.

[39] ——, "ILNP: Mobility, Multi-homing, Localised Addressing and Security Through Naming," *Telecommunication Systems*, vol. 42, no. 3, pp. 273–291, Dec 2009.

[40] D. Rehunathan, R. Atkinson, and S. Bhatti, "Enabling Mobile Networks Through Secure Naming," in *IEEE MILCOM 2009*, Oct 2009.

[41] D. Rehunathan and S. Bhatti, "A Comparison of Routing for Mobile Networks," in *WiMob2010 – 6th IEEE Intl. Conf. on Wireless and Mobile Computing, Networking and Communication*, Oct 2010.

[42] R. Atkinson and S. Bhatti, "Site-Controlled Secure Multi-homing and Traffic Engineering for IP," in *IEEE MILCOM 2009*, Oct 2009.

[43] S. Bhatti and R. Atkinson, "Secure & Agile Wide Area Virtual Machine Mobility," in *IEEE MILCOM 2011*, Oct 2012.

[44] B. Simpson and S. N. Bhatti, "An identifier-locator approach to host multihoming," in *AINA 2014 - IEEE 28th Intl. Conf. Advanced Information Networking and Applications*, May 2014.

[45] D. Phoomikiattisak and S. N. Bhatti, "IP-Layer Soft Handoff Implementation in ILNP," in *ACM MobiArch 2014*, Sep 2014.

[46] T. Li (Ed), "Recommendation for a Routing Architecture," IRTF, RFC 6115 (I), Feb 2011.

[47] S. Bhatti, R. Atkinson, and J. Klemets, "Integrating Challenged Networks," in *IEEE MILCOM 2011*, Nov 2011.

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

[49] J. M. Chapin and V. W. S. Chan, "Architecture Concepts For A Future Heterogeneous, Survivable Tactical Internet," in *IEEE MILCOM 2013*, Nov 2013, invited paper.

[50] R. Atkinson, S. Bhatti, and S. Hailes, "Harmonised Resilience, Security and Mobility Capability for IP," in *IEEE MILCOM 2008*, Nov 2008.

[51] R. Atkinson, M. Lad, S. Bhatti, and S. Hailes, "A Proposal for Coalition Networking in Dynamic Operational Environments," in *IEEE MILCOM 2006*, Oct 2006.

[52] H. Soliman (Ed), "Mobile IPv6 Support for Dual Stack Hosts and Routers," IETF, RFC 5555 (PS), Jun 2009.

[53] R. Wakikawa and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6," IETF, RFC 5844 (PS), may 2010.

[54] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," IETF, RFC 3963 (PS), Jan 2005.

[55] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype traffic: when randomness plays with you," in *ACM SIGCOMM 2007*, 2007, pp. 37–48.

[56] K. Chen, C. Huang, P. Huang, and C. Lei, "Quantifying Skype user satisfaction," in *ACM SIGCOMM 2006*, 2006, pp. 399–410.

[57] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of YouTube network traffic at a campus network - Measurements, models, and implications," *Computer Networks*, vol. 53, no. 4, pp. 501–514, Mar 2009.