A Novel Incrementally-Deployable Multi-granularity Multihoming Framework for the Future Internet

Jianli Pan, Raj Jain, and Subharthi Paul

***Abstract*—Multihoming practice in the current Internet is limited to hosts and autonomous systems (ASs). It is "connectivity-oriented" without support for user or data multihoming. However, the swift migration of Internet from "connectivity-oriented" to "content-oriented" pattern urges to incorporate user and data level multihoming support in architecture designs instead of just through ad-hoc patches. In this paper, based on our previous research experience, we expand the multihoming concepts to both user and data levels based on the "multiple points of attachment" in a way similar to host multihoming. We propose a new incrementally-deployable multihoming framework by introducing a "realm" concept. The high-level user and data multihoming support can be built on top of the host and AS level multihoming in an incrementally-deployable and flexibly-assembled manner. Realms form a hierarchy of functionally dependable blocks. We define a new dimension of building block--*slice* which is an incrementally implementable functional unit for multihoming. Besides the long-term support for user and data multihoming, the first step deployment of the new framework is also able to address the short-term routing scalability challenge by reducing the total inter-domain routing table size gradually.**

***Index Terms*—Multihoming; Multi-granularity; User and data multihoming; Future Internet Architecture; Routing scalability**

I. Introduction

Multihoming is the key to continuous connectivity, improved throughput, fault tolerance, traffic engineering, load balancing, application-specific quality of service (QoS), and seamless mobility. Recent mobile devices have multiple interfaces. This ensures that the service is not interrupted if one of the interfaces is not available or breaks down. One can combine multiple interfaces to get higher throughput and to control what portion of the traffic goes over each interface (traffic engineering).

Multihoming usage is consistently increasing at all levels. Based on autonomous systems (AS) level data [1], the number of the multihomed stub ASs has doubled over the last 5 years and the routing prefixes announced by them in the global routing table have increased by 50%.

Multihoming is not properly supported in the current Internet. The original Internet design assumed single address for each named end-host. Each connection was between two IP addresses on a single path. Numerous patches have been developed and are being discussed. They include patches to DNS (e.g., "Google.com" resolves to different IP addresses in different locations), patches to inter-domain routing (CIDR [6], BGP [13], etc.), patches to intra-domain routing, development of new transport protocols (e.g., SCTP [4]), and so on. These solutions span multiple protocol stack layers and still do not satisfy the ever-increasing new needs. In particular, the Provider Independent (PI) addresses based host and AS multihoming have resulted in serious scalability issues [2] and the problem is expected to be worse with IPv6 deployment [3].

Moreover, the current multihoming practice is mostly characterized by multiple links or attachments to the Internet and is limited to hosts and ASs multihoming. However, the key trend driving the growth of Internet over the last decade is the profusion of content services such as Google, Facebook, and YouTube over the Internet. Cloud computing and proliferation of mobile devices have accelerated such growth. The role of data and users needs to be present and emphasized from architecture level to address the challenges. Specifically, users and data may be "virtually" attached to multiple end-hosts. For example, multiple copies of the same data may reside in multiple hosts owned by various individuals and organizations after appropriate authorization. Access to the data may be provided without interruption by this multihoming. Similarly, a user may own multiple hosts including mobile smart phone, laptop, home desktop, office desktop, etc. He may want coordination among these hosts. Thus, between the user and these hosts, there is a virtual multi-attachment relationship. In other words, the swift migration from "connectivity-oriented" to "content-oriented" pattern of Internet requires support of such high-level "multi-attachment" features in the future Internet architecture.

In this paper, we coin a new term "user and data multihoming" to present the emerging demands. We build upon our previous ID/locator split idea [1] which creates independent host ID namespace administered by a "***realm***" manager for host multihoming. The high-level user and data multihoming can be built on top of the host and AS level multihoming in an incrementally deployable and flexibly assembled manner. The basic building blocks are the Realm Hierarchy Blocks (RHBs) and their dynamic functional combination named "***slice***".

Authors are with the Department of Computer Science and Engineering, Washington University in Saint Louis, MO 63130. Email: jp10, jain, pauls @cse.wustl.edu.

This work was supported in part by a grant from Intel Corporation and NSF CISE Grant #1019119.

IEEE Global Communications Conference (Globecom) 2012, Anaheim, CA, December 3-7, 2012

Besides the long-term support for user and data multihoming, the new framework is also able to address the short-term routing scalability challenge by reducing the total inter-domain routing table size gradually.

The key contributions of the paper are that we: (1) identify and address the high-level "multi-attachment" demands from the emerging services and coin the new concepts of "user and data multihoming", (2) leverage our previous work and create an incrementally deployable framework for multihoming supports at different levels, (3) design a framework to reflect a series of new principles and goals, and (4) analyze the first step of our framework in addressing the routing scalability issue by supporting host multihoming without using PI addresses.

In the rest of the paper, we discuss some related work (Section II), the key design principles and goals (Section III), framework details (Section IV), some discussions (Section V), and conclusions (Section VI).

II. Related Work

Multihoming can be realized in different layers of the protocol stack. An example of transport layer solution is SCTP (Stream Control Transmission Protocol) [4] which needs host stack support from all the corresponding hosts and servers, and it cannot address the scalability problem. Network layer multihoming solutions include those for IPv4 and those for IPv6. IPv4 based multihoming solutions are of two types. One is Network Address Translation (NAT) multihoming [5] which avoids dependence on the routing system for multihoming but it removes the uniqueness of the IP address and hence it is difficult to support "non-client-and-server" applications. The other type generally depends on the routing system (BGP multihoming [5] by using PI addresses) and violates the CIDR (Classless Inter-Domain Routing) [6] prefixes aggregation rules, and thus, suffers from the routing scalability problem. Shim6 [7] is a host-based layer-3 multihoming solution for IPv6 via ID locator split. There are other papers [8-10] on site multihoming which also have negative impacts on routing scalability.

Other works, though not directly related to the conventional multihoming, include those trying to incorporate high-level services, users, data, and contents into the network architecture design. These works attack the deficiency of the host-based design and present alternatives. Typical ones include data-oriented network (DONA) [17], content-based networks [18], and some basic idea of human-centric networks such as Mobile People Architecture (MPP) [19-20]. However, most of them are clean-slate designs and different from the evolutionary framework in this paper. More detailed discussions of these solutions can be found in our surveys [14-15].

We build upon our prior research in several areas. The **first** is a new architectural view named Internet 3.0 [11] that allows policy-based secure communication that is aware of different organizational policies. The **second** is MILSA [1] which is a proof-of-concept ID locator split design introducing the ideas of host and infrastructure realms to solve the problems such as host mobility, routing scalability, and trust-based security in the current Internet. The **third** is the "Multi-tier diversified Future Internet Architecture" [12] which envisions a 3-tiered architecture: infrastructure tier, host tier, user and data tier. This enables building a host- user- or data- centric network.

III. Design Principles and Goals

We discuss our design principles and goals in this section.

A. Design Principles

(1) **Evolution (Not Revolution) and Coexistence (Backward Compatibility)**: Today, Internet connects billions of nodes and has millions of applications developed over the last 40 years. We believe new architecture should be designed with this reality in mind otherwise they are bound to fail. Legacy nodes and applications should be able to communicate over the new architecture without change and the new nodes and applications should be able to communicate over the existing Internet.

(2) **Incremental Deployment**: The deployment can start in a small scale and new nodes can be added incrementally. The early adopters should have economic incentives for change. The payoff will increase as the deployment of the new technology increases. Economies of scale reduce the cost and eventually the old architecture deployed base will diminish and disappear.

(3) **Organizational Control**: Organizations that own the hosts, users, data, or infrastructure will want to keep control over their resources and enforce policies about who is authorized to use those resources. In case of multihoming, the organization would want to determine how their different interfaces are used.

(4) **Location Privacy**: Location is private information and so the ID owner would want to control over who gets the location information. This is similar to the current cellular network where location is not divulged to correspondents.

B. Design Goals

(1) **Extensibility and Flexibility**: The new architecture has to be flexible and extensible.

(2) **Support for a Scalable Internet**: The architecture should not depend on the global routing protocol (e.g., PI addresses) to fulfill multihoming. The multi-granularity multihoming is achieved by new IDs rather than the IP address space.

(3) **Easiness of Developing a Prototype for Incremental Development**: We aim to follow a spiral development model. The multiple levels of multihoming are not in parallel, higher level may depend on lower level multihoming. Hence, the spiral model is appropriate for design validation and real incremental deployment.

(4) **Smooth Integration of Security, Mobility, and other Functions**: Decoupling identities from locator and introducing new ID layer and namespace not only benefits multihoming but also the other security, mobility, and inter-domain routing solutions. However, security is not a major focus of this paper.

IV. Key Concepts And Models

In this section, we present the framework details.

A. Fundamental Concepts

The first fundamental functional building unit of the framework is the "**realm**". It is derived from our prior work [1, 11, 12] and we apply here for multihoming. It is well known that in the current Internet, the IP address is overloaded as both "identifier" and "locator" [3], i.e., the single namespace serves two purposes which leads to a series of problems [3]. That is the reason we bring in the new concept of "realm" which separates the organizational relationship from the physical connectivity. "**Realm**" is defined as objects grouping together according to their common affiliation or policy. For example, all the end-hosts belonging to a single logical organization form a realm; a user has his/her user realm, and a company has its own realm. Similarly, the routing service providers (ISPs) are also organized as multiple infrastructure realms, commonly known as "autonomous systems" (AS) in the current Internet. The realms can be further classified into multiple tiers. A "**tier**" is defined as the class of realms with similar function, such as infrastructure tier, host tier, user and data tier [12].

To avoid the semantic overloading problem, realms are organized and identified by new independent ID namespaces which enable aggregatable routing locator and ID spaces, hence achieve scalability. After the logical decoupling, IDs now uniquely identify the logical entities in a specific realm, and the term "**routing**" in this framework becomes more general and represents the process that one logical entity in the realm of a certain tier finds a path to the other such entities in that tier.

Each realm has a management plane called **Realm Manager (RM)**. It is responsible for important functions such as: assignment and management of the hierarchical IDs, relationship management among its sub-realms and with the other realms, mapping retrieval/delivery/updates, and boundary traffic policy enforcements.

We further define a new term "slice" in the multihoming framework. "**Slice**" is an extensible grouping of realm hierarchy (or **Realm Hierarchy Block, RHB**s) to realize specific multihoming function. It represents the re-organization of resources for functionalities such as data or user multihoming. In other words, slices are incrementally deployable, extensible, and dynamically constructible RHBs to realize the data or user multihoming function. Fig. 1 is a simple example illustration of the concept. It consists of one slice-0 (routing slice), two slice-1s (with host, site, and enterprise multihoming) and one slice-2 (above plus user multihoming). One slice may depend on the others to realize their functionalities. For example, in Fig. 1, slice-1 depends on slice-0, and slice-2 may depend on slice-0 and 1 for high-level multihoming. Thus, to build a slice-2, at least one slice-1 is needed. By building slices incrementally, more complex multihoming functions can be achieved with parallel or extended slices. After decoupling the conflated IP semantic and creating independent ID spaces, the slice-0 becomes a pure locator-based routing system delivering packets from location A to location B and free of the scalability difficulty. Separate host ID semantic is put into the additional RHBs to form multiple slice-1s. A ***slice-1*** builds upon slice-0 incrementally and is capable of the newly added host multihoming function. Similarly, after a slice-1 is built, a ***slice-2*** can be built to support user and data multihoming. Even further levels of slices can be similarly built in this extensible framework. Note that each ellipse in the figure is a RHB which consists of a hierarchy of realms; the inner structure for each such block is similar to the hierarchical ISP realm structure of RHB-0 shown in Fig. 4.

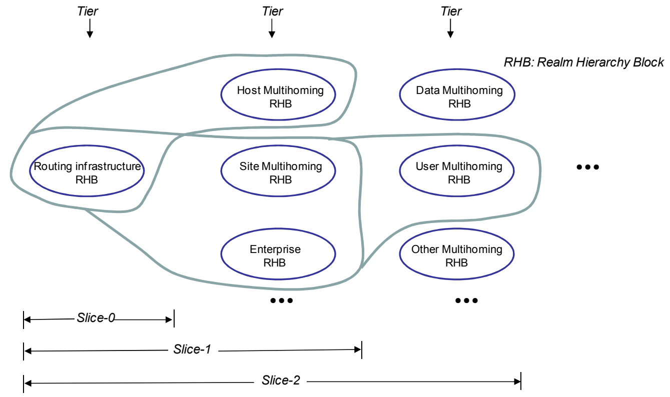

Fig. 1: Example of flexible and extensible slice structure (Slice 0, 1, and 2)

Note that the slice is different from the "layer" concept in the protocol stack since layers depict the software structure in a single networked machine, while slice describes the macro-scale structure constructed by realm blocks for a specific multihoming function. Since we inherit multiple terms from our previous work and introduce new multihoming function units, we summarize and compare them in Table I.

Table I: Comparison of terminologies

	Scope	Definition	Example
Layer	Single host protocol stack	Software stack encapsulated into logically dependent but implementation independent layers	Physical layer, link layer, network layer, transport layer, application layer
Hierarchy	Any hierarchical structure	Arrangement of items into vertically or horizontally ordered set or acyclic directed graph	Routing hierarchy, social hierarchy
Realm	Same organization or policy boundary	Objects grouped together according to common affiliation or policy	A user realm, a company realm, a department realm, a routing realm
Tier	Multiple realms	Realms with similar function	Infrastructure tier, host tier, data tier
Slice	Across multiple tiers and RHBs	Extensible grouping of realms or RHBs to realize specific multihoming function	Slice-0, slice-1, slice-2

B. Simple Host, User, and Data Multihoming Examples

1) Host Multihoming

We present a simple example on how the host multihoming works in this framework. In Fig. 2, a multihomed host (MH) is connected to ISP A and B and gets locator A and locator B, respectively. MH always monitors the links status and updates the bindings between the MH host ID and locators to the RHB-1. The changes will be automatically propagated to the host realms. The correspondent host (CH) will only know the MH's host ID and communicate with MH using its host ID. Network translates the ID to the correct locator of the MH. Thus, location privacy is maintained. RHB-1 and RHB-0 form slice-1. The multihoming policy is configured by the host realm owner represented by the RMs. Note that RHB-1 and RHB-0 have separate and independent ID spaces. Hence, achieving multihoming functions does not add any complexity in the inter-domain routing system.

Note the difference from Shim6 [7], though both use variations of ID locator split idea. Shim6 is an end-host based solution, while we intend to build a whole multihoming signaling plane containing functional realm blocks which is much more powerful than the end-host based solution.

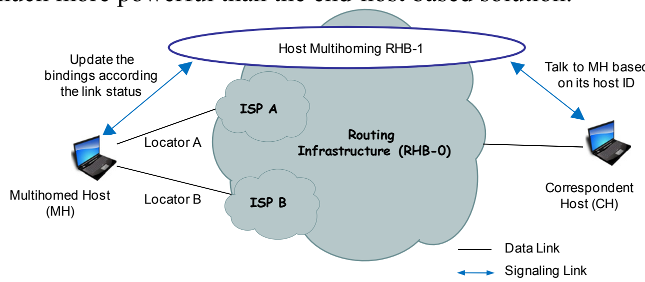

Fig. 2. A simple host multihoming example

2) User and Data Multihoming

As discussed above, the key to support user and data multihoming is to create the virtual "multi-attachment". In our framework, we achieve this goal by decoupling the ID and locator semantic and creating "realms" to manage the IDs. We generalize it to user and data realms above the host realms and create dynamic binding and mapping across different tiers just like what is achieved for the host multihoming solution [1].

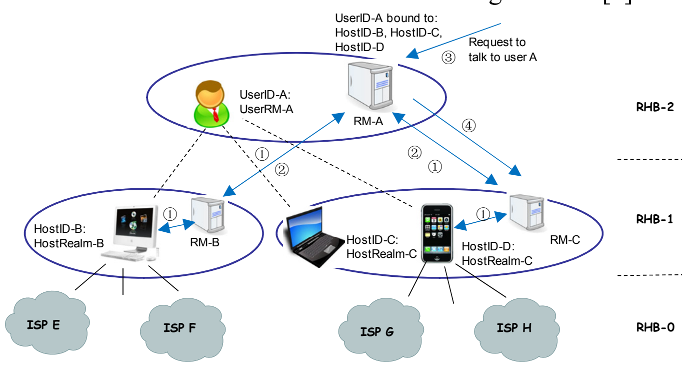

Fig. 3. A simple user multihoming example

We illustrate how the user multihoming works by simple example shown in Fig. 3. A user has three end-hosts. The desktop (host B) in his office is owned by the company and is in company host realm B, while the laptop (host C) and iPhone (host D) are the user's private properties and are in private host realms C. He uses the desktop and iPhone simultaneously for some coordinated service (e.g., Photo Stream app that pushes photos to user's all Apple devices). Each end-host belongs to a host realm and is managed by a realm manager (RM). RM-B and C manage the end-hosts in their realms and are in charge of the host multihoming function. They also talk to RM-A of the above user realm to achieve user multihoming. For step ① shown in the figure, the end-hosts update their host ID information with the local host RMs and then report to the user RM. The user RM keeps a copy of up-to-date binding information for the users and its bound hosts. In this example, the user initially uses the desktop and iPhone at the same time for some service, and then switches to use the iPhone only. As step ② shows, the user RM updates the bindings to reflect such changes. After that, whenever outside correspondents want to talk to the user, the requests will be forwarded to the iPhone directly (step ③ and ④) and the service for the user is without any interruption in case of any end-host failure or service handover initiated by the user. This example is only about user multihoming; data multihoming is similar to this procedure.

C. 3D Multihoming Framework

After the above simple example, we discuss the framework in a larger scale in Fig. 4. The 3D model consists of three dimensions: hierarchy, layer, and slice. The RHBs comprise of new realms related to each other for specific multihoming function. For example, RHB-1 comprises of host realms which help build host-to-host trust, security, or connection based relationships. RHB-1 has multihoming specific function based upon RHB-0. A simple illustration of what RHB-N means to RHB-(N-1) is like "RHB-N provides realm ID dynamic binding and mapping for RHB-(N-1) for multihoming purpose".

The model shows a slice-0, a host multihoming slice-1, and a user multihoming slice-2. It consists of 4 RHBs from left to right while potentially there can be more in the future due to the extensible design. The ellipses and circles in the RHB are the realms hierarchy and the dots insides the realm are the entities under the supervision and management of RMs represented by the triangles. The dotted circles with overlapped parts in RHB-1 and RHB-2 mean the multihomed realms in which the overlapped part represents the realms in RHB-N are bound to the two realms in RHB-(N-1) for multihoming. Fig. 4 considers only 2-homed cases but in general, there can be multiple homes. For each RHB-1 and RHB-2, the inside realms are also hierarchical as in RHB-0. Between two neighbored RHBs, there is a "RHB Coordination Agent" (RCA) which interacts with all the elements in a RHB and is responsible for the coordination with other RHBs. It also provides interfaces for interactions.

The RHB-0 in Fig. 4 illustrates two multihoming cases: (1) realm A, which is a local access realm, is connected to two different regional realms B and C; (2) customer network (stub network) realm D is connected to the two different local access realms E and F. Case (2) is called stub Network Address Translation (NAT) multihoming in survey [5]. Then we build new RHBs to enable a flexible host and user multihoming in a way that will not negatively affect others in the Internet. Thus, in RHB-1, the host realms are logically connected (they may connect to each other through logical links or through a central intermediate agent). The host realm D" represents the host realm of the customer network realm D which is multihomed to two local access realms of E and F. The overlapped dotted circles and the arrow pointing to the host realm D" represent the multihoming in RHB-1. Hence there is a hidden underlying binding between the host realm D" in RHB-1 and the realm D in RHB-0, and the binding and interaction is done through the RCA between RHB-0 and RHB-1. Host multihoming is achieved through the dynamic binding between host realm and

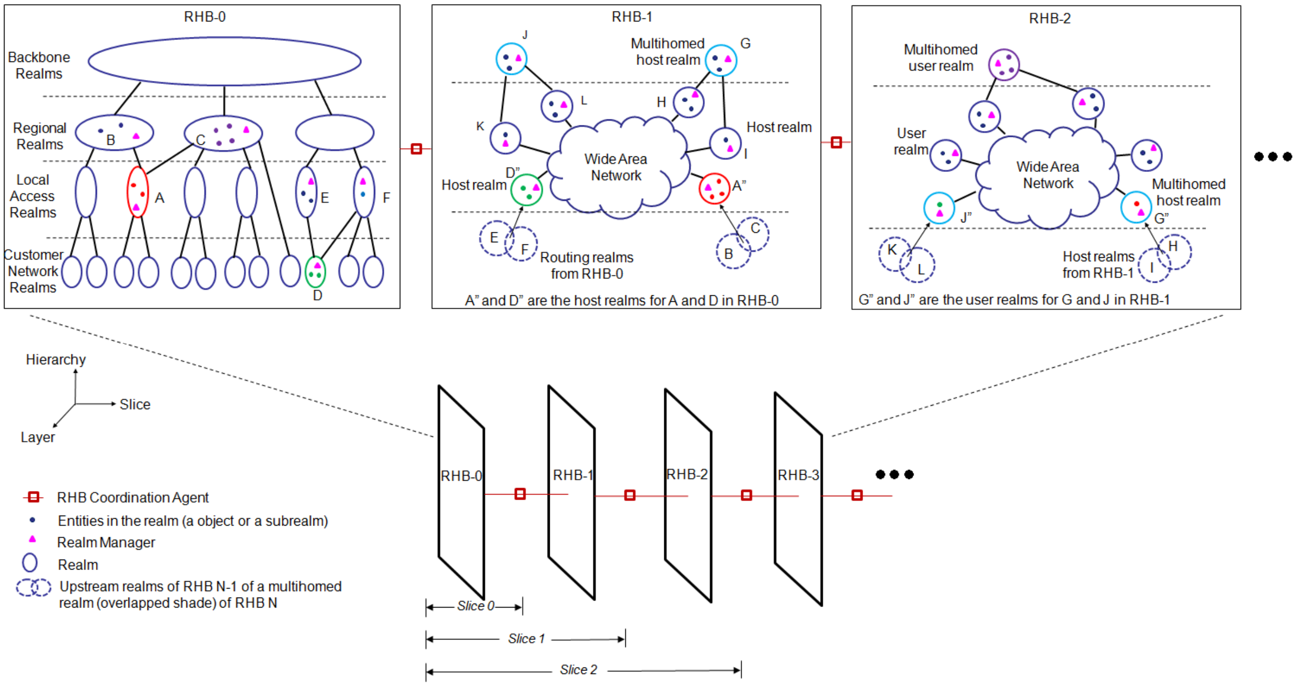

Fig. 4. An example illustration of the 3D multihoming framework

customer network realm and the cooperation among RHB-0, RHB-1, and RHB-2. Similarly for the multihomed host realm in RHB-1, there is a corresponding user realm in RHB-2 to realize the dynamic mapping and binding between RHB-1 and RHB-2. Multiple RHBs are then assembled to form the complete slices by using RCAs as the intermediate agents.

In the model, Slice-(N-1) multihoming is conceptually characterized as **realm overlapping** in the slice-N and can be achieved by dynamic mapping delivery and update in slice-N. This applies to all types of multihoming.

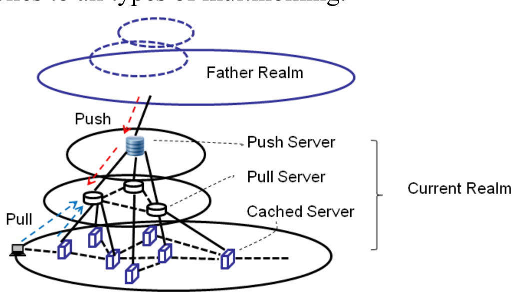

Fig. 6. Hybrid push/pull/caching mapping delivery

D. RHB Mapping and Caching Mechanism

In our framework, the dynamic binding between the ID spaces need to be retrieved and updated quickly and cost-effectively. The original DNS system was designed for static mapping and cannot handle such frequent updates. Also, for the users, they may have different demands on re-homing frequency and it is a waste to provide the same service for those who rarely change their mappings. So we have a mapping structure that provides differentiated and customizable service. We classify all the customers into several categories based on re-homing frequency. We then structure the mapping realm mangers (RMs) in a specific RHB into a hierarchy with a hybrid "push/pull/caching" mechanism. As shown in Fig. 6, the RMs with different update rates interact with each other to keep the mapping data up-to-date. Moreover, these RMs will serve with different priorities to the customers of different classes. Top mapping servers keep the most up-to-date binding by pushing from the outside. Middle mapping servers update their local copies of the bindings from time to time by pulling data from the top servers. The bottom servers have even lower update rates. The lookup speed can be very high with caching mechanism.

E. Multihoming ID Structure and Aggregation

Building extensible slices means that the newly created ID spaces are routable in the slices for ID scalability in that slice just as the current IP prefix-based routing in the Internet [6]. Currently, flat IPv4 or IPv6 addresses are used by some solutions as the EID (Endpoint ID). They lack EID aggregation in the mapping overlay. Therefore, we try new designs for the host-ID which include hybrid ID format and ID aggregation. It is an enhanced version of our prior MILSA ID [1] with tree structure combined with a hierarchical part for the logical affiliation inside the host realm and the flat cryptographic part similar to the current Host ID in the HIP [16] for security. This way, the closer the hierarchy of mapping system is to the hierarchy of the host realm, the more benefits in scalability, ease of policy enforcement and management, and host multihoming.

V. Discussions

In this section, we present some analysis and discussions.

A. First-Step Deployment Benefits

The existing multihoming practice contributes a lot to the current routing scalability issue. The first step in deploying the new architecture is to decouple host realms from routing realms and create an infrastructure of slice-0 to support future slice-1 and 2 deployments for user and data multihoming. One of the benefits of our framework is that the deployment of this first step for host and AS multihoming can alleviate the routing scalability problem by reducing the total inter-domain routing table size. We do a preliminary evaluation of this effect based on the approach we used in our previous work [1]. Currently, both the number of multihomed ASs and the prefixes they announce in the routing table are increasing approximately linearly (8.45% annual increase every year). In our evaluation, we consider two cases: first, deploy in 10% of these multihomed ASs every 6 months and finish the whole deployment in 5 years; second, deploy in 20% of these multihomed ASs every 6 months and finish the whole deployment in 2.5 years. We estimate that with the deployment, the total prefixes contributed by the multihomed stub-ASs can decrease significantly depending on various deployment speeds. Specifically, for the first case, the prefixes for the multihomed ASs decrease from around 90K to 30K in 5 years in which case the prefixes announced by them are very close to 1 prefix per multihomed AS, i.e., the low bound. The second case takes almost 2.5 years to achieve the same results.

B. *How Design Principles Are Reflected in the Framework*

(1) **Evolution and Coexistence**: The new framework is built upon the current Internet and protects the existing investment. It is different from many other clean-slate designs which ignore the practical deployment constraints for the current Internet. Due to this feature, the framework is backward compatible and the coexistence of the new and old technique and equipments will be a norm during the evolution process of the architecture.

(2) **Incremental Deployment**: The deployments of realms and slices in the new framework are incremental based on the existing Internet. The first step can be decoupling of host realm from routing realm and creating an infrastructure of slice-0 to support further slice-1 and 2 deployments for user and data multihoming. Every step of the deployment results in incentives by protecting existing investments along with providing new services without disrupting the existing services.

(3) **Organizational Control**: The incorporation of realm and RMs enables the framework to perform organizational control for data and users which is currently missing in the existing Internet architecture. The user and data level policies are no longer mixed and conflated with routing policies as in the current routing system. More high level features like security, finer-grained policy control, and content base services become possible in the new framework.

(4) **Location Privacy**: In the new framework, the locator based routing system is liberated from the above business policies. The user and data multihoming are performed by separate new ID spaces. Location information of the user is transparent to the correspondents who only know the IDs for communications.

VI. CONCLUSIONS

In this paper, we discussed a new multihoming framework for the future Internet. We broadened the multihoming concept into multiple granularities including the user and data multihoming, and tried to address the challenges through a scalable, extensible, and evolutionary 3D model. In this design, user and data multihoming can be built on top of the host multihoming through incremental slices in an evolutionary and extensible way. A preliminary evaluation shows the effects of the first-step deployment in reducing the inter-domain routing table size.

REFERENCES

[1] J. Pan, R. Jain, S. Paul, and C. So-In, “MILSA: A New Evolutionary Architecture for Scalability, Mobility, and Multihoming in the Future Internet,” IEEE Journal on Selected Areas in Communications (JSAC), Special issue on Routing Scalability, Vol. 28, Issue 8, pp. 1344-1362, October 2010.

[2] J. Abley, K. Lindqvist, E. Davies, et al., “IPv4 Multihoming Practices and Limitations,” RFC 4116, July 2005.

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

[4] R. Stewart, “Stream Control Transmission Protocol,” RFC 4960, September 2007.

[5] X. Liu, L. Xiao, "A Survey of Multihoming Technology in Stub Networks: Current Research and Open Issues," IEEE Network, Vol. 21, Issue 3, May-June 2007.

[6] V. Fuller, T. Li, “Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan,” RFC 4632, August 2006.

[7] E. Nordmark, M. Bagnulo, “Shim6: level 3 multihoming Shim protocol for IPv6,” RFC 5533, June 2009.

[8] C. de Launois, B. Quoitin, and O. Bonaventure, “Leveraging Network Performances with IPv6 Multihoming and Multiple Provider-Dependent Aggregatable Prefixes,” Elsevier Computer Networks, Vol. 50, June 2006.

[9] R. Atkinson, S. Bhatti, and S. Hailes, “A Proposal for Unifying Mobility with Multi-Homing, NAT, & Security,” ACM MobiWac, Greece, 2007.

[10] T. Bates, Y. Rekhter, “Scalable Support for Multihomed Multi-provider Connectivity,” RFC 2260, January 1998.

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

[12] S. Paul, R. Jain, J. Pan, "Multi-Tier Diversified Architecture for the Next Generation Internet," Proceedings of International Conference on Cloud Computing and Virtualization (CCV 2010), Singapore, May 17-18, 2010.

[13] Y. Rekhter, T. Li, S. Hares, “A Border Gateway Protocol 4 (BGP-4),” RFC 4271, January 2006.

[14] J. Pan, S. Paul, and R. Jain, "A Survey of Research on Future Internet Architectures," IEEE Communications Magazine, Vol. 49, No. 7, July 2011, pp. 26-36.

[15] S. Paul, J. Pan, R. Jain, "Architectures for the Future Networks and the Next Generation Internet: A Survey," Computer Communications, UK, Volume 34, Issue 1, 15 January 2011, pp. 2-42

[16] R. Moskowitz, P. Nikander and P. Jokela, “Host Identity Protocol (HIP) Architecture,” RFC 4423, May 2006.

[17] T. Koponen, M. Chawla, B.-G. Chun, et al., "A Data-Oriented (and Beyond) Network Architecture," ACM SIGCOMM 2007, Kyoto, Japan, August 2007.

[18] V. Jacobson, D.K. Smetters, et al., "Networking Named Content," in Proceedings of CoNEXT 2009, Rome, Italy, December 1-4, 2009.

[19] P. Maniatis, M. Roussopoulos, E. Swierk, et al., "The Mobile People Architecture," ACM Mobile Computing and Communications Review (MC2R), Vol. 3, No. 3, July 1999.

[20] M. Roussopoulos, P. Maniatis, E. Swierk, et al., "Person-Level Routing in the Mobile People Architecture." USENIX Symposium on Internet Technologies and Systems, October 1999.