MILSA: A Mobility and Multihoming Supporting Identifier Locator Split Architecture for Naming in the Next Generation Internet

Jianli Pan, Subharthi Paul, Raj Jain
Department of Computer Science and Engineering
Washington University in Saint Louis
{jp10, pauls, jain}@cse.wustl.edu

Mic Bowman
Intel Systems Technology Lab
Intel Corporation
mic.bowman@intel.com

***Abstract*— Naming and addressing are important issues for Next Generation Internet (NGI). In this paper, we discuss a new Mobility and Multihoming supporting Identifier Locator Split Architecture (MILSA). There are three main contributions of our solution. First, we separate trust relationships (realms) from connectivity (zones). A hierarchical identifier system for the realms and a Realm Zone Bridging Server (RZBS) infrastructure that performs the bridging function is introduced. Second, we separate the signaling and data plane functions to improve the performance and support mobility. Third, to provide transparency to the upper layer applications, identifier locator split happens in network layer. A Hierarchical URI-like Identifier (HUI) is used by the upper layers and is mapped to a locators set by HUI Mapping Sublayer (HMS) through interaction with RZBS infrastructure. Further scenarios description and analysis show the benefits of this scheme for routing scalability, mobility and multihoming.**

Keywords— ***Identifier Locator Split, naming and addressing, mobility, multihoming, MILSA, Next Generation Internet***

I. INTRODUCTION

Naming and addressing are important Internet design issues. The availability of large-scale heterogeneous networks and the request for multiple services make it necessary to identify all the objects, especially for the mobile and multihomed hosts [1]. Current IP address centered and DNS based naming and addressing scheme cannot tackle these challenges.

In the current Internet, IP address performs the dual function of "identifier" and "locator" in which identifier explains "who" the host is and locator uniquely defines "where" the host is. This overloading leads to three main problems. First, it breaches the independence between the layers in protocol stack since application programs need to use the lower layer address directly. The second concern is mobility. Currently, DNS is used to map the names in application layer to the IP addresses in the network layer to decide where to send the data packets. However, the caching mechanism in DNS and the binding of TCP connections to IP address make it hard to provide continuous connectivity for mobile users. Mobile IP [2] is one solution, but it suffers from the triangular routing problem. Routing Optimization (RO) for Mobile IPv6 [3][13] tries to address the problem, but requires considerable changes to both end hosts. SIP [4] supports real-time multimedia applications and high-level mobility. However, the mobility in higher layer needs more support from lower layers. The third challenge is multihoming. Current solutions require announcing Provider Independent (PI) addresses in the global routing table, which increases routing tables' size and limits the routing scalability [5].

The Internet Activity Board (IAB) workshop on routing and addressing [6] reached a consensus on the scalable routing issue and the overloaded meaning of IP addresses. It urged further discussion and experiments on decoupling the dual meaning of IP addresses in the long-term design of NGI. Currently, there are several proposals for Identifier Locator Split such as HIP [7], Shim6 [8], LISP [9], GSE [10], I3 [11], etc. But most of them cannot provide a complete solution to the naming and addressing issue in the combined context of mobility, multihoming and security.

Our aim is not only to apply Identifier Locator Split to diminish the ambiguity of IP addresses and solve the routing scalability issue, but also to design a new architecture to concretely distinguish the different functional roles between organizations (trust domains) and service providers (connectivity domains) to support mobility, multihoming and routing scalability better. So we design a new Identifier Locator Split Architecture MILSA according to the three dimensional reference model shown in Fig. 1. For the management plane, we design the realms (trust domains) and zones (connectivity domains) structure, the HUI (Hierarchical URI-like Identifier) assignment and management mechanism, and the trust relationships. For the user plane, we define a new sublayer called HUI Mapping Sublayer (HMS) where Identifier Locator Split happens, and define its relationship with the Routing Sublayer (RS). In control plane we introduce a hierarchical RZBS (Realm-Zone Bridging Server) infrastructure to bridge between realms and zones. RZBSs perform the mapping between the identifiers and locators and they can backup each other for load spreading. Objects proxy mechanism enables the objects act as agents for each other.

The rest of this paper is organized as follows. Section II gives a brief description of the MILSA architecture and several design assumptions. Detailed design issues are discussed in Section III. The basic connection setup, mobility, and multihoming scenarios are analyzed and discussed in

[1] This research was sponsored in part by a grant from Intel Corporation.

Proceedings of IEEE Global Communications Conference (GLOBECOM) 2008, New Orleans, LA, USA, 30 November - 4 December 2008

Section IV. We then describe the related works in Section V. The conclusions and future works follow in Section VI.

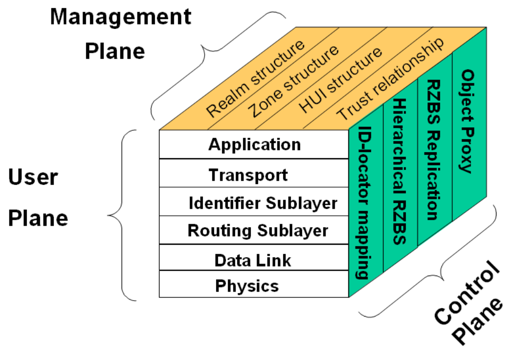

Fig. 1. MILSA Reference Model

II. SYSTEM ARCHITECTURE AND ASSUMPTIONS

A. Terminology and System Architecture

We give definitions for several keywords in MILSA [1]:

Realm is a hierarchical group of objects that logically belong to the same organization and trust each other.

Zone is a unit of topologically aggregated physical network, in which the addresses are assigned and aggregated topologically. Each object in a zone acquires at least one address or locator.

Identifier is the identity assigned to an object by its realm authority. In our design, the ID of the object plus the concatenation of hierarchical realm IDs constitute the unique identifier for the object. While IDs can be binary and are easy for machines to handle, each object may also have a human readable name. In this sense, names and IDs are interchangeable in our architecture.

Locator is the address assigned by the zone authority which uniquely identifies the current location of the object.

RZBS is server bridging between realms and zones. RZBS keeps track of the current location of the object and maps its identifier to its locator(s). RZBSs are hierarchical in accordance with the realm structure.

The MILSA architecture is illustrated in Fig. 2. The architecture has three hierarchies: realm hierarchy, RZBS hierarchy and zone hierarchy. Zone hierarchy consists of topological physical network links between the end hosts. The RZBS hierarchy is made up of an overlay network consisting of RZBSs which map identifiers to the locators. Signaling links are set up between RZBSs, which are also connected to the routing networks. The hierarchically logical belongingness and the trust relationship between different groups of objects are depicted in the realm hierarchy. Realm is a logical concept and does not have physical link to the other two hierarchies. Realm hierarchy is mapped into the RZBS hierarchy by a one-to-one or one-to-many mapping (many RZBSs serve the same realm level for robust failure tolerance or load spreading considerations) (Fig. 2 only shows one-to-one mapping).

Trust relationships are set up among RZBSs and they can authenticate or even can act as proxies for each other.

MILSA objects, including clients and servers, can have multiple identifiers belonging to different realms. Clients can have multiple locators to support multihoming.

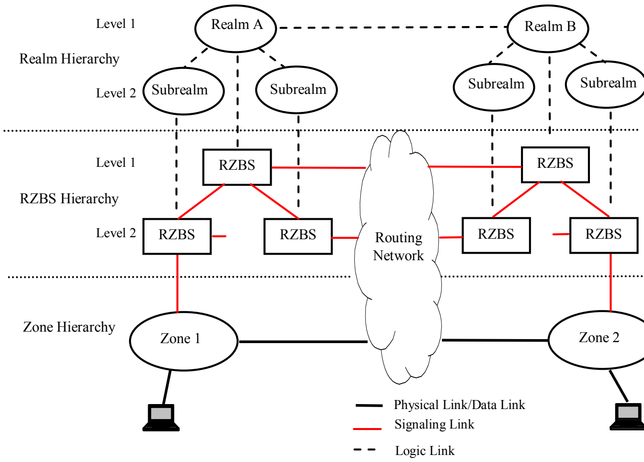

Fig. 2. MILSA Architecture

B. Assumptions

For simplicity, the following assumptions are made regarding MILSA architecture shown in Fig. 2:

1. The physical routing network uses current IP based routing mechanism, although it can adapt any new routing scheme in the future, e.g., ROFL [12].
2. For better routing scalability, the address space is strictly topologically aggregated. With this assumption, the term "locator" is used interchangeably with "address" and either one can precisely reflect the current location of the object.
3. The realm and zone membership of an object are independent. Therefore, a RZBS is not bound to any particular zone, although it may have its own attachment to a zone as shown in Fig. 2.
4. Only two levels are shown for realm hierarchy and RZBS hierarchy in Fig. 2, but actually it can be any number of levels. Zone hierarchy is not shown in Fig. 2, but it is also hierarchical, as in the current structure of IP networks.

III. DESIGN ISSUES FOR THE ARCHITECTURE

As shown in Fig. 1, MILSA involves functions in three planes. In this section, we discuss several key design issues.

A. Hierarchical Identifiers System

Based on the realm and zone structure illustrated in Fig. 2, we design a Hierarchical URI-like Identifier (HUI) system to name the objects in the network. For example, an object Bob physically connected to zone 1 and logically belonging to Subrealm-1 in Realm-A will be assigned an identifier as "Bob.Subrealm-1.Realm-A", in which the leftmost part "Bob" is the flat name in the leaf subrealm while the remaining part reflects its logical position in the realm hierarchy. The flat name part of the identifier is required to be unique in the subrealm to prevent collision. For communication inside a subrealm, we don't have to use the full identifier. E.g., in Fig. 3, if Bob wants to talk to Alice in the same Subrealm-1, he may just use "Alice" instead of the full identifier. If Bob wants to talk to Mike in Subrealm-2, he has to use "Bob.Subrealm-1" and send to "Mike.Subrealm-2". Note that the grammar of MILSA IDs is somewhat similar to those of URIs (Universal Resource Identifiers).

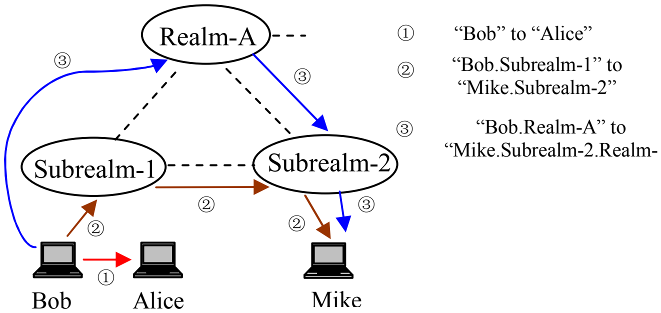

Fig. 3. Hierarchical Identifiers example

Since one object may have more than one identifiers and it is not imperative to use identifiers strictly according to the hierarchy, it's possible for this object to get another identifier "Bob.Realm-A" directly assigned and maintained by Realm-A as long as the leftmost part of an identifier "Bob" is also unique in that realm. For example, in Fig. 3, the name "Bob" should be unique in Subrealm-1 as well as in Realm-A if he uses two identifiers as shown in Fig. 3.

However, to benefit from this flexible multiple identifiers feature, we require that there exist trust relationship between the realm and subrealms, and among the subrealms.

B. *Trust Relationship*

The trust relationship is set up and maintained based on the hierarchical structure. Higher level realms have higher privilege and are in charge of all their subrealms, and the subrealms should be authenticated before they are fully trusted by the higher level realms or their peer subrealms. The trust relationship may or may not be transitive depending on the policy. For example, realm A trusts realm B, and realm B trusts realm C. Whether realm A should trust C through B, or require direct authentication of C depends on the operational policy of A. Every realm can set its internal policy for its subrealms and can set different external policy. More explicitly, suppose in Fig. 3, Realm-A has trust relationship with all his subrealms but there is no trust relationship between Subrealm-1 and Subrealm-2. In this case, if Bob want to talk to Mike, their signaling messages may have to pass their common parent Realm-A. Further trust relationship may be set up between Subrealm-1 and Subrealm-2, but initially they will use existing trust relationships for security.

Due to the mapping between realms and RZBSs, the logical trust relationships between realms can be established by authentication between RZBSs of different hierarchical levels.

C. *Identifier Locator Split*

To split identifiers from locators, we introduce a new Mapping Sublayer (HMS) into the network layer. As shown in Fig. 4, this sublayer is located between the Endpoint Headers and the Routing Sublayer (RS). The Endpoint Headers include AH (Authentication Header) and ESP (Encapsulating Security Payload) headers, Fragmentation/Reassembly Header, and Destination Option.

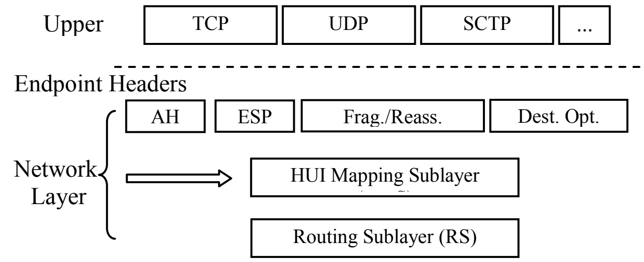

Fig. 4. HMS in the protocol stack

HMS has three main functions. Firstly, it separates the identifier from locators. The upper layers only use HUI for communication, and they don't know the current address or locator of the end host, i.e., the location information is transparent to upper layers. The lower layers only use locators for routing and they don't know anything about the identifiers used in upper layers. Secondly, HMS performs mapping from HUI to locators by interacting with the RZBSs infrastructure. Thirdly, if multihoming is enabled, the HMS maintains the HUI to locator mapping state and keeps monitoring the reachability of all the links. The mapping is set up through a set of secure signaling between the multihomed end host and the nearest RZBS. Multiple HUI to locator mapping entries are set up in the RZBS cache, each of which represents one active locator. The HUI to locator mapping can be changed based on the links availability and the policy of the host.

We design HMS this way for several reasons. HMS is below IPSec's AH and ESP headers so that the IPSec need not be aware of the locator changes due to mobility or multihoming. Thus the locator changes in the lower layer will have no influence on the stability of the IPSec security association. The fragmentation and reassembly header is also above the HMS to make reassembly robust when using different locators for different fragments if there is a broken multi-path routing.

In short, the design of HMS and its interactions with RZBSs constitute the key features of MILSA that are different from other Identifier Locator Split solutions.

D. *Signaling and Data Separation*

Another key feature of MILSA is the Signaling and Data separation. Flat structure of current Internet provides enough openness and flexibility, in which all the signaling is exchanged as data packets and there is no distinction between signaling and data. However, the flat design is also disadvantageous compared to the conventional telecom networks in the sense of efficiency and manageability. SIP [4] successfully implements Signaling and Data Separation in application layer and offers better high-level mobility and security features. However, other applications cannot benefit from it unless they deploy new SIP overlay infrastructure for the specific application, which is expensive and infeasible.

Thus, to enable more applications to benefit from MILSA, we designed the Signaling and Data Separation in network layer instead of in application layer and deployed the RZBSs infrastructure to deliver the signaling messages, while the data is transferred directly through the routing infrastructure. Since RZBSs are dedicated signaling servers, MILSA offers more efficient routing and forwarding performance.

Due to this design, the upper layer protocols and currently deployed applications do not need to be modified to benefit from the separation. The detailed separation mechanism and its mobility and multihoming scenarios will be explained with an example in the next section.

E. RZBS

Another key feature of MILSA is the hierarchical RZBSs . RZBSs are required to bridge between the logical realms and physical topological zones. They perform the mapping from HUI to locators. As discussed in Section II, several parallel RZBSs may serve the same realm. These parallel RZBSs can act as replication for each other and provide better fault-tolerance and load-balancing.

Due to the Signaling and Data Separation design, signaling messages such as the identifier to locators mapping requests and responses, the mapping coordination requests and responses, the mapping entries update requests and responses, and the authentication or other security associated messages all go through RZBSs infrastructure. The RZBSs are in charge of maintaining the most up-to-date identifier to locator mapping. Since RZBSs belong to a realm, they can be located near the backbone network, and the signaling messages can be designed to be light-weight, so that the signaling messages can be propagated at a high speed. This is important to keep the system efficient and the mapping data up-to-date.

Not like the common DHT overlay network, most parts of RZBSs infrastructure are preconfigured and are tightly coupled to the realm. Their structure is stable, so we can use DNS to map a given RZBS's identifier to its locators. (Note that DNS is used to resolve only RZBS's IDs. We then use RZBS's address to communicate and resolve the IDs of other objects using that RZBS). In Section IV, we give an example of how RZBSs handle signaling messages exchange.

F. Objects Proxy

As another important control plane feature of MILSA, Objects proxy enables all objects in MILSA including end hosts, RZBSs and other servers to act as proxies for each other after proper authentication. The proxy can be client to server, client to client, and server to server. Objects proxy offers great flexibility for the implementation of MILSA. It also provides location privacy and transparency for the roaming users. For example, if object A with an identifier in Realm-A wants object B with an identifier in Realm-B to act as a proxy for it, it can simply announce this to the RZBS of realm-A, so that all traffic to object A will be forwarded to the object B. Of course trust is required between object A and object B. Object A should also be able to terminate this proxy relationship any time it wants by sending another message to the RZBS of Realm-A. Further strict policy and security mechanism can be enforced to avoid the potential hacking or misusing.

IV. System Scenarios

In this section, we discuss several example scenarios and design considerations of MILSA.

A. Connection Setup

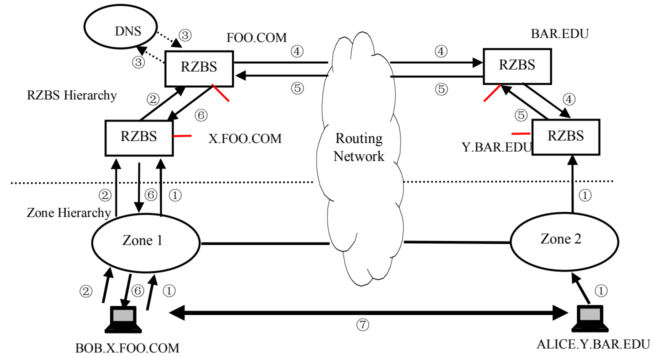

Fig. 5. Connection setup scenario

Suppose Bob has an HUI of "BOB.X.FOO.COM" and wants to communicate with Alice who is in another realm and has an HUI of "ALICE.Y.BAR.EDU". For this example, we also assume that there are only two levels of realms involved.

As shown in Fig. 5, the connection setup procedure can be described as follows:

① Bob registers his identifier to locator binding to the RZBS in charge of realm "X.FOO.COM." Similarly, Alice registers her binding to the RZBS in charge of realm "Y.BAR.EDU".

② Bob sends a request to its RZBS server to resolve the ID "Alice.Y.BAR.EDU." Suppose currently, there is no direct trust relationship between realm "X.FOO.COM" and realm "Y.BAR.EDU", thus the request is forwarded to the higher level server "FOO.COM."

③ If a direct trust relationship exists between "FOO.COM" and "BAR.EDU", we can skip to the next step 4. Otherwise, RZBS for "FOO.COM" sends a DNS request to get one locator of the RZBS for "BAR.EDU". After setting up a trust relationship with "BAR.EDU", it moves to step 4. We don't always need this DNS inquiry step. This step may only be needed once for two realms who have never talked before and no trust relationship exists between them as of yet.

④ RZBS for realm "FOO.COM" sends the request to the destination realm "BAR.EDU" which realizes that the requested identifier belongs to one of his subrealm "Y.BAR.EDU."

⑤ RZBS for realm "Y.BAR.EDU" knows the current registered locator for Alice and sends back a response.

⑥ The response is sent back to Bob.

⑦ Direct data link is set up between Bob and Alice.

The packets in the direct data link contain the source and destination HUIs, and the source and destination locators. The locators are used only for routing and the HUIs are bound to upper-layer sessions. The setup and state maintenance of the mapping from HUI to locators are fulfilled by the cooperation between RZBS infrastructure and the HMS shown in Fig 4. Note that the Signaling and Data Separation is also achieved by the RZBSs who have their own identifiers and locators.

B. Mobility

We discuss the mobility issue in three cases:

B.1 Pre-Communication Mobility

If there aren't on-going sessions with other correspondents and every time the end hosts change locator due to mobility, they should update the new locator to their nearest RZBS to enable potential correspondents to find them.

B.2 Mid-Communication Mobility

If two end hosts are talking through a direct data link and one end host moves and gets a new locator, it may want the correspondent to send subsequent packets to his new locator. In this case, the mobile host can directly announce the new locator to the correspondent through piggyback mechanism. The handover can be fulfilled with the assistance of lower layer (such as link layer) handover technologies. At the same time, the mobile host should update his locator with his nearest RZBS just as in B.1 case. Note that due to our Identifier Locator Split design, the upper-layer sessions are bound to the HUIs instead of locators and thus won't break up when locators change. MILSA mobility model supports the scenario of both sender and receiver moving and changing locators at the same time.

B.3 Roaming

Suppose a roaming user needs bridging and mapping service provided by another realm because there is no such service available close to him from the realm he belongs to. In this case, first of all, there needs to be a trust relationship between the two realms. Secondly, the roaming user should pass some authentication and authorization procedure, and he may have to accept billing and other requirements from the foreign realm. Then the foreign RZBS can act as the proxy for the home RZBS and signaling messages destined to the home RZBS to query for the current locator of the roaming user will be directed to the foreign RZBS.

This mobility model has several advantages over current solutions: First, signaling and data separation facilitates the update process of identifier to locator binding. Second, Identifier Locator Split design makes the locator changing transparent to the upper layer. Third, there is no triangular routing problem. Fourth, roaming function is supported.

C. Multihoming

As discussed in III.C, one of the major functions of the HMS is to maintain HUI to the locator set mapping context, which means if multihoming is enabled, more than one locator can be active for the end host and multiple HUI to locator mapping entries should be registered with the nearest RZBS. The end host should also be responsible for selecting or setting the policy to use for which locator to choose initially as the primary mapping for the given HUI of the end host. Since the multihomed end host can use multiple links which may belong to one or more service providers, the HMS should keep monitoring the state of these links, and update the status to the nearest RZBS so that the overlay RZBS infrastructure can find the current active locators of the end host.

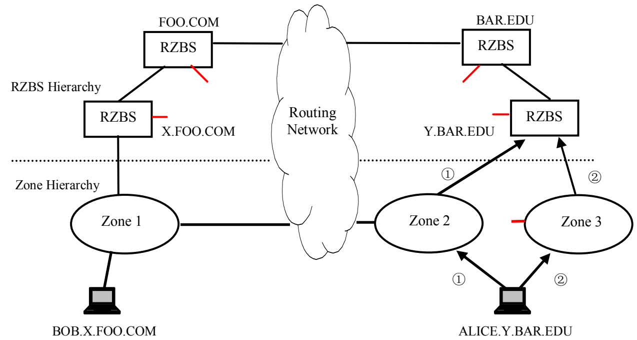

Fig. 6. Multihoming Scenario

A multihoming example is shown in Fig. 6. We suppose that "ALICE.Y.BAR.EDU" is a multihomed end host with two links connected to zone 2 and zone 3 respectively. Zone 2 and zone 3 may or may not belong to the same service provider. Zone 2 and zone 3 each assign a locator to the end host. The multihomed end host will announce these locators separately through zone 2 and zone 3 to the nearest RZBS, as shown in steps ① and ② in Fig. 6. After that, multiple HUI to locator mapping entries are set up in the RZBS hierarchy.

Based on the policy of the multihomed end host, the traffic may use the locator provided by zone 2 or by zone 3, or use them both at the same time for load spreading.

There are two cases when the link failure is detected by HMS of the multihomed end host. The first one is that link failure happens during the communication process. HMS will notify the correspondent to switch to another locator directly. Simultaneously, it will update the mapping entries in the nearest RZBS. The second case is that if the multihomed end host is not in the process of communication with other nodes, it will simply update the mapping entries at the nearest RZBS.

MILSA multihoming is more effective than other solutions. Firstly, it doesn't use any PI addresses that can result in routing scalability issue [6]. Secondly, with the RZBS and HMS, multihoming in MILSA is very efficient.

D. Mobility and Multihoming at the same time

The situation of combining mobility and multihoming is more complicated. At this time, there are no other solutions that can deal with this situation efficiently. The design of MILSA makes it simple. Both mobility and multihoming basically require updating the locator entries in the bridging layer. So mobile and multihomed hosts simply need to keep the mapping entries in the RZBS up to date, and maintain the locator set and mapping context in HMS of the local protocol stack. MILSA's Identifier Locator Split design and signaling data separation design are two key contributors to this benefit of supporting the combination of mobility and multihoming.

E. Multihoming and Multi-identifiers at the same time

We should also distinguish the concept of multihoming from multiple identifiers. Multihoming means using multiple links from one or more service providers. It's about network connectivity, not logical membership. With multiple identifiers, a user may want to use different identifiers for different applications or connections. With MILSA, the user can use both these features at the same time to achieve robustness against links failure.

F. MILSA implementation considerations

MILSA allows step by step deployment and backward compatibility. It can also adapt to future routing technologies. During the transition period, users can choose to support MILSA or not. MILSA will be transparent for those who don't want to enable MILSA. Overlay RZBSs only offers mobility and multihoming for those who register these services. This gradual deployment strategy prevents sudden infrastructure change costs and supports long term evolution of the Internet.

V. RELATED WORKS

There have been several past efforts on the design of Identifier Locator Split for the architecture of NGI. They have focused on different issues and have different features.

HIP [7] introduces a new public keys based namespace. Mobility and Multihoming are also under development in some drafts. However, for HIP, although the flat cryptographic based identifier is useful for security, it is not human-understandable. It also requires changes to the current DNS system. Mobility is achieved in two ways: UPDATE packets and rendezvous servers. First way is simple but it doesn't support simultaneous movement for both end hosts. Rendezvous servers are better but cannot reflect the organizational structure (realm), and there is no explicit signaling and data separation in the network layer.

I3 [11] adds an indirection infrastructure above the routing network to provide better multicast, anycast, and mobility. The mapping from identifier to address is called "trigger" stored in the overlay servers. However, this trigger based mobility support is limited. I3 also requires a globally unique new flat namespace for the identifiers, which may not be scalable.

Shim6 [8] uses one of its IPv6 addresses as the ULID (Upper Layer IDentifier), and it doesn't introduce new namespace. But ULID cannot reflect the logical organization (realm) and connectivity (zone) relationships. Shim6 applies only to IPv6, and requires the correspondent to be also Shim6 enabled. Mobility is not considered in Shim6.

LISP [9] uses IP-in-IP tunneling to split identifiers from locators which enables multihoming without changes to the end hosts. The mapping from identifier to RLOC (Routing Locator) is performed by the edge routers. LISP also doesn't introduce a new namespace, but requires using PI addresses as identifiers which will impair the scalability of the routing system. Fast mobility is listed for future development.

GSE [10] divides IPv6 address into identifier and locator parts. However, splitting IP address reduces the address space. The identifier is the flat MAC address and cannot reflect the organizational membership (realm structure). Mobility and multihoming are not considered in GSE.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed MILSA as a new mobility and multihoming supporting Identifier Locator Split architecture for NGI. We designed a three dimensional MILSA reference model and separated the design goals and issues in the three planes: Management Plane, User Plane and Control Plane. We use realms to represent organizational membership and zones to represent connectivity. A hierarchical RZBSs layer performs the bridging function between realm and zone. A Hierarchical Identifiers System is also designed. In addition, we introduced Signaling and Data Separation to support better performance. Identifier and Locator Split is achieved by defining a HMS sublayer to handle the mobility and multihoming and interact with RZBSs. Based on our analysis, MILSA has several advantages over other Identifier Locator Split solutions.

For our future work, we plan to design secure signaling messages and mechanisms for MILSA. Other issues such as secure and trustable routing, location privacy, network address translators (NATs), traffic engineering, multicast, and the high-level service discovery mechanisms need to be designed. We plan to do further research on these areas related to MILSA.

REFERENCES

[1] Jain, R., "Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation," IEEE MILCOM 2006, Washington, DC, October 23-25, 2006.

[2] Perkins, C., Ed., "IP Mobility Support for IPv4", RFC 3344, August 2002.

[3] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[4] J. Rosenberg, H. Scheulzrinne, G. Camarillo, et al., "SIP: Session Initiation Protocol," RFC 3261, June 2002.

[5] J. Abley, K. Lindqvist, E. Davies, et al., "IPv4 multihoming practices and limitations," RFC 4116, July 2005.

[6] D. Meyer, L. Zhang, K.Fall, "Report from IAB workshop on routing and addressing," RFC 4984, September 2007.

[7] R. Moskowitz, P. Nikander and P. Jokela, "Host Identity Protocol (HIP) Architecture," RFC4423, May 2006.

[8] E. Nordmark, M. Bagnulo, "Internet Draft: Shim6: level 3 multihoming Shim protocol for IPv6," draft-ietf-shim6-proto-09, October, 2007.

[9] D. Farinacci, V. Fuller, et al, "Internet Draft: Locator/ID Separation Protocol (LISP)," draft-farinacci-LISP-03, August 13, 2007.

[10] Mike O'Dell, "Internet Draft: GSE - an alternate addressing architecture for IPv6," draft-ietf-ipngwg-gseaddr-00, February 24, 1997.

[11] Ion Stoica, Daniel Adkins, et al, "Internet Indirection Infrastructure," ACM SIGCOMM '02, Pittsburgh, Pennsylvania, USA, 2002

[12] M. Caesar, T. Condie, J. Kannan, et al, "ROFL: routing on flat labels," ACM SIGCOMM '06, Pisa, Italy, Sept. 11-15, 2006

[13] Jianli Pan, Shanzhi Chen, "A Mobile IPv6 firewall traversal scheme integrating with AAA", IEEE WiCOM 2006, Wuhan, China, September, 2006.