A Future Internet Architecture Based on De-conflated Identities

Subharthi Paul, Jianli Pan, *Student Members, IEEE,* Raj Jain, *Fellow IEEE*
Department of Computer Science and Engineering
Washington University in Saint Louis
{spaul, jain, jp10}@cse.wustl.edu

ABSTRACT: We present a new Internet architecture based on de-conflated identities (ADI) that explicitly establishes the separation of ownership of hosts from the underlying infrastructure connectivity. A direct impact of this de-conflated Internet architecture is the ability to express organizational policies separately and thus more naturally, from the underlying infrastructure routing policies. Host or organizational accountability is separated from the infrastructure accountability, laying the foundations of a cleaner security and policy enforcement framework. Also, it addresses the present Internet routing problems of mobility, multihoming, and traffic engineering more naturally by making a clear distinction of host and infrastructure responsibilities and thus defining these functions as a set of primitives governed by individual policies.

In this paper, we instantiate the primitive mechanisms related to the issues of end-to-end policy enforcements, mobility, multihoming, traffic engineering, etc., within the context of our architecture to emphasize the relevance of a de-conflated Internet architecture on these functions.

Keywords

De-conflated Identities, Future Networks, Next Generation Internet, Internet Architecture, Multi-homing, Mobility, Routing, Scalability, Traffic Engineering

I. INTRODUCTION

A. PROBLEM BACKGROUND: The current Internet architecture suffers from the problem of conflated identities. While the underlying "communication system" of the Internet evolved over the extremely flexible and resilient packet switching primitive, its "communication paradigm" was designed along the same conversational, unicast model of the circuit switched telephone network. Within this "communication paradigm", a communication instance is completely specified by the connectivity between two service interfaces within the "communication system," which in the case of the Internet is the physical packet-switched path between the source and destination IP addresses. The context in which the Internet was originally designed did not warrant the need to treat the "communication paradigm" separately from the "communication system." Eventually, in its commercial avatar, the Internet overlaid its "communication system" with a multi-ownership policy-enforcement model. However, as a result of the original assumption, these policies of the "communication system" were naively entwined with the "communication paradigm." To emphasize this point more concretely, consider enterprise networks. An enterprise network is generally a "stub network," that is, it is either the source or destination of data traffic. The policies of an enterprise in a communication paradigm is mostly pertaining to its data, users and hosts. However, the lack of an explicit representation of the communication paradigm, forces host, user and data policies to be "unnaturally" enforced by masking them over the communication system policies that mostly dictate connectivity. This crude generalization in the basic underlying model of the Internet is responsible for a gamut of problems ranging from the complexity of enabling host, user and data mobility, multi-homing, multicasting and anycasting dissemination paradigms, security and policy enforcements at the required level of granularity, etc.

The problem is more relevant today than ever before in the "tiered-context" of the present communication paradigm. An extremely likely scenario representing this "tiered-context" is a distributed enterprise application hosted on compute resources leased from multiple public cloud computing platforms, which are connected over the Internet through multiple ISPs. This highly dynamic and diverse context clearly motivates the need for de-conflating identities and establishing separate ownerships of data, host, users and infrastructures. This shall allow a more natural representation of the present communication paradigm. The added complexity of the proposed architecture may be justified on the basis that externally overlaid incremental (often architecturally ugly) mechanisms introduce inconsistencies and non-determinism into the overall architecture. Also, these mechanisms are severely restricted in their effectiveness owing to inherent constraints imposed by the original design. The Internet is rife with instances of such tussles, be it between P2P providers and ISPs, NAT mechanisms and end-to-end protocols, policy control mechanisms and security mechanisms, underlay routing policies and overlay routing requirements, etc. The non-determinism manifests in the fact that a new internet-wide standardized mechanism can no longer be guaranteed to perform, as determined, across the whole system and also it could potentially break some of the existing mechanisms.

B. REFERENCE ARCHITECTURE: We propose a new Internet architecture based on the de-conflation of identities. Our architecture is based on a "*three-tier object model*" (Figure 1). The bottom tier consists of a high-speed network infrastructure owned by multiple ISPs. The second tier consists of hosts owned by different organizations such as DoE, DARPA, Amazon, etc. The third tier consists of users and data objects.

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

Proceedings of IEEE Globecom 2010, Miami, FL, December 6-10, 2010.

The “*objects”* in the different tiers are defined in the context of “owner” realms. The “realms” represent an administrative and/or ownership domain. Each realm has a logical entity called the Realm Manager (RM). The realm manager is responsible for explicitly defining the ownership model of our architecture and is responsible for policy enforcement of all objects belonging to the realm. Also, realm managers are responsible for inter-realm negotiations for a complex communication context involving multiple realms. A more detailed description of the three-tier object model and the underlying policy-oriented network architecture can be referred to in [10].

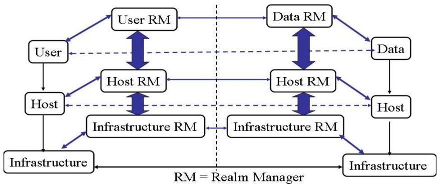

Figure 1. Three-Tier Object model

C. SPECIFIC FOCUS: In the present paper we limit our discussion to the bottom two tiers of the three-tier object model, focusing on problems related to the interaction between the host and infrastructure tiers. Discussion on host-data tier interactions that enable a gamut of extremely interesting dissemination and service deployment scenarios are deferred to future work.

It is interesting to note that de-conflating identities in the bottom two tiers (infrastructure and host) is closely aligned to the research on Identifier/Locator split architectures being actively pursued at the IRTF [5]. However, unlike ID/Locator split proposals, the central design point of our architecture is to realize an explicit separation of ownership between hosts and infrastructure, to cater to a cleaner policy enforcement interface. The goal of this paper is to show that this separation also provides a natural solution to routing problems of scalability, multihoming, mobility and traffic engineering through a set of primitives governed by individual object realm policies.

II. ADI CONCEPTUAL OVERVIEW

In this section we define some key concepts underlying the ADI design.

2.1. ADI Definitions: Identifiers and Realm Managers: An ADI host may belong to multiple isolated logical contexts or *host realms*. These separate logical contexts, each with their own set of policies are instantiated within the host as *host objects.* Each host object has an ID (HID) which is organized as <Host Realm ID (HRID), Host Object ID (HOID))>. HRID is globally unique and may have several hierarchical levels to represent logical context boundaries for policy enforcements. HOID is local within the context of the HRID.

Similarly, infrastructure realms (IR) can advertise their capabilities as multiple logical infrastructure objects. Each logical infrastructure object is mapped to a common shared physical substrate. Logical infrastructure objects too are assigned IDs of the form <Infrastructure Realm ID (IRID), Infrastructure Object ID (IOID)>. IRID represents the hierarchical relationships of infrastructure realms. IOIDs are local to the network site advertising the object.

“Realm manager” in ADI is a logical aggregation of functions required to manage the objects within a specific realm. In general these logical functions provide specific services to the objects within the realm and also enforce organizational policies on each object of the realm. We shall discuss some of these functions in the context of the ADI design.

2.2. Management and Control Planes: The realm managers in each tier participate in the management and control plane to implement distributed management and control functions pertaining to that tier. Examples of these functions include routing, policy and security negotiations, setting up trust chains, authorization and authentication mechanisms, etc. Also, these management and control functions shall allow inter-tier interactions to aid optimized cross-tier functions such as cross-tier ID mappings, multi-homing, QoS mapping, tiered service interfaces, multi-tier service integration for diversified networking contexts, etc. There are broadly three types of inter-tier interactions: A) **INFORMATIONAL:** where the tiers request and exchange tier-specific information with the other tiers, B) **POLICY:** where the tiers negotiate tier-based policies for building a cross tier function, and C) **FUNCTIONAL:** where a function in one tier is dependant on some function(s) in some other tier(s).

However, the scope of the present paper is limited to discussions of only the key functions that implement the most basic communication model, within the purview of the ADI architectural framework.

2.3 ADI HID Layer: All host based services connect using host IDs. Hence, for performing these services, ADI introduces a new *ADI HID Layer* between the transport layer and the network layer of the current host protocol stack.

Thus, the introduction of HID layer serves two primary purposes: 1) It implements the end-host responsibilities of the distributed host realm functions, and 2) It de-couples the concerns of a logical end-to-end connectivity over host realms from the concerns of physical end-to-end connectivity over infrastructure realms.

The separation of logical connectivity/physical connectivity concerns has huge implications on the flexibility and functionality of the Internet. In the context of ADI, logical connections over immutable host IDs are shielded from Infrastructure ID changes as a result of host mobility over multiple infrastructure domains or due to specific types of multi-homing solutions. Also, a logical link can accrue attributes of security, trust and reliability through inter-realm negotiations during connection-setup.

Please note that the dynamic binding of HID-IID is implemented at the HID layer to ensure interoperability with legacy applications that are implemented over transport protocols that statically bind a communication session with routing identifiers such as IP addresses. The dynamic binding capability is therefore moved to the HID layer and remains transparent to legacy transport protocols. However, newer transport protocols could be designed to implement dynamic binding capabilities within them. We, however do not suggest such designs as it interferes with the end-to-end definition of transport protocol functions and might introduce newer complexities. Instead, we suggest a "dynamic-binding" aware transport protocol which can communicate directly with the HID layer and allow transport mechanisms to be aware of dynamic-binding functions in a lower protocol layer. Such a transport protocol shall be able to share end-to-end path condition information directly with the HID layer, allowing the HID layer to implement its functions without having to "snoop" on transport protocol packets that it currently does.

2.4 Infrastructure Realms: The infrastructure realm (IR) is primarily responsible for the distributed "routing function". For this, they distribute reachability data amongst themselves and compute forwarding tables. The IR objects include IR border routers and IR core routers. Similar to the host realm manager functionality, the IR manager functions are distributed across these objects. At the host-side, the network layer of the host protocol stack is responsible for the responsibilities specific to the IR.

The concept of infrastructure realms is synonymous to autonomous systems (AS) in the current Internet with respect to the fact that both represent a single administrative and management domain. Also, the associated routing function is synonymous to and the present IPv4 based routing system. However, as shall be seen in Section 3, explicit ownership and policy enforcement framework of ADI allows IRs to better express and enforce policies across their domains and thus implement IR specific functions more naturally and efficiently. Also, unlike the legacy IPv4 routing system, <IRID.IOID> assigned to ADI hosts in the context of IRs no longer need to be contextually overloaded to serve as 1) session identifier for TCP sessions, 2) host identifiers for host based policy enforcements and 3) locators in the global routing tables, all at the same time. As a result, the IOIDs in the context of IRIDs attribute to more scalable, dynamic and efficient routing function.

2.5 ADI Host ID (HID): ADI host IDs (Figure 2) are 128 bit binary strings. The choice of its size is, by design, made to resemble IPv6 addresses for the purposes of injecting minimal changes to application layer and transport layer interfaces. Most implementations of transport layer protocols and most modern day applications are already IPv6 aware. The choice of 128 bit host IDs reduce the burden of re-defining many of the upper layer protocol interfaces.

Other than its size, the ADI host ID does not resemble IPv6 addresses in any way. ADI host IDs need to be globally unique. As discussed in Section 2.1, the scope and purpose of host IDs is to enumerate logical organizational structure. The hierarchical ID structure should aid the formation of such efficient overlay networks. Apart, from these, the host ID establishes the identity of a host in the context of its host realm. This identity may be used for the purpose of authentication and authorization of the host. This adds the requirement on host IDs to be secure.

The ADI Host ID design has been made keeping in mind all the above requirements. The 128 bit binary string is overlayed with 'n' bit Host Realm ID (HRID) and '128-n' bit local host ID. The issue of whether 'n' shall be fixed to a constant number or whether ADI shall use ideas from CIDR-like prefixes and variable length masking remains open to future research.

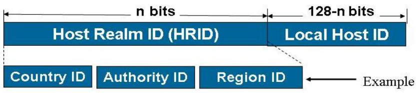

Figure 2. ADI Host ID (HID)

The HRID aids the representation of the organizational ownership of the host realm and the local host ID part is a flat '128-n' bit cryptographic hash (of the host realm ID and the public key of the host) to fulfill the requirements of a secure ID. The HRID may be further be partitioned into a hierarchical geographical and organizational structure such that it helps in the formation of an efficient overlay of the host realm managers implementing the ID management plane. It must be carefully noted that data plane is completely oblivious to the hierarchical structure of the HRID. The HRID aids locating the service access points in the ID management plane for functions such as ID-locator mapping, security authentication, authorization, policy enforcements etc. The global uniqueness of the ADI HID should be guaranteed through some registration mechanism. Since the HRID is globally unique and controlled by the ID registration and administrative authorities from different countries, the second part of the ADI HID, the hash value just needs to be unique within the host realm scope. The purpose of the hierarchical host ID in ADI is to ease the management of the global ID namespace and hold the economic and trust model in the ID/locator mapping system. In short, we gain several advantages through this concatenation structure of ADI ID.

2.6 ADI Infrastructure ID (IID): The ADI IID is 128 bits long. As seen in Figure 3, the first part of 96 bits IRID can be used to globally uniquely identify each IR and it serves as a /96 IPv6 prefix. The IRID has a hierarchical structure for topological aggregation, inter-infrastructure realm relations and routing flexibility. The second part of 32 bits IOID is an IPv4 address and each IR adopts independent local IPv4 address space. The IOID is not globally unique, however, <IRID, IOID> is globally unique. This locator design helps ease the renumbering process. Moreover, the first part of IR ID is used to perform the global routing and the second part is used to accomplish local routing in a specific IR.

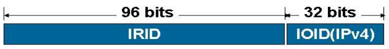

Figure 3. ADI Infrastructure ID (IID)

III. ADI OPERATIONS

Having presented an overview of the ADI design elements, we discuss the operational details of the ADI functions in this section.

3.1 ADI Operation Overview: Figure 4 shows an overview of ADI operations. It is a two step process for end-to-end connection establishment.

Step 1: Assuming the source host has learnt the HID of the destination host, the source host resolves the destination HID to the destination IID through the "mapping and negotiation function".

Step 2: Having learnt the destination IID and having completed initial inter-realm negotiations between the source and destination host realm managers, the "routing function" is responsible for routing the packets between the source and the destination.

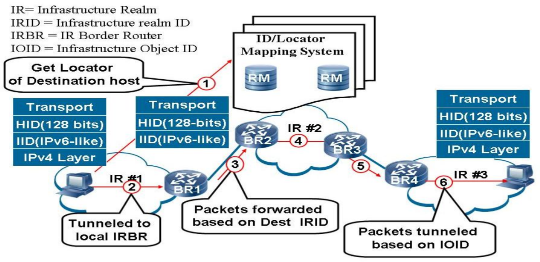

Figure 4. ADI Operations – Overview

The "mapping and negotiation function" is in accordance with our idea of realms (Section 2.1) and unlike most other proposals do not employ DHT(on flat IDs) or DNS mechanisms. DNS-like mechanisms though highly scalable hurt the dynamicity of the mapping information and DHT (on flat IDs) mechanisms violate organizational structure. However, hierarchical DHT mechanisms [14] may be used within hierarchical boundaries of the Host ID, such that they preserve organizational structure.

The proposed architecture needs to be incrementally deployable over the existing system. This is the primary reason for choosing 128-bit HIDs and IIDs. 128 bit HID allows interoperability with IPv6 based applications while the 128-bit IID achieves interoperability with the IPv6 based Internet routing plane. Here we have assumed that most modern applications have portability over IPv6. However, it would be preposterous to make the same assumption about IPv6 routing. Thus, the routing plane in ADI is designed as an IPv6-like overlay over IIDs while the underlying network predominantly remains IPv4-based. Hence, as shown in Figure 4, the core routers for a particular transit network may still be IPv4. IPv6 over IPv4 tunnels are employed to solve this problem of interoperability. Hence in Figure 4, steps 2, 4, 6 may have to employ tunneling if the core routers of IR 1, IR 2, and IR 3 respectively, are not IPv6 enabled. Steps 3 and 5, however, do not need to employ any tunnels as they represent communication between two IPv6 enabled border routers (IRBRs).

3.2 The Mapping and Negotiation Function: The "mapping function" involves the following control plane sub-functions: 1) MAP BUILD: builds the distributed mapping table for HID-to IID mappings, and 2) MAP RESOLVE: resolves a mapping query control message by routing it over the host tier control plane, MAP UPDATE: renews/updates HID-IID bindings in the distributed mapping table. The MAP BUILD function involves building a hierarchical DHT-based mapping table over the participating host realm mapping servers. The MAP RESOLVE function involves setting up an overlay forwarding plane over the host realm mapping servers based on the hierarchical HRID part of the HID. The MAP UPDATE function involves registration control message routing over the MAP RESOLVE function and synchronized concurrent update of the HID-IID bindings in the distributed mapping table. It must be noted that the MAP BUILD and MAP UPDATE functions involve cross tier *informational* and *policy* interactions whereas the overlay forwarding mechanism of MAP RESOLVE requires both *informational* and *functional* interactions between the host and infrastructure tiers.

The "negotiation function" implements, both 1) inter-realm, intra-tier negotiations for policy enforcements, security, and other parametric exchanges for various control and management functions, and 2) inter-realm, inter-tier negotiations that has already been classified under *policy* interactions to aid optimized cross-tier functions as discussed in Section 2.2.

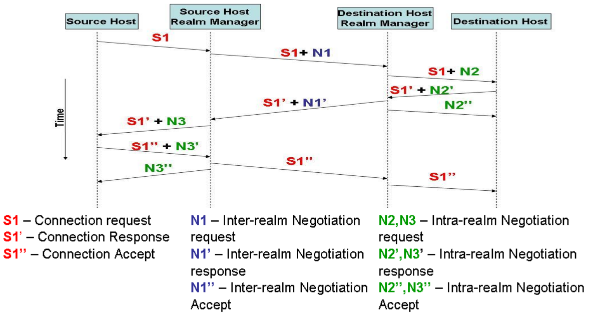
Source Host
Source Host Realm Manager
Destination Host Realm Manager
Destination Host
Time
S1
S1+ N1
S1+ N2
S1' + N2'
S1' + N1'
N2''
S1' + N3
S1'' + N3'
S1''
N3''
S1''
S1 – Connection request
S1' – Connection Response
S1'' – Connection Accept
N1 – Inter-realm Negotiation request
N1' – Inter-realm Negotiation response
N1'' – Inter-realm Negotiation Accept
N2,N3 – Intra-realm Negotiation request
N2',N3' – Intra-realm Negotiation response
N2'',N3'' – Intra-realm Negotiation Accept

Figure 5. ADI End-to-End Connection Setup (demonstrating inter-realm, intra tier policy exchanges in the host tier)

Figure 5 shows the sequence of control message exchanges in the host realm (inter-realm, intra-tier) for end-to-end connection setup. It involves exchanges of 2 types of packets: 1. S-type: "end-to-end connection request/response packet" exchanged between the source host and the destination host. These packets consist of host-host connection parameters. 2. N-type: "Negotiation request/response packet" exchanged between 1) host and host realm manager, and 2) source host realm manager- destination host realm manger. These are XML–like packets for parameter based negotiations facilitating security, authorization, authentication and other policy related negotiations. At the end of the connection process, the end-hosts share common end-to-end connection parameters and part of inter-realm negotiation parameters that are applicable to them.

3.3 The Global Routing Function: The global routing function is an inter-infrastructure realm control plane function. The routing function of ADI is very similar to the current Internet routing in that both setup the distributed forwarding table state in the data plane. However, there are some basic differences, *Firstly*, the routing function of ADI is implemented around hierarchical and topologically aggregatable IIDs which allow higher scalability owing to smaller forwarding state. *Secondly*, the IRID hierarchy explicitly encodes inter-infrastructure realm relationships. This sets up an explicit policy framework that shall allow more flexible routing paradigms such as source routing, multipath routing etc. to evolve more naturally. *Thirdly*, the present BGP-based inter-domain routing enforces transit policies across an AS by selectively advertising its connectivity information through route export policies [12]. This "ugly" policy enforcement mechanism has several implications. One such implication is that this mechanism resists the introduction of topologically aggregated locators (IP addresses) since they implicitly divulge topological inter-connectivity. In ADI, the explicit policy framework together with the explicit encoding of inter IR -relationship into the IRIDs decouples policy from being expressed in as a function of the control plane state (in this case forwarding tables and export rules). This shall allow a much cleaner and more expressive implementation of inter-realm policies in the infrastructure tier, thus allowing a topologically aggregatable routing function. *Lastly*, *informational* cross-tier interactions shall allow better monitoring and measurement tools, laying the bases for an "accountable network architecture" as discussed in [2]. Similarly, *policy* cross-tier interactions allow efficient implementation of routing resilience mechanisms through multihoming, multipath, etc.

3.4 ADI Site Multi-Homing and Routing Scalability: There are two related issues: 1) **Routing Scalability:** A provider independent (PI) locator site connected to more than one transit provider networks and having each transit provider advertise the PI prefix reachability from itself to the global routing function is the most natural form of multi-homing, and one that is commonly seen in today's Internet using IPv4 PI addresses. However, as the number of stub-sites (infrastructure realms that do not provide transit services to the global routing function) wanting to implement multi-homing using PI locators increases, the routing function is faced with a scalability problem. Each PI locator adds an entry to the global routing table and the Routing Information Base (RIB) exchange messages.2) **Tussle in Host-based vs Site-based Multihoming control:** This tussle results from the fact that the network site uses multihoming for local link redundancy and traffic engineering while hosts use it to identify better end-to-end paths through the end-to-end path diversity offered by multihoming. Both these uses of multihoming are well justified. Present BGP-based inter-domain routing is designed for scalability and thus compromises on adaptability. This results in long convergence time [6] (often in the order of minutes) in BGP to be able to route around failures. Also, BGP does not implement explicit congestion signaling to optimally route traffic around network hot-spots. Thus, multihoming in the infrastructure tier is mostly used for redundancy against local access link failure or for traffic engineering to optimize transit costs, avoid local access link congestion, engineer inter-domain flows uniformly among site core and border routers, etc. On the other hand, end-host transport protocols implement mechanisms to predict end-to-end path problems through their flow and congestion control methods and are thus better suited to react to end-to-end path problems. However, hosts don't have any control over (and mostly oblivious to) site traffic engineering policies.

In ADI, the infrastructure realm is separated from the host realm. This allows an end-to-end communication session to bind to immutable HIDs while the underlying HID-IID binding could change dnamicaly.This property, which is a direct connotation of all ID/locator split mechanisms, is especially important with respect to multihoming since this allows us to solve the "routing scalability" issue associated with multi-homing. Explicit separation of HIDs from IIDs allow IIDs to evolve along the topological hierarchy representing Internet connectivity. An HID may be associated with multiple IIDs. The IRID of each IID is allocated from the IRID prefix allocated to each provider IR. The IOID of each IID is same and is used for the intra-infrastructure realm routing function. Dynamic mapping of HID to any IID allows the same (actually more [11]) flexibility with regards to multihoming as compared to PI based multihoming. Additionally, it allows better aggregation of routing identifiers thus contributing to the scalability of the global routing function.

The next issue of tussle in host-based versus site-based multihoming control is resolved through *informational* and *policy* based inter-tier interactions between the host tier and the infrastructure tier. The host-tier "informs" the infrastructure tier of end-to-end path problems and might in turn be informed about the local link availability of the different multihoming links of the site together with policy connotations of using each. Also, the infrastructure tier informs its traffic engineering policies to the host tier through the *information* channel. The *policy* channel is then used to negotiate *path switching* through multihoming route control between the host and infrastructure tiers. [1] and [11] analyze the effectiveness of path-switching based on path properties in terms of path quality and path diversity (availability of alternate paths, low correlation of failure of alternate paths etc) over the actual topological diversity of the Internet. The results of the analysis establish that edge diversity through multihoming expose significant diversity in end-to-end Internet paths. Details of the ADI policy based, host-infrastructure tier cooperative multihoming route control can be referred to [11].

3.5 ADI Stub-site Traffic Engineering: Traffic engineering is mostly an infrastructure-tier function. However, traffic engineering policies often clash with host-tier functions such as source routing, host-initiated multihoming route control, etc. Again the *policy* and *information* cross tier interactions help resolve these clashes.

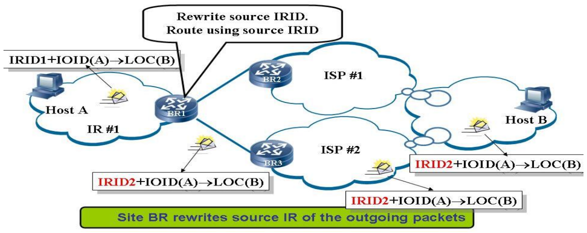

Figure 6. ADI- Site Traffic Engineering

The infrastructure-tier functional part of traffic engineering in ADI is implemented, as shown in Figure 6, through the IR border router (IRBR) re-writing the IRID part of IID on packet headers and forwarding them to the upstream IR that manages this IRID space. HIDs protect end-to-end connections against IID rewriting, similar to multi-homing. This mechanism allows the stub-IR to exert both, outbound as well as inbound traffic engineering. Inbound traffic engineering is ensured because the global routing function aligns the forwarding function in the data path to the IRID hierarchy.

V. RELATED WORK

The initial inspiration for designing an Internet architecture based on de-conflated identities came from the "Layered Naming Architecture (LNA)" proposal in [4]. While LNA is motivated to establish the functional independence of the layered model of Internet communication paradigm, ADI proposes a tiered architecture representing the communication paradigm that explicitly expresses the multi-ownership policy framework of a communication context. ADI is motivated by the model of service integration to define diverse application contexts within a clean and explicit policy negotiation and enforcement framework.

The discussions in this paper are particularly close to discussions on ID/locator split ideas and other proposals to address the problems of mobility, multi-homing, traffic engineering, routing scalability, etc. being actively pursued by various groups at RRG[5] in IRTF. However, ADI differs significantly from these proposals such as LISP[7], HIP[8], ISATAP[13], RANGI[14], Shim6[9], Six/One[15], etc, in that it tries motivate the need to consider ownership and policy control issues together with purely functional issues of these mechanisms. We contend that merely technical excellence of a contextually motivated solution does not ensure its fitness into the synergy of a diverse system such as the Internet. The Internet is rife with examples of such specific contextually motivated solutions that have introduced indeterminism and inconsistency into the overall architecture.

Finally, we are motivated to define an architecture that prevents the future Internet from the impasse of ossification [3] that it is currently faced with. ADI is designed to provide a broad framework wherein each context can express its requirements through a set of architectural abstractions and implemented over a set of basic policy primitives.

VI. SUMMARY

In this paper we presented an Internet architecture based on de-conflated identities and discussed the issues of routing scalability, multihoming and traffic engineering within the context of the proposed architectural design. Our architecture on de-conflated identities has a much larger context than presented in this paper. We limit our discussion specifically to an instance of infrastructure and host tier interactions to motivate a solution to the immediately relevant problems of routing.

ACKNOWLEDGEMENT

Authors would like to thank Xiaohu Xu for our initial joint work on RANGI [14].

REFERENCES

[1]A. Akella, J. Pang, B. Maggs, S. Seshan, A. Shaikh., "A Comparison of Overlay Routing and Multihoming Route Control" ACM SIGCOMM, *Portland, OR,* 2004

[2] D. G. Andersen , H. Balakrishnan , N. Feamster , T. Koponen , D. Moon , S. Shenker, "Accountable internet protocol (aip)," ACM SIGCOMM Computer Communication Review, v.38 n.4, October 2008

[3] T. Anderson., L. Peterson., S. Shenker, J. Turner, "Overcoming the Internet Impasse through Virtualization," Computer 38, 4, Apr. 2005, 34-41.

[4]H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S.Shenker, I. Stoica,, M. Walfish, "A Layered Naming Architecture for the Internet," SIGCOMM, Portland, OR, USA, 2004.

[5] Internet Research Task Force Routing Research Group Wiki page, 2008

[6] C. Labovitz, A. Ahuja, A. Bose, F. Jahanian, "Delayed internet routing convergence," Proc. ACM SIGCOMM, Stockholm, Sweden, 2000, pp. 175–187.

[7] D. Meyer, "The locator identity separation protocol (LISP)," The Internet Protocol Journal, Vol. 11, No. 1, pp. 23, 2008

[8] R. Moskowitz, P. Nikander, "Host Identity Protocol (HIP) Architecture," RFC 4423, May 2006.

[9] E. Nordmark, M. Bagnulo, Shim6: Level 3 Multihoming Shim Protocol for IPv6 , RFC5533, June,2009.

[10] S. Paul, R. Jain, J. Pan, and M. Bowman, "A Vision of the Next Generation Internet: A Policy Oriented View," British Computer Society conference on Visions of Computer Science, September 2008.

[11] S. Paul, R. Jain, J. Pan, "An Identifier/Locator Split Architecture for Exploring Path Diversity through Site Multi-homing - A Hybrid Host-Network Cooperative Approach," Proc. of IEEE ICC 2010, Cape Town, South Africa, May 23-27, 2010. http://www.cse.wustl.edu/~jain/papers/multihom.htm

[12] Y. Rekhter, T. Li, S.Hares, Editors, A Border Gateway Protocol 4 (BGP-4), RFC4271, Januay, 2006

[13] F. Templin, T. Gleeson, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)," RFC 5214, March, 2008.

[14] Xiaohu Xu, Dayong Guo, Raj Jain, Jianli Pan, Subharthi Paul, "RANGI: Routing Architecture for Next Generation Internet," Presentation to Routing Research Group (RRG), Internet Research Task Force, meeting, Minneapolis, MN, November 21, 2008,

[15] Vogt, C., "Six/one router: a scalable and backwards compatible solution for provider-independent addressing," Proc. of the 3rd international Workshop on Mobility in the Evolving internet Architecture (Seattle, WA, USA, August 22 - 22, 2008). MobiArch '08. ACM, New York, NY, 13-18.