

A Less-Is-More Architecture (LIMA) for a Future Internet

J. Li*, M. Veeraraghavan*, M. Reisslein†, M. Manley*, R.D. Williams*, P. Amer‡ and J. Leighton‡

*University of Virginia, Charlottesville, VA, {jl3yh,mv5g,mcm5na,rdw}@virginia.edu

†Arizona State University, Tempe, AZ, reisslein@asu.edu

‡University of Delaware, Newark, DE, {amer,leighton}@cis.udel.edu

Abstract—A new addressing and routing design called the Less-Is-More Architecture (LIMA) is proposed as an inter-domain solution for a future Internet. Unlike recently proposed identifier-locator split solutions, LIMA uses just (topological) location-independent names and location-dependent addresses. The feasibility of using a policy combination of restricting stubs to provider-aggregatable addressing only, and disallowing stub-level reachability from being propagated into the global routing tables, is studied. This policy combination results in significantly smaller global routing tables but creates four challenges of address renumbering (when stubs change providers), multihoming, mobility, and traffic engineering. Solutions to these challenges include the use of multiaddressing, name based sockets, a LIMA concept of address dismemberment, transport protocols such as SCTP that are capable of dynamic address reconfiguration, and new management-plane and control-plane procedures. Preliminary RIB data analysis quantify the benefit of LIMA in global routing table size reduction (to 6815 entries from today’s 335K entries), and a cost of LIMA in terms of number of provider changes made by stubs in the last six months (about 2450 provider changes per month across 33K stubs).

I. INTRODUCTION

A. Background

Global routing tables maintained by Tier-1 ISPs, also referred to as the Default-Free Zone (DFZ), have been “growing at an increasing and potentially alarming rate,” per a 2007 Internet Architecture Board (IAB) Workshop report [1]. A contrarian view is espoused in [2], which states that the 17% exponential yearly growth rate is in step with improvements in memory technologies. The above argument notwithstanding, in the current environment of scaling back on resource consumption, whether for energy savings, or more broadly, sustainability, new designs for a future Internet of Things should keep the capital expenditures (router memory costs), and operating (administrative and power) expenditures low.

B. Hypothesis

The *hypothesis* of our proposed Less-Is-More Architecture (LIMA) is that it is feasible to adopt (i) an address assignment policy in which stubs (enterprises) are restricted to Provider-Aggregatable (PA) addressing, and (ii) a routing policy in which stub-level reachability information is not propagated into the global routing tables, without a significant administrative overhead or network usage inefficiencies. This policy combination will quite obviously result in smaller global routing tables with lower growth rates. But this policy combination creates four challenges: address renumbering, multihoming,

traffic engineering, and mobility. In developing solutions to these challenges, we combine proposed mechanisms with a novel idea of *address dismemberment*.

C. Key contributions

This work proposes a future internet architecture called LIMA in which per-packet data plane actions are kept minimal, while adding onus to handling relatively rare events such as address renumbering and access link failures. Analysis of BGP RIB data characterizes a benefit of LIMA in global routing table size reduction, and a cost of LIMA in the number of provider changes made by stubs.

After reviewing other work in Section II, and presenting the LIMA addressing/routing design in Section III, our solutions to the four challenges are presented in Section IV. Section V describes LIMA components. Section VI presents preliminary analysis of the benefits and costs of LIMA. Finally, section VII concludes our work.

II. RELATED WORK

We categorize recent solutions to the global routing table problem into four classes as shown in Table I. What allows the solutions in the top-right cell of the table to have the advantage of limiting the size of the global routing table without having to change applications and transport protocols is the introduction of a third parameter called an *identifier*. While ideally, location-independent names and topological location-dependent addresses (also called *locators*) should be sufficient, because applications embed IP addresses, and TCP does not have built-in support for address migration, this third parameter, identifier, is used to serve these two roles while locators are used for global routing. Furthermore, without the identifier, all host and router interfaces of stubs would need to be renumbered when stubs change their providers if all that exist are locators. While this concept of locator-identifier split has the above advantages, its disadvantages are that additional actions are required for every data-plane packet, plus control-plane mapping is required between identifiers and locators. Examples of data-plane overhead include address translations in NPTv6 [9] and DRUID [12], and IP-in-IP tunneling in LISP [5] and shim6 [10]. While the control-plane overhead of mapping from identifiers to locators may seem trivial from a bandwidth perspective, it could add operational costs in troubleshooting misconfigurations. In contrast, the LIMA solution does not require identifiers; it uses just names and addresses.

TABLE I: Classification of addressing and routing mechanisms

Routing Policy \ Address Assignment	Stub reachability permitted in global routing tables	Stub reachability not permitted in global routing tables
Provider Independent (PI) addresses permitted for stubs	Today's Internet (IPv4 and IPv6)	eFIT [3], ILNP [4], LISP [5], HIP [6], MILSA [7], FARA [8], NPTv6 [9], Shim6 [10], TurfNet [11]
Only Provider Aggregatable (PA) addresses for stubs	None	LIMA (our strawman solution)

LIMA's hierarchical addressing solution is the opposite of schemes that propose flat addressing, such as Routing on Flat Labels (ROFL) [13], [14]. The latter notes that besides the path stretch problem, hierarchical schemes complicate management, mobility and multihoming. It is precisely these challenges that are addressed in this work.

Why our solution is "less-is-more". We found that the LIMA policy combination as stated in Section I-B, and the dismembered addressing concept, can be tested with IPv6 as the network layer (NL), as it supports a key requirement of our design, multiaddressing, whereby an interface can be addressed with multiple addresses. LIMA is "less-is-more" because relative to the current-day IPv6 solution, it eliminates ARP and longest-prefix matching, relative to LISP and NPTv6, no tunneling or translations are required, and relative to Named Data Networking (NDN) [15], which requires name based per-packet lookups, and Accountable Internet Protocol (AIP) [2], which uses 160-bit addresses, LIMA requires lookups of much smaller (e.g., 32 bit) fixed-length dismembered address components. To avoid these more complex per-packet processing actions, LIMA pays a penalty in requiring additional management, but only for the relatively rare events of address renumbering and access link failures for multihomed stubs. Also, while the network layer is left untouched, almost all other aspects such as applications, socket interface, transport layer protocols, DHCPv6, DNS, and BGP, require changes to support LIMA to handle the four challenges it creates. But these changes can be evolved into the current-day Internet.

III. LIMA ROUTING AND ADDRESSING

LIMA is a design for inter-domain communications. Hence LIMA routers, which are envisioned to be IPv6 routers with some additional LIMA control-plane functionality, are designed for use as *border routers*, not internal routers. After presenting the LIMA addressing and routing schemes, we describe two examples of intra-stub networks.

A. Addressing

Simply put, LIMA addressing is a hierarchical scheme. It is similar to IPv4/IPv6 addressing in that there is a global routing prefix and an interface identifier. LIMA differs in the following ways:

- 1) it uses autonomous system (AS) numbers as prefixes
- 2) globally unique AS numbers are assigned only to providers, and stubs¹ are assigned provider-local AS

¹A "stub" is an enterprise such as a business, university or governmental agency.

numbers by their providers; this approach is unlike in IP where PI addressing is allowed for stubs

- 3) it reuses the address used in intra-domain networking as the interface identifier; this reuse is comparable to the IPv6 option in which MAC addresses are expanded to the EUI-64 format and used in the interface-ID field.

The first concept has been proposed by others such as [2]. The number of prefixes in today's IPv4 Internet (~335K) is much higher than the number of provider AS numbers (~6185, as reported in a later section). The second concept appears in other work, such as eFIT [3], which distinguishes user networks from provider networks, though as mentioned in Section I, eFIT requires an "identifier" while LIMA does not. The third concept is in keeping with the less-is-more theme in that it eliminates ARP and associated security threats. Even though ARP is eliminated, we propose the use of DHCPv6 in LIMA and not IPv6 Stateless Address Auto Configuration (SLAAC) [16] for security reasons. MAC addresses are sometimes viewed as confidential (e.g., they reveal the NIC manufacturer's name), but this can be circumvented with dynamically assigned MAC addresses. To prevent spoofing, source address filters can be added.

While the above description of LIMA addressing offers readers a differential view relative to IP addressing, the basic principle is to *dismember the address* into three distinct components: globally unique provider AS number, provider-local stub AS number, and stub-local intra-domain address (IDA). These components can be mapped on to the IPv6 address structure, by assigning, for example, the top 4 bytes to the globally unique provider AS number, the next 4 bytes to the provider-local stub AS number, and the bottom 8 bytes to carry the IDA. DHCPv6 and DNS will be modified to support this concept; the terms *L-DHCPv6* and *L-DNS* are used to represent the LIMA versions of these protocols. Providers can be assigned multiple AS numbers, and a provider can allocate multiple AS numbers to its stubs.

On the question of how to determine whether an organization is a stub or a provider, consider Content Delivery Network (CDN) providers, such as Google, Yahoo, Microsoft and Akamai. These providers do not typically offer IP transit service, and yet their domains are connected to many other stub domains. Similarly, some consumer ISPs do not provide transit service in that all packets either originate or terminate from their customers. One *answer* to this question is to use the number of inter-domain links from a given AS as the determinant for classifying an AS as a provider or a stub. This

is an issue for further study.

B. LIMA routing

LIMA routing differs from IP routing as follows. In IP, Tier-1 routing tables include information about PI and multihomed PA stubs, and routers implement longest prefix matching. In LIMA, Tier-1 routers will not have any information about stubs, and longest-prefix matching is eliminated. Instead, in LIMA, separate routing tables are maintained for the provider AS number and stub AS number in *provider network border routers*. Parallel lookups of both routing tables for fast operation can be implemented in hardware. For datagrams reaching the destination provider network, the stub AS number table is consulted to determine the border router within that provider network to which the datagram should be forwarded. *Stub border routers* consult the provider AS number routing table for outgoing datagrams, and the stub AS number routing table for incoming datagrams in stubs with multiple stub AS numbers.

C. Intra-stub network examples

We illustrate the concept of dismembered addressing with two intra-stub examples: a flat Ethernet switched network, and a hierarchical private IPv4 routed network. In the Ethernet case, IDAs are MAC addresses. As mentioned earlier, we propose that dynamic MAC addresses be assigned by the L-DHCPv6 server for strong asset management. In addition, the L-DHCPv6 server sends the {provider AS number, stub AS number} pairs to endpoints during initialization. L-DHCPv6 clients concatenate these dismembered components to create their IPv6 addresses for interface configuration. A host in a multihomed stub with a single Ethernet interface will have a single IDA (which is the MAC address), but multiple IPv6 addresses created by concatenating multiple {provider AS number, stub AS number} pairs with the MAC address. Similarly, a stub with a hierarchical private IPv4 routed network will use private IPv4 addresses as IDAs, which will be carried in the interface ID field of IPv6 addresses.

Each host interface will be assigned one (canonical) name corresponding to the IDA. Additionally a name can be assigned to a host, and mapped to the multiple canonical names of its interfaces. The L-DNS server will store a mapping between the name and the IDA for all endpoints, and a single entry mapping the organization name to its {provider AS number, stub AS number} pairs. A fully-qualified domain name for a host is created by concatenating the organization name with the host's name. L-DNS queries and secure dynamic DNS updates will support the dismembered address structure, and can be for just the stub name or for a particular endpoint name.

IV. STRAWMAN SOLUTIONS FOR THE FOUR CHALLENGES

A. Address renumbering

To achieve fully automated renumbering, LIMA adopts mechanisms from [17], [18], and classifies them as: (i) host-related, (ii) DNS-related, and (iii) router-related.

Host-related. The key features are: (a) multiaddressing, (b) Name Based Sockets (NBS) [19], (c) the LIMA concept of

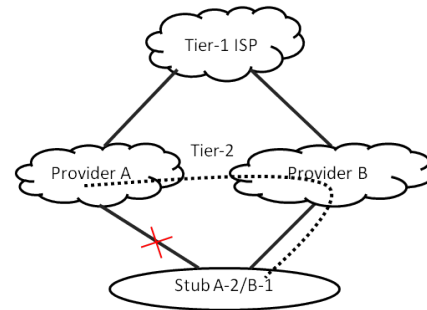


Fig. 1: LIMA multihoming

address dismemberment, and (d) the use of SCTP [20] or MPTCP [21]. *Multiaddressing* is key to address renumbering with zero downtime as the stub can maintain its link to the old provider for a day or two while executing all the steps needed for address renumbering. Next, restricting applications to only use domain names to prevent any address caching within applications is feasible with *NBS*. Applications only store and deal with names, while the NBS layer translates names to addresses. Third, address dismemberment allows for a *broadcast push* of the new {provider AS number, stub AS number} to all endpoints within the stub where L-DHCPv6 clients receive the pushed parameters, create IPv6 addresses by concatenating with the unchanging IDAs, and configure their interfaces. Finally, as most TCP connections are short-lived, and the access link to the old provider needs to be maintained for some time to allow DNS cached entries to expire (as explained next), these connections will terminate before the old provider based addresses are no longer usable. But given that there are some long-lived TCP connections, a transport protocol that supports dynamic address reconfiguration is preferred to one that requires reconnections. We propose the use of *SCTP* or *MPTCP* both of which support this feature.

DNS-related. Next, consider DNS updates and DNS caching. Today's DNS servers maintain a full IP address record for each domain name. In LIMA, we propose organizing this database to hold one entry that maps an organization name (e.g., virginia.edu) to one or more {provider AS number, stub AS number} pairs, and then have individual records that map host names to IDAs. Such a structure would make it easier to handle a provider change. To allow *cached* DNS records in other stubs and providers to work, stubs will maintain the link to the old provider for the maximum time-to-live value.

Router-related. We propose a LIMA router controller that runs (i) an L-DHCPv6 client, (ii) an L-DNS client, and (iii) a programmatic interface to the router. Techniques developed for automatic router configuration, such as Netconf [22], will be adopted for the dismembered addressing style of LIMA. Tunnel configuration applications should be designed to use names and not IP addresses. Automated techniques for updating firewall filters are required.

B. Multihoming

Fig. 1 shows what would happen under the LIMA policy in which the stub will have received a provider-local stub AS number from each provider forming its {provider AS number, stub AS number} pairs, e.g., A-2 and B-1. If the stub's link to provider A fails, given LIMA's routing policy restriction, the Tier-1 ISP will not have an entry for A-2, and consequently will not route datagrams addressed with A-2 via provider B. Our proposed solution to this problem is to provision a tunnel from the stub border router to provider A's border router passing through provider B's network as shown with the dotted line in Fig. 1, and use this tunnel as a backup path for the direct access link from the stub to provider A. A similar tunnel should be provisioned a priori to protect the access link to B. The Multi-Exit Discriminator (MED) field in BGP can be used to set the direct link as the primary option, and the backup tunnel as the secondary option, leading to seamless packet forwarding in case of access link failures.

In addition to this hitless forwarding of datagrams, *three* actions are required. *First*, to prevent new connections from using the A-2 addresses, the stub's fault management system, upon receiving SNMP traps from the router indicating link failures, informs the L-DHCPv6 server causing it to broadcast messages to alert all endpoints to stop using A-2 based global addresses. *Second*, the fault management system should also inform the L-DNS server (a LIMA version of DNS server) causing it to stop providing A-2 addresses in its replies to queries. *Third*, the L-DHCPv6 clients at endpoints should initiate SCTP or MPTCP dynamic address reconfiguration for any ongoing connections.

C. Mobility

While a significant fraction of the devices in a future Internet of Things (IoT) will likely be wireless, and of this fraction, a significant portion may be mobile, the fraction of this set that is roaming (i.e., outside the home location) is likely to be small. We thus conclude that a hierarchical structure such as LIMA can be used, with mobility handled in much the same way as mobile IP.

However, to reduce the *path stretch* problem, we propose augmenting a mobile IP type solution with a dynamic DNS solution. There have been a number of proposals to use the secure dynamic DNS update feature to handle mobile location management [23], [24]. LIMA incorporates these solutions. Involving the DNS server is useful in LIMA because this allows the stub DNS server to inform its roaming mobiles of a change in its home {provider AS number, stub AS number} when a provider is changed. A mobile IP like solution is still needed in LIMA to handle connections initiated with DNS cached addresses wherein the home stub border router supports home agent functionality for its own endpoints, and foreign agent functionality for visitors.

D. Traffic engineering

Provider traffic engineering. The elimination of current-day prefixes in favor of ASNs and longest-prefix matching in

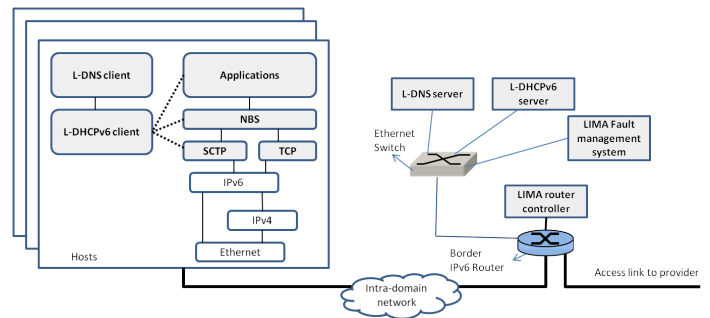


Fig. 2: LIMA stub architecture

LIMA, and the routing policy of disallowing stub reachability in global routing tables, can cause path stretches. For example consider two backbone providers, Internet2 and ESnet. These two networks interconnect at Los Angeles, Seattle, Chicago, New York, and Washington. Consider two stub customers of ESnet, one in California (CA) and the other in New York (NY). If stub reachability propagation is allowed, ESnet can report longer-prefix reachability to Internet2 for these two stubs. Packets destined to the ESnet CA stub originating at a stub connected to say a Kansas City router in Internet2 would be forwarded westward within Internet2 towards its Seattle router, while packets destined to the NY stub would be forwarded in the opposite direction toward Chicago if different MED values were configured by ESnet in the BGP updates sent by different routers for each stub. However, in LIMA, if ESnet is allowed to advertise only one provider AS number, such efficient routing cannot be achieved.

Our *proposed solution* is to allow the assignment of multiple AS numbers to providers allowing them to address different parts of their networks with different provider AS numbers. Analysis will be undertaken in future work to achieve a good trade-off between keeping the number of assigned AS numbers small so as to not increase the global routing table size significantly while achieving low path stretch values.

Stub Traffic Engineering. To load balance incoming traffic across its providers in today's Internet, a multihomed stub can selectively send longer prefixes into the global routing table through each of its providers. But with the LIMA policy, this is not possible. We propose a *DNS and DHCP based solution* to this stub traffic engineering problem. The authoritative DNS server for the stub could order the multiple addresses returned in DNS responses if applications are programmed to choose addresses accordingly. For outgoing traffic engineering, the L-DHCPv6 client could use different orders when communicating the assigned {provider AS number, stub AS number} pairs to its endpoints, and have the NBS layer choose addresses accordingly.

V. LIMA COMPONENTS

Fig. 2 shows the internal architecture of a stub network with a border IPv6 based LIMA router. Applications will need to be modified to use NBS sockets rather than TCP or UDP sockets. NBS defines a new family type AF_NAME for the socket call.

The `listen` and `accept` calls for the receiving side, and the `open` call for the sending side use domain names instead of IP addresses. `Read` and `write` system calls interface with the NBS socket descriptor. The source name is carried in an IPv6 extension header to the destination in the first packet for the receive side application to use cached information about sources but with names rather than IP addresses (e.g., licensing servers).

Besides modifying DHCPv6 to support the *dismembered structure* of LIMA, a *broadcast push* operation is required for address renumbering or when an access link fails. The DHCPv6 `Reconfigure` message cannot be used as this message was designed for reconfiguring a single client, while we need a message to add or delete just the {provider AS number, stub AS number} pairs for all globally addressable endpoints within the stub. It is also sufficient for the L-DHCPv6 server to send just the IDAs of the stub border router interfaces (equivalent to Gateway address in today's Internet) and the IDA of the DNS server (today the whole IP address of the DNS server is sent to DHCP clients). As LIMA will rely more on domain names than the current Internet, more frequent secure dynamic DNS updates are anticipated from DNS clients running on endpoints as shown in Fig. 2. Initial registration of names to IDA mappings will be executed through the L-DHCPv6 server given that new hosts will not have the required certificates for authenticated DNS updates. This requires adding names to DHCPv6 messages.

Similarly modifications are needed to DNS to support the dismembered structure. For example, the resource record (RR) database structure will be modified as described in Section IV-A. An Autonomous System (AS) resource record, and support for mobility will also be added.

The *LIMA fault management system* shown in Fig. 2 is required to interface with the L-DHCPv6 and L-DNS servers via a new protocol for access link failure handling. The *LIMA router controller* will support address renumbering of the router interfaces during initialization and provider changes.

Modifications to BGP are anticipated to support LIMA's dismembered addressing. For example, as the stub AS numbers are provider-local, BGP updates regarding stub AS numbers are required between stub border routers and their provider border routers, while provider AS number reachability will be propagated between providers.

VI. ANALYSIS

While a number of different analysis and prototyping efforts are required to fully evaluate LIMA, in this preliminary study, we report two sets of analyses. Section VI-A examines the benefit of LIMA in global routing table size reduction, while Section VI-B characterizes a measure of address renumbering.

A. Routing data analysis

By analyzing the Routeviews RIB data [25] for the last ten years, we characterize the growth in the total number of AS numbers, stub AS numbers, and provider AS numbers, as shown in Fig. 3. In LIMA, global routing tables will follow the low-rate growth pattern seen in the provider ASs plot, with a

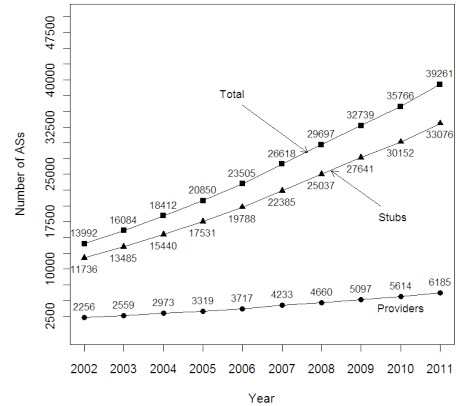


Fig. 3: Provider, stub, and total number of ASs

TABLE II: Across all stubs (approx. 33K)

Month	#Provider additions	#Provider deletions
05/2011	1396	1049
06/2011	1408	1024
07/2011	1454	1112
08/2011	1435	1102
09/2011	1317	943
10/2011	1359	1092

current-day number of 6185 providers. Contrast this with 335K prefixes today, and the exponential yearly growth rate of 17% [2].

B. A measure of the address renumbering overhead

To characterize the costs of address renumbering, we analyzed Routeviews RIB data to determine the frequency with which stubs add and/or delete providers as shown in the Table II. Some stubs just added or deleted a provider. But as both provider additions and provider deletions incur a renumbering operation, both are listed in the table.

Fig. 4 plots the mean number of provider changes per month made by stubs as a function of their prefix block sizes. The larger the prefix block size, the more potential for trouble tickets even if the address renumbering procedure is fully automated. While the absolute numbers have grown with time, the ratio has remained almost constant. For example, in years 2011 and 2002, the average monthly number of provider changes made by /24 stubs are 543 and 195, and the total number of stubs in 2011 and 2002 are 29466 and 11250, respectively. Therefore, the average monthly number of provider changes per stub is 0.018 in 2011 and 0.017 in 2002.

VII. CONCLUSIONS

It appears that a bold address assignment and routing policy combination, which is likely to be unpopular with today's administrators, is feasible with innovative schemes to handle the challenges of address renumbering, multihoming, mobility, and traffic engineering. This policy combination calls for eliminating Provider Independent (PI) addressing for stubs, and

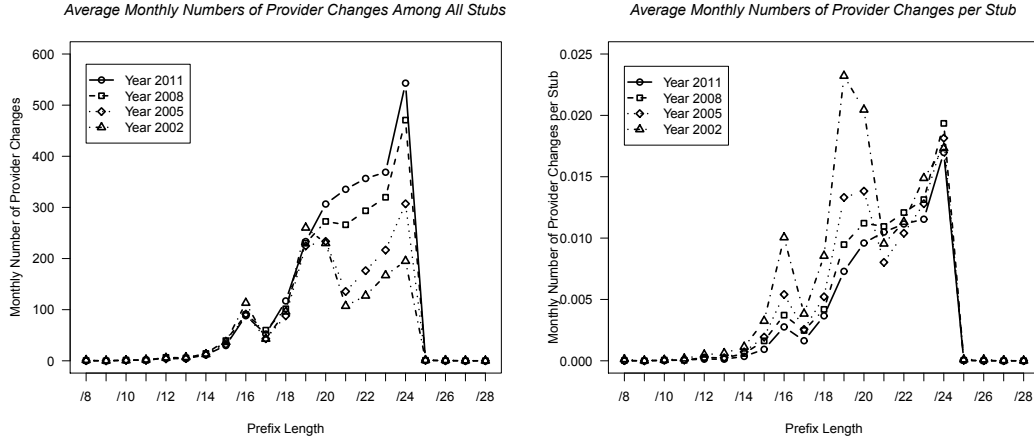


Fig. 4: Average per-month numbers

for disallowing the propagation of stub-level reachability into global routing tables. While address renumbering of a large stub is challenging today, with some fundamental changes, such as disallowing the use of IP addresses in applications in favor of names, name based sockets, and dismembered addresses allowing for broadcast pushes of new provider numbers, this solution seems promising. LIMA scales back the per-packet processing actions relative to even today's IP (by eliminating longest-prefix matches) as well as the newer locator-identifier split solutions. Instead it adds control- and management plane actions, which are however required only for handling relatively rare events such as provider changes and access link failures. This less-is-more approach will decrease capital expenditures of routers by lowering memory and processing costs, as well as operational (administrative and power consumption) costs.

REFERENCES

- [1] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB workshop on routing and addressing," Internet Engineering Task Force, RFC 4984, Sep. 2007. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc4984.txt>
- [2] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Accountable internet protocol (AIP)," in *Proc. of the ACM SIGCOMM*, 2008, pp. 339–350.
- [3] D. Massey, L. Wang, B. Zhang, and L. Zhang, "Enabling future Internet innovations through transitwire (eFIT)." [Online]. Available: <http://www.nets-find.net/Funded/eFIT.php>
- [4] R. Atkinson, S. Bhatti, and S. Hailes, "Evolving the internet architecture through naming," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1319–1325, Oct. 2010.
- [5] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol (LISP)," IETF Draft Version 16, Tech. Rep., Oct 2011.
- [6] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host identity protocol," Internet Engineering Task Force, RFC 5201, Apr. 2008. [Online]. Available: <http://tools.ietf.org/html/rfc5201>
- [7] J. Pan, R. Jain, S. Paul, and S.-I. Chakchai, "MILSA: A new evolutionary architecture for scalability, mobility, and multihoming in the future internet," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1344–1362, Oct. 2010.
- [8] D. Clark, R. Braden, A. Falk, and V. Pingali, "FARA: Reorganizing the addressing architecture," in *Proc. of ACM SIGCOMM Workshop on Future Directions in Network Architecture*, 2003, pp. 313–321.
- [9] M. Wasserman and F. Baker, "IPv6-to-IPv6 network prefix translation," Mar. 2011. [Online]. Available: <http://tools.ietf.org/html/draft-mrw-nat66-12>
- [10] C. de Launois and M. Bagnulo, "The paths toward IPv6 multihoming," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 2, pp. 38–50, Second Quarter 2006.
- [11] J. Pujol, S. Schmid, L. Eggert, and M. Brunner, "Scalability analysis of the TurfNet internetworking architecture," in *Proc. of IEEE GlobeCom*, Nov. 2007, pp. 1878–1883.
- [12] J. Touch, I. Baldine, R. Dutta., G. Finn, B. Ford, S. Jordan, D. Massey, A. Matta, C. Papadopoulos, P. Reiher, and G. Rouskas, "A dynamic recursive unified internet design (DRUID)," *Computer Networks*, to appear 2011.
- [13] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker, "ROFL: Routing on flat labels," in *Proc. ACM SigComm*, 2006.
- [14] A. Singla, P. B. Godfrey, K. Fall, G. Iannaccone, and S. Ratnasamy, "Scalable routing on flat names," in *Proceedings of ACM Co-NEXT*, 2010, pp. 20:1–20:12. [Online]. Available: <http://doi.acm.org/10.1145/1921168.1921195>
- [15] L. Zhang, D. Estrin, and J. B. et. al., "Named Data Networking (NDN) Project," Tech. Rep., 2010. [Online]. Available: <http://www.named-data.net/ndn-proj.pdf>
- [16] S. Thomson and T. Narten and T. Jinmei, "IPv6 stateless address autoconfiguration," IETF RFC 4862, Sep. 2007.
- [17] B. Carpenter, R. Atkinson, and H. Flinck, "Renumbering still needs work," Internet Engineering Task Force, RFC 5887, May 2010. [Online]. Available: <http://tools.ietf.org/html/rfc5887>
- [18] T. Chown, M. Thompson, A. Ford, S. Venaas, C. Schild, and C. Strauf, "Cookbook for IPv6 renumbering in SOHO and backbone networks," University of Southampton, Tech. Rep., 2005. [Online]. Available: <http://www.6net.org/publications/deliverables/D3.6.1.pdf>
- [19] J. Ubillos, M. Xu, Z. Ming, and C. Vogt, "Name-based sockets architecture," Sep. 2010. [Online]. Available: <http://tools.ietf.org/html/draft-ubillos-name-based-sockets-03>
- [20] P. Natarajan, F. Baker, P. Amer, and J. Leighton, "SCTP: What, Why, and How," *IEEE Internet Computing*, vol. 13, no. 5, pp. 81–85, sept.-oct. 2009.
- [21] A. Ford, C. Raiciu, and M. Handley, "TCP extensions for multipath operation with multiple addresses," Oct. 2010. [Online]. Available: <http://tools.ietf.org/html/draft-ietf-mptcp-multiaddressed-02>
- [22] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network configuration protocol (netconf)," IETF RFC 6241, 2011.
- [23] A. Ahmed, S. Reaz, M. Atiquzzaman, and S. Fu, "Performance of DNS as location manager," in *IEEE Int. Conference on Electro Information Technology*, May 2005, pp. 1–6.
- [24] B. Yahya and J. Ben-Othman, "Achieving host mobility using DNS dynamic updating protocol," in *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, oct. 2008, pp. 634–638.
- [25] Route Views Project. [Online]. Available: <http://www.routeviews.org/>