Enhanced MILSA Architecture for Naming, Addressing, Routing and Security Issues in the Next Generation Internet

Jianli Pan[1, 4], Raj Jain[1], Subharthi Paul[1], Mic Bowman[2], Xiaohu Xu[3], Shanzhi Chen[4]
{jp10, jain, pauls}@cse.wustl.edu, Washington University in Saint Louis [1]
mic.bowman@intel.com, Intel Systems Technology Lab [2]
xuxh@huawei.com, Huawei Technologies Co., Ltd., China [3]
chenshanzhi@yahoo.com.cn, Beijing University of Posts and Telecommunications, China [4]

***Abstract*—MILSA (Mobility and Multihoming supporting Identifier Locator Split Architecture) [1] has been proposed to address the naming and addressing challenges for NGI (Next Generation Internet). we present several design enhancements for MILSA which include a hybrid architectural design that combines "*core-edge separation approach*" and "*split approach*", a security-enabled and logically oriented hierarchical identifier system, a three-level identifier resolution system, a new hierarchical code based design for locator structure, cooperative mechanisms among the three planes in MILSA model to assist mapping and routing, and an integrated MILSA service model. The underlying design rationale is also discussed along with the design descriptions. Further analysis addressing the IRTF (Internet Research Task Force) RRG (Routing Research Group) design goals [9] shows that the enhanced MILSA provides comprehensive benefits in routing scalability, traffic engineering, mobility and multihoming, renumbering, security, and deployability.**

Keywords— *identifier locator split, naming and addressing, mobility, multihoming, MILSA, Next Generation Internet Architecture*

I. INTRODUCTION

The interplay between the end-to-end design of IP and the vested interests of competing stakeholders has led to the Internet's growing ossification. New designs to address the major deficiency or to provide new services cannot easily be implemented other than by step-by-step incremental changes. Typical disadvantages of the current Internet design include difficulty in supporting routing scalability, traffic engineering, mobility and multihoming, renumbering, and security.

Routing scalability due to the dramatic expansion of the global routing tables is one of the most urgent among the challenges. Although noticed many years ago, it was initially alleviated by the progress in hardware technology. However, for multihoming and renumbering benefits, recently, more users are using the PI (Provider Independent) address space more like an "identifier" than an address, which breaks the address aggregation rules for scalable routing and has pushed the BGP routers in DFZ (Default Free Zone) to their capacity limit. From design perspective, it is also believed that the overloaded IP address semantics of "identifier" and "locator" is one of the major reasons for these disadvantages that are addressed in the Internet Activity Board (IAB) workshop on routing and addressing [2].

We divide the current available solutions into two classes. The first class uses "***core-edge separation approach***" trying to temporarily solve the routing table size problem by "address indirection" or "Map-and-Encap" whose goals are to keep the de-aggregated IP prefixes out of the global routing table. Typical solutions include IVIP, DYNA, SIX/ONE, APT, TRRP (all from [3]), and LISP [6]. The other class using the "***split approach***" decouples identifier from locator in IP address to solve part of the problems other than routing scalability. Typical solutions as HIP [4], Shim6 [5], I3 [7], Hi3 [8] are examples of this class of solutions.

MILSA [1] tries to design the architecture as a hybrid style that combines the two approaches in one solution to solve all the problems identified by the IRTF RRG design goals [9]. It prevents the PI address usage for global routing, and implements identifier locator split to provide routing scalability, mobility, multihoming, and traffic engineering.

The rest of this paper is organized as follows. Section II is the overview of the MILSA architecture. Detailed design enhancements and the rationale are discussed in Section III. In Section IV, the MILSA's answers to the RRG design goals are discussed. The conclusions follow in Section V.

II. MILSA OVERVIEW

In MILSA, ***realm*** [10] is a hierarchical group of objects that logically belong to the same organization. ***Zone*** is a unit of physical network topologically aggregated. Identifiers are assigned and managed by the ***realm servers***, while locators are assigned and managed by the ***zone routers*** (maintained by service providers). Zone routers are structured hierarchically which includes ***AZR*** (Access Zone Router) in the edge and ***BZR*** (Backbone Zone Router) in the trunks. AZRs perform the PI-PA addresses indirection, get the identifier to locator mapping from RZBS (Realm-Zone Bridging Server), and route the packets according to the hierarchical locator to the remote host through BZRs. The routing process can be

[1] This research was sponsored in part by grants from Intel Corporation and Huawei Technologies Col., Ltd..

assisted by the trust information in realm servers and the policies from the ***policy enforcement servers*** [11]. The hierarchical ***RZBS*** infrastructure is a global mapping system that keeps track of the current location of an object and maps its identifier to its locator(s) [1].

The secure signaling and data delivery in MILSA is ensured by a two-level trust relationship. The ***"inter-realm trust"*** is the trust relationship between organizations, which is absent in the current flat Internet design. The other level is the "***inter-host trust***" which means the end-to-end security and trust between two end-hosts. Simple example of this inter-host trust is the on-session security association negotiation between any two peers. In fact, HIP [4] supports only inter-host trust while MILSA adopts both.

MILSA uses a secure ***HUI*** (Hierarchical URI-like Identifier) system to name the objects in the network. A HUI contains two parts: a flat part for inter-host trust and a hierarchical part for inter-realm trust.

The identifier locator split happens in the network layer, which is divided into two sublayers. The identifier sublayer performs the mapping from identifier to locator by interacting with the RZBSs infrastructure, and if multihoming is supported, it keeps monitoring the reachability of all the links and notifies the RZBS of any changes. Routing sublayer is like the current IP. It only cares about locator based routing and doesn't know identifiers.

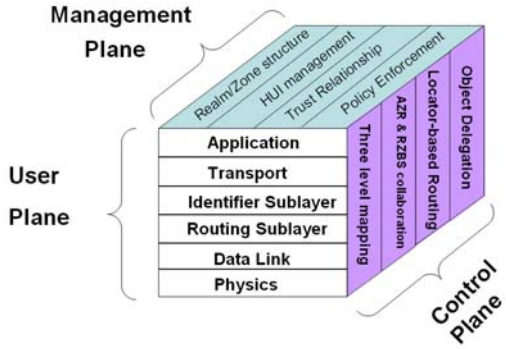

Fig. 1. Revised MILSA reference model

As Fig.1 shows, our revised MILSA reference model is to act as a design guide for NGI in the context of a converged network. In the ***user plane***, the overloaded IP address is decoupled as identifier and locator. IP address is only about "location" and is used for routing and is transparent to users and upper layer applications. Upper layer protocols are bound to HUI instead of IP addresses. The ***control plane*** is in charge of the mapping from identifier to locator and performs the locator-based routing. The ***Management plane*** function is responsible for the management of the realms and zones structure, the identifiers assignment and management, inter-realm and inter-host trust, and the policy enforcement.

MILSA follows a signaling and data separation design to gain efficiency, controllability and manageability similar to that of the conventional telecommunication networks. Dedicated RZBSs form the signaling level, while the data routing level consists of the AZR and BZR hierarchy.

Objects delegation enables all objects in MILSA to act as proxies for each other after proper authentication which offers great flexibility for the implementation and also provides location privacy for roaming users.

III. Design Enhancements And Rationale Analysis

Several design enhancements to MILSA architecture are analyzed and discussed in this section.

A. Hybrid Architecture Design

MILSA's hybrid design combines the core-edge separation approach and the split approach. By using PI-PA address indirection, it allows users to continue to use PI addresses transparently without scalability problem and renumbering cost. By using identifier locator split and the overlay RZBS signaling infrastructure, it can meet all the goals [9] and support easy transition and long-term evolution.

In MILSA, we let the RZBS, AZR, and DNS work together to implement the hybrid design. Suppose the PI address site is not MILSA-aware and wants to use PI address as usual, to make this happen without harming the global routing system, what we do is to let the PI prefixes bind to a group HUI by MILSA without users' participation. Since the binding of PI prefixes to HUI is not very dynamic or even a one-by-one binding, it can be stored and retrieved by adding new RR (Resource Record) into DNS. At the same time, the closest AZR of the end-host assigns a entry router's PA address to this site, and the triple-binding of "HUI—PI prefix—PA address" is registered in the overlay RZBS infrastructure, i.e., the PI prefix is mapped to the entry router's PA address for global routing and PI address is only used for local routing in the edge network. Again, this registration procedure is transparent to the legacy PI address edge network.

When the PI legacy edge network is attached to the core network, the AZR queries the HUI of the PI prefix from DNS and assigns its own PA address and registers the triple-binding in the RZBS infrastructure.

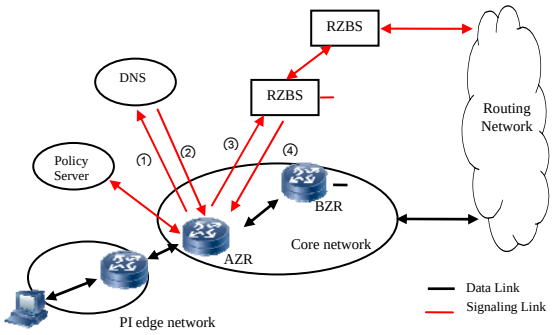

Fig. 2. Registration signaling procedure

The registration procedure is illustrated in Fig. 2 as follows:

①② Local AZR assigns its own PA address as the entry router for the PI edge network and looks up the HUI corresponding to the PI prefix through DNS.

③④ The AZR registers the triple-binding with the RZBS.

After these steps, all the PI address users in that edge network are globally reachable without any changes to the end-host and the DFZ routers only use the PA addresses for global routing. If the end-host sends packets to a globally reachable locator, the AZR simply replaces the source PI

address with the exit AZR's PA address and routes the packets to the destination locator.

If the end-host is MILSA-aware, it will contact the RZBS directly and get the remote hosts' current locator for the given HUI and send out the packets. The registration and related signaling scenarios illustrated in Fig. 2 are for enabling coexistence of the legacy PI address hosts in the new MILSA network. For the legacy PA address networks, since they won't harm the routing scalability, we allow them to be used continuously in the current network, which means that in the transition period of MILSA evolution, we allow the legacy PI/PA hosts to talk to the legacy PI/PA hosts, and MILSA hosts talk to MILSA hosts. However, in the future MILSA design, we will also consider the situation of allowing the MILSA hosts talk to legacy hosts through some kinds of proxy mechanisms. Before that happens, it is also possible to introduce "dual stacks" or "stack bypassing" in the MILSA hosts' network stack to let them talk to the legacy hosts. However, in these scenarios, MILSA hosts degrade to normal IPv4/IPv6 hosts.

In the current scenarios, whether the end-host is MILSA-aware or not, the end-user's PI prefix won't be injected to the DFZ global routing tables and won't create the routing scalability issues. Policy operations may be involved which are not shown in detail for simplicity.

In short, this hybrid design enables PI addresses to be used without violating the topological prefix aggregation rules for scalable routing, and it is transparent to the end-host. It can also benefit from the overlay RZBS infrastructure in mobility, multihoming, renumbering, and traffic engineering. It is a solution for short-term as well as long-term design goals.

B. Secure Hierarchical Identifier System Enhancements

The new HUI structure consists of two parts: the ***flat cryptographic part*** and the ***hierarchical logical part***. The first part is the hash of public key that uniquely identifies an object. This flat part is similar to the HIT (Host Identity Tag) of HIP [4] which is used for end-to-end authentication and data security. However, it is also possible to incorporate other new global end-to-end mechanisms in the future. The second hierarchical part defines the organizational affiliation which logically locates the position of the object in the realms. This part is different from the DNS name in that HUI is purely about logical affiliation while in the DNS name the location and logical affiliation are somewhat overlapped, which is exactly like the "identifier" and "locator" overlapping in the current IP address. In both cases the overlapping causes problems. Moreover, the DNS name is only used in application layer but HUI is also used in transport layer and network layer. As a simple example to illustrate the difference between DNS name and our identifier, "mail.yahoo.com.cn" represents a host or server located in China belonging to Yahoo. Although we can explain ".cn" to be a super big organization, this part of DNS name is more an indication of location. To avoid this ambiguity, in MILSA, the HUI hierarchy in the second part only indicates the logical organizational affiliation.

We concatenate these two parts into a HUI. For example, the HUI for a mail account object may be named like:

"{Hashed Key}.MichaelPhelps.mail.us.yahoo.com"

The left part is the hashed public key of the object and the right part is the logical affiliation of the object.

We now discuss the rationale for the design of this concatenated HUI. To name an object, we can use flat, hierarchical, or descriptive names. Since they have different advantages and disadvantages, a combined one is desirable. In our HUI design, the first flat part is easy to process by machines but hard to remember for people, so it is used to perform the end-to-end security which may have critical speed or performance requirement; the second hierarchical part is easy to understand by people and easy to organize and manage by using a tree-like realm structure (refer to the realm-zone definition in [1]). In short, we gain several advantages through this concatenation. Moreover, to ease the transition process, we can also encode the HUI to fit the HUI structure to the fixed length, e.g., the 128 bits long as IPv6 address. Actually, descriptive name is also used for service discovery in the service model of MILSA which will also be addressed later in this paper. Note that this HUI design is also consistent with our trust relationship design in which the first part is for end-to-end inter-host trust while the second part is for hierarchical realms' inter-realm trust.

C. Three-level Mapping

We also need to clarify how HUIs are used in MILSA and the underlying rationale. We assume humans prefer using the easy-to-understand DNS name in the application layer. So we allow the mapping from the general DNS name to the HUI since the first part of our HUI contains the "not very user-friendly" hashed key. Note that this mapping is not very dynamic and we can implement it by adding a new RR type into DNS. After getting the HUI for the given DNS name, it can be further resolved into the current locator of the object.

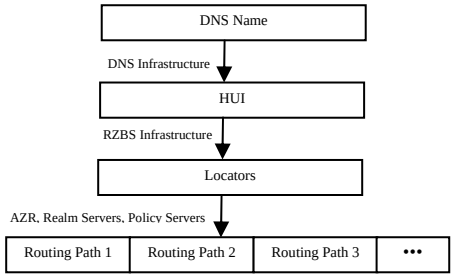

Fig. 3. Name resolution and mapping

As for the mapping from HUI to locators, the detailed design for RZBS hierarchy was presented in MILSA [1]. However, the protocol to achieve potentially greater efficiency in this overlay network is open for future design. LISP-DHT [12] is one of those mechanisms but still needs further discussion. Fig. 3 illustrates the three level mapping and the entities or systems involved in initiating or assisting the mapping. For the mapping from the locator to the routing paths, we allow the cooperation among the three planes to assist in deciding the routing paths and policies.

D. Hierarchical Code Based Locator Structure

Different from the current IP prefix based flat routing which leads to several problems, we design a ***Hierarchical Code***

Based Locator Structure with an example shown in Fig.4 [13].

Fig. 4. Hierarchical code based locator structure

The length of various code fields can be set appropriately. For backward compatibility during transition, we can allow the "End-host code" part to be the current IP address. However, in this case, this part is no longer used for global routing but may still be used for local routing in the edge network. This means that different zones can have the same end-host code space thus the address space is enlarged. The End-host code by itself is not globally unique, but with the other codes, the whole locator is globally unique. The code in one level can be distributed as a block and the service provider can split it further for finer granularity aggregation.

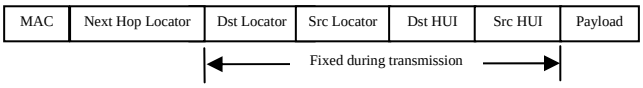

Fig. 5. Packet format

Packet format is shown in Fig.5. The packets are routed hierarchically by the AZRs and BZRs based on the destination locator, i.e., the locator is mapped into routes. In every forwarding hop between two neighboring Zone Routers, the "Next Hop Locator" is filled with the locator of next-hop Zone Router. Since more than one route to a specific destination may exist, we allow "inter-realm trust" policies and security operation to be involved in deciding which route to choose for a single level code in the locator. This means several BZRs can serve one zone level (country, province, or region) for replication or security reasons. Note that regardless of whether an end-host uses PI address or not, after the packet is sent out, the source and destination locator headers remain unchanged until it reaches the destination AZR which will perform local routing.

This locator structure and routing scheme ensures that the locator is topologically aggregated, and allows us to continue using PI addresses without harming the global routing. It also allows trust relationship, security, policies, and replication to be taken into routing decision.

E. Cooperative Mechanisms Among The Three Planes

End-to-end design of the current Internet lacks the manageability, accountability, and security compared with the telecommunication networks. Indirection designs such as I3 [7] and Hi3 [8] have emerged to assist the mobility, multihoming, and multicast handling. For the future Internet design, we believe both designs should be supported. In this section, we discuss some cooperative mechanisms among the control, user, and management planes.

E.1 Control Plane and Management Plane

MILSA allows management plane function such as the inter-realm trust and the other policies to assist the locator-based routing in the control plane. The three-level mappings also require them to work together.

E.2 Control Plane and User Plane

For better mobility and multihoming support, in user plane, end-hosts need to maintain their active locator sets and update the mappings with the RZBS infrastructure in the control plane. The indirection of PI address to PA address also needs the cooperation between these two planes.

E.3 User Plane and Management Plane

The end-host gets an HUI from realm servers after proper authentication; the DNS name to HUI mapping stored in DNS may also be requested by the end-host; the public key based first part of the HUI may also require authentication and authorization from realm servers; end-host may also have to obey the policies enforced before using the network resources. All these operations require proper cooperation between the user plane and the management plane.

F. Multicast and Manycast

End-to-end design of the current Internet basically doesn't support multicast well. IP multicast is not widely deployed due to scalability and other problems and is not available for average users. Multicast in MILSA is HUI-based instead of address-based. Sender sends the packets to a group HUI instead of IP addresses which makes MILSA multicast like "deliver this information to these people" instead of "deliver these packets to these addresses". In basic MILSA multicast design, we designate a specific ***multicast HUI*** for a multicast group. The locator bound to this HUI should be a locator of an AZR or BZR instead of an end-host. This AZR or BZR is in charge of maintaining a state list of the group, i.e., the end-hosts who want to join this multicast HUI group register their HUI and corresponding locator with the AZR or BZR which own the group HUI. After the multicast packets arrive at the AZR or BZR, it will look up the group state and replicate the packets to the members. In practice, to facilitate this procedure, we can use dedicated multicast routers for the zones or sub-zones depending on the specific requirement.

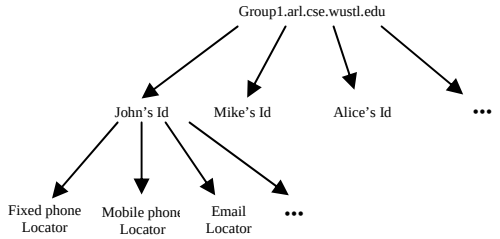

Fig. 6 Simple many-cast example.

To ensure efficient multicast of messages over each link, in MILSA, multicast server doesn't replicate the packets directly to each locator. The topologically aggregated locators in the state list form a tree with the root node of the multicast server, and the locators with the same hierarchical codes will be served by their multicast server in that level. Actually a tree comprised of multicast servers in different levels is constructed to ensure efficient multicast. We can also use the multicast server's locator in its upper level multicast server's state list to replicate packets to the whole sub-zone without designating every locator in the state list.

We also have ***manycast*** in MILSA to enable the packets to be delivered to a user with different locators for different devices or services. Fig. 6 gives a simple manycast example.

Note that MILSA keeps the global routing system unaware of the multicast thus improves the system scalability.

G. Integrated MILSA Service Model

MILSA naturally supports improved upper-layer services because of the identifier locator split, signaling and data separation, and secure hierarchical HUI design. MILSA entities may request a specific service from an object or ask implicitly for certain services available from the network. MILSA service model is designed to support both cases.

G.1 Service Integration

For example, a user named John has different identifiers for different services. For email service, he may have both "John.us.gmail.com" and "John.cse.wustl.edu"; for Instant Messenger, he can have "John.us.hotmail.com"; for mobile phone service, he can have "314-xxx-xxxx.tmobile.com". In the current Internet, every service is independent. However, in MILSA, different services can be integrated. John may need a uniform identifier to allow others to reach him by all the available means without knowing every detailed identifier assigned by specific service provider. Furthermore, John can set his own "***profile***" for the policy of the different services. For example, John may not want to be disturbed after 10 pm then the mobile phone identifier can be disabled after 10 pm and the policy can be updated in the server. To accomplish this, we allow a user or object to have an "***integrated identifier***" or "***master identifier***". Identifiers for different services can be bound to this master identifier. This binding and the profile can be stored in the "***profile server***". When this user or object is called, the identifier-to-locator mapping request will be forwarded to the RZBS of this user who will further look up the related identifiers and the profile from the profile server and decide which service and locator to return to the caller. The caller certainly can narrow down the services wanted by explicitly indicating it along with the callee's master identifier. In the future, the multicast or manycast servers can work with the profile servers and be integrated into the service model.

G.2 Service Discovery

Since the user may not know exactly what identifier or DNS name to use to get a specific service, our service model also allows implicit service discovery by users. Descriptive name is suitable for service discovery. By using specific service descriptor, the network can respond with the closest server's DNS name which will be further mapped into HUI and locator. A service discovery example is shown in Fig.7.

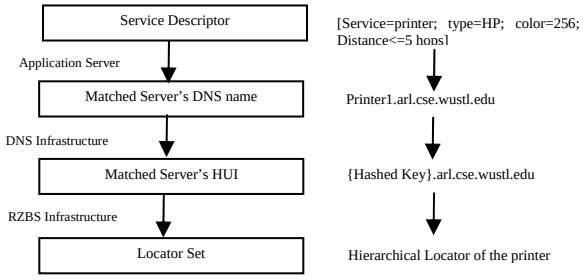

Fig. 7 Service discovery example

IV. MILSA'S ANSWER TO THE RRG DESIGN GOALS

In this section, we analyze and discuss how MILSA meets the design goals [9] set by IRTF RRG.

A. Routing Scalability

MILSA's hybrid design adopts short-term PI-PA addresses indirection mechanism as well as identifier locator split to tackle routing scalability challenges. PI-PA address indirection in AZR makes it possible to continue using the PI addresses transparently without affecting the global routing system. The identifier locator split mechanism further eliminates the necessity of using the PI addresses. Only topologically aggregated PA addresses are used in backbone routing and the size of DFZ global routing table is kept small.

B. Traffic Engineering

Traffic engineering in the current Internet is often fulfilled by injecting more-specific prefixes into the global routing table, which leads to a negative impact on routing scalability. In MILSA, a given identifier can be mapped to different locators to support multihoming. These locators may be preferred with different priority or sequence for load-balancing or load-spreading. Both end-host and the RZBS can participate in the selection of the locator. Thus, the RZBS infrastructure can easily be used for traffic engineering of incoming packet flows. Moreover, for the hierarchical locator based routing mechanism, MILSA allows selecting different routing paths in a specific level according to the inter-realm trust and other policies, which also allows scalable support for traffic engineering for different locators.

C. Mobility and Multihoming

Mobility and multihoming are discussed in detail in MILSA paper [1]. For mobility, since upper layer protocols are bound to identifiers instead of IP addresses, sessions are portable for mobile users whose locators change due to mobility. MILSA supports two models of mobility. One is a simple model with end-to-end secure locator updates. However, to support initial communication with the identifier and to allow both peers to be mobile, a global RZBS mapping system is needed. MILSA mobility performance can be improved with the help of layer 2 handover mechanisms and potential cross-layer designs. Note that the global RZBS infrastructure also helps in supporting global roaming and object delegation.

General IPv4 multihoming depends on the global routing system and has negative impact on routing scalability. IPv6 multihoming (Shim6 [5]) is end-host based and transparent to global routing system but needs support from the peer host and is not scalable. The global RZBS infrastructure makes it easier for both IPv4 and IPv6 networks, and the remote peer doesn't need to be multihoming-aware. The identifier locator split design can work closely with RZBS to support scalable multihoming, load balancing or spreading.

D. Simplified Renumbering

Renumbering is no longer costly in MILSA. When users change service providers and get different locator blocks, their identifiers remain unchanged. The renumbering will be taken

care by the global mapping system of RZBSs to rebuild the identifier to locator mappings. PI addresses are not used in the global routing system, which also makes renumbering easier. In the legacy Internet, IP addresses are also often used for packets filter, access control list, or management. In these cases, the semantics are often that of an identifier or a host instead of a locator and can be replaced by a fixed identifier in an automated fashion with less renumbering disruption.

E. Decoupling Location and Identification

In MILSA, the decoupling in network layer enables MILSA to maintain session portability in case of locator changes.

F. Routing Quality

Latency and reliability can be used to determine the routing quality. The hierarchical code based routing mechanism allows BZRs to select paths with shorter delay or better performance according to inter-realm trust or other policies which makes the topological routing more efficient. Furthermore, the hybrid design reduces the size of global routing table and decreases the packet forwarding delays. Since the edge address changes are transparent to the global routing system, the routing table updating frequency can also be reduced, which increases the routing stability. However, the first packets of a new session may suffer from the latency of the mapping system. The mapping from DNS name to HUI is done by DNS which is a proactive pull system. Since this mapping is static to some extent, a caching mechanism can help reduce this latency. The mapping from HUI to locator is fulfilled by the dedicated RZBS infrastructure (also a proactive pull system) that has predetermined location in the backbone network, which can help reduce the latency. Proactive push systems can avoid extra delays at the cost of higher state requirement by maintaining a complete mapping database at or close to the sender side. Future mapping systems with features of both types may be investigated.

G. Routing Security

Security is considered in several aspects of our design. MILSA uses DNS and the RZBS system for mapping, and AZR and BZR for packet routing. DNS is well proven to be secure in handling brutal attacks. RZBS is also transparent or invisible to the end-hosts. Inter-host trust and inter-realm trust are defined to provide end-to-end and inter-realm security to prevent potential DDoS attacks or limit them in a small scale. The participation of trust relationship and policies in deciding the optimal routing path can also reduce the potential indirection attacks. Moreover, since the edge network addresses are kept out of global routing system, it is also hard for the attackers to inject bogus mappings into the mapping system for eavesdropping, redirection, or flooding attacks.

H. Incremental Deployability

Like the Internet's original intention of interconnecting different networks of different technologies, MILSA basically is an overlay architecture to bring as few drastic changes as possible to the current core technologies, to allow step-by-step deployment, backward compatibility, and long-term evolution. As for deployment, we separate it into several gradual steps:

1. Deployment of PI-PA addresses indirection for routing scalability. In the current design, we need a PI-prefix to HUI mapping registered in DNS and a triple binding maintained in the RZBS infrastructure.
2. Deployment of the user plane identifier and locator split, end-to-end mobility and security support (may need inter-host trust public key distribution and algorithms).
3. Realm-zone assignment and management, inter-realm trust setup, and DNS name to HUI mapping registration in DNS.
4. Hierarchical locator deployment and hierarchical routing protocol deployment.
5. Secure signaling of the three planes cooperation, policies, and an integrated service model.

These steps are flexible and in the very first transitional period, we allow end-hosts to choose to support MILSA or to use the current DNS-IP two-level mapping. The deployment is also open to potential new technologies and enhancements.

V. CONCLUSIONS

In this paper, we discussed several architectural enhancements which include a hybrid architecture design, a security-enabled and logically oriented hierarchical identifier system, a three-level identifier resolution system, a new hierarchical code based locator structure and routing scheme, cooperative mechanisms among the three planes in MILSA model to assist mapping and routing, and an integrated MILSA service model. We also analyzed and discussed the underlying rationale for the design. The MILSA's answers to the IRTF RRG design goals show that the enhanced MILSA has comprehensive benefits in mobility, multihoming, routing scalability, security, and support in future service model. Furthermore, MILSA supports both short-term goals and long-term evolution by allowing incremental deployment and is open to future new technologies for enhancements.

REFERENCES

[1] Jianli Pan, Subharthi Paul, Raj Jain, Mic Bowman, "MILSA: A Mobility and Multihoming Supporting Identifier Locator Split Architecture for Next Generation Internet", IEEE GLOBECOM 2008, New Orleans, LA, December 2008, http://www.cse.wustl.edu/~jain/papers/milsa.htm

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

[3] Internet Research Task Force Routing Research Group Wiki page, 2008. http://trac.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup

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

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

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

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

[8] P. Nikander, et al, "Host Identity Indirection Infrastructure (Hi3)," in the Second Swedish National Computer Networking Workshop 2004 (SNCNW2004), Karlstad, Sweden, Nov. 2004.

[9] T. Li, "Design Goals for Scalable Internet Routing," draft-irtf-rrg-design-goals-01 (work in progress), July 2007.

[10] Raj Jain, "Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation," in Proceedings of Military Communications Conference (MILCOM 2006), Washington, DC, October 23-25, 2006, http://www.cse.wustl.edu/~jain/papers/gina.htm

[11] Subharthi Paul, Raj Jain, Jianli Pan, and Mic Bowman, “A Vision of the Next Generation Internet: A Policy Oriented View,” British Computer Society Conference on Visions of Computer Science, Sep 2008.

[12] L. Mathy, et al, “LISP-DHT: Towards a DHT to map identifiers onto locators,” draft-mathy-lisp-dht-00, February, 2008

[13] Xiaohu Xu, Dayong Guo, “Hierarchical Routing Architecture,” Proc. 4th Euro-NGI Conference on Next Generation Internetworks, Krakow, Poland, 28-30 April 2008, 7pp.