University of Amsterdam
System & Network Engineering

Large Installation Administration

Managing IPv6 address assignment in large scale networks

Authors:
Aleksandar Kasabov aleksandar.kasabov@os3.nl
Marek Kuczyński marek.kuczynski@os3.nl

March 28, 2011

Contents

1 Introduction

Large data centers serve a great number of online services. Even in restricted networks, such as computing clusters, servers need to join and leave the LAN on demand to ensure a highly-available service. Virtualization helps administrators to create and replace hundreds of heterogeneous network resources on-the-fly. This makes their addresses difficult to manage and manual address assignment does not scale.

Additionally, new hosts on the network must be updated with the network settings for each individual company site. This way they can have in- and outbound connectivity and become aware of the available network services. IPv6 introduces stateless address autoconfiguration but cannot register the connected host into company infrastructure. This requires additional installation of software on each joining host which would contact a central naming service for registration. Only from that point on, it is discoverable and available from outside the company network.

2 Research question

This project was brought up to investigate the current practices in managing IPv6 addresses in large-scale datacenters. The paper looks at the ways for assigning addresses and configuring routable network hosts. Hence, the research group formulated the following research question and three sub-questions which would build the foundations for the project conclusion:

- What is the best practice for managing globally routable IPv6 address space in a large scale network?
  - Comparison of stateless and stateful autoconfiguration in large scale networks.
  - How to locate and address a server in an IPv6 network?
  - What current solutions facilitate the administration of IPv6 address space in large scale networks?

The following sections of this paper describe stateful and stateless approaches in leasing addresses. The last section highlights one newly developed framework for managing and transitioning to IPv6 addresses.

3 Stateful addressing

It is convenient to configure hosts on a network using a stateful approach. This can be done through a central service which new hosts on the network contact in order to be assigned with an address. The term *stateful* stems from the fact that the server must keep track (*state*) of the addresses it leases to prevent duplicates. Additional configuration parameters can also be pushed to network hosts. The de facto standard for stateful addressing is the Dynamic Host Configuration Protocol, which has also been designed for IPv6[2].

3.1 DHCPv6

A centralized approach allows sophisticated control over the configuration process, allowing administrators to fine-tweak a configuration policy[1] and let the DHCP server apply it to newly discovered client hosts. The Internet Systems Consortium (ISC) has implemented the DHCPv6 protocol into their DHCP daemon. It supports a large set of configuration parameters [2] which can be pushed client hosts. In early years, this list was very limited and previous research [3] has noted that as an issue when deploying IPv6 in large scale sites. However, the current version is mature and provides various configuration options to network hosts among which:

- Default gateway
- VLAN information
- DNS servers list
- MAIL, SIP, NTP, WINS, etc.

DNS service registration is an essential feature to provide out-of-site host reachability through naming. It requires the DHCP server to push a Dynamic DNS (DDNS) update and register every client hostname. It is also possible for each client system to contact the DHCP server directly but that requires additional software to be installed, e.g *ddclient* on linux and *Active Directory* on Windows. Authenticated DNS record updates are also supported by using Transaction Signatures (TSIG) [3].

For network with high availability, administrators can deploy a DHCP failover server. This means that DHCP servers can share a common pool of addresses and lease those to new hosts. If the first server fails, or needs to be temporarily down (e.g. due to a software update), the failover server would replace its job. The ISC DHCP server currently supports a maximum of 2 servers - primary and secondary - which are not load balanced and at some point can become a limitation in large-scale networks. Figure 1 illustrates the process of configuring client hosts using two DHCP servers and further propagate domain name updates to the master DNS server.

[1] `http://www.cisco.com/en/US/docs/ios/ios_xe/ipv6/configuration/guide/ip6-dhcp_xe.html`

[2] `http://www.iana.org/assignments/bootp-dhcp-parameters`

[3] `http://www.ops.ietf.org/dns/dynupd/secure-ddns-howto.html`

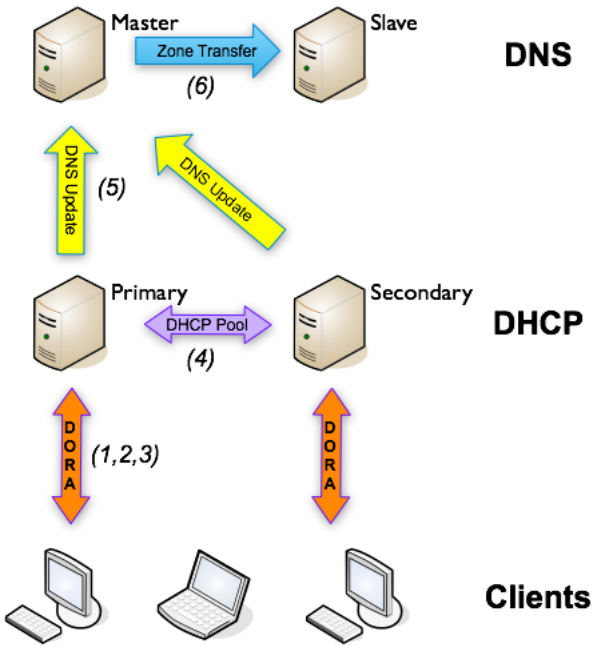

Figure 1: DHCP failover
[4]

3.2 Discover, Offer, Request and Accept

"DORA" stands for Discover, Offer, Request and Accept to identify the possible types of packets which are exchanged between the DHCP client and server. It is the first step of the configuration process after which client hosts are fully network configured and have connectivity (at least, within the internal network). The next action (denoted with number 4) which occurs is that the DHCP server synchronizes its DHCP pool with the other DHCP server. Effectively, this means that the newly assigned host IP is communicated with the other server. Further on, the DHCP server, which was contacted for leasing an IP address, contacts the master DNS server to push a DNS update. The latter is propagated to the slave DNS server as well.

DHCP Relay is another feature of the ISC DHCP application. It allows multiple networks to be served by one central DHCP server. This facilitates the administration hierarchy in large network sites and the management of various configuration policies. A DHCP relay servers can be deployed to handle DHCP requests for each sub-network and further forward those to the primary DHCP server. There is now limitation on the DHCP relays which helps for scaling company sites. Figure 2 demonstration the various ways in which hosts can make use of a DHCP service, either directly, or through a DHCP relay.

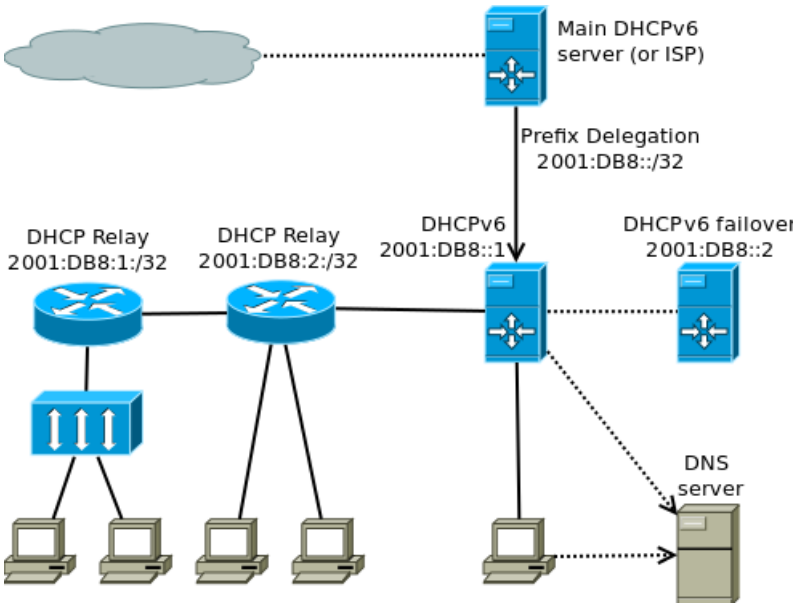

Figure 2: DCHP relay and prefix delegation

3.3 Prefix Delegation

The figure also demonstrates the use of DHCP prefix delegation. It allows the main DHCP server to be configured across administration domains. This way the router of the ISP (noted as *Main DHCP server* on the figure) can push a configuration prefix (2001:DB8::/32) to a company DHCP server which will then be used for address assignment of all company nodes. Additionally, the DHCP relay can also push DDNS updates. The ISC has implemented the relay feature for IPv6 in Jan 2011 [5], but its popularity is still low.

A tool which provides a GUI for administering the DHCP server is called *gadmin-dhcpd*. It provides a convenient configuration overview for a DHCP server instance. It allows new subnets to be added to the DHCP configuration and shows currently leased IP addresses. However, it does really provide any automation which still makes the DHCP server maintenance a manual task.

[5] `http://ftp.isc.org/isc/dhcp/dhcp-4.1.1-P1-RELNOTES`

4 Stateless addressing

IPv6 comes with several new feature which were not present in IPv4. One of them is called Stateless Address Autoconfiguration (SLAAC) [4] and it simplifies the aspects of address assignment. It defines fixed size of the network prefix, which allows hosts to build their own link-local addresses by using their link-layer (unique) MAC address. It is represented as *IEEE 802 Address* on figure 3. However, it needs to be extended with two fixed bytes (0xFFFF and 0xFFFE) in order to generate a 64 bits address, called Extended Unique Identifier (EUI-64).

The figure also demonstrates one additional bit conversion to the resulting address. It is the flip of the *Universal/Local* bit from the MAC address. If it is 1, it denotes that the MAC address is locally assigned by the administrator. However, IPv6 addresses have long sequence of digits 0, and it is convenient to substitute them with the double colon (*::*) notation. Thus, it is convenient to flip the *Universal/Local* bit to 0 too.

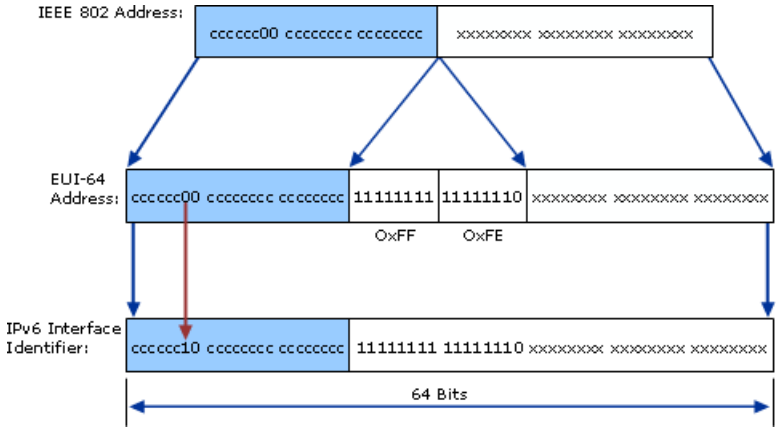

Figure 3: IPv6 stateless addressing

4.1 Including the IPv6 prefix

In order to construct a globally unique IPv6 address, the host need to be updated with the network prefix of the network which it is on. The IPv6 protocol offers Prefix Advertisements as a mechanism for doing that. A designated router would run the prefix advertisement service (RADVD[6] daemon on linux) and broadcast messages containing the network prefix. Additionally, Router Advertisements can push a list of recursive DNS servers to IPv6 hosts. This ensures their outbound connectivity and makes them be able to discover services and their addresses.

However, no other configuration options can be provided to network hosts. This is a current limitation of the protocol and new supported options are expected to be developed. Nevertheless, the IPv6 society has considered the issue since the beginning of the design of IPv6. The solution is that IPv6 hosts would make use of well-known multicast addresses to discover required services. Naturally, available services must also be aware of the particular multicast addresses

[6] `http://www.litech.org/radvd/`

that they are required to join. This approach strives for well defined service-to-multicast-address mappings in advance which are difficult to agree upon across autonomous administration domains.

When an institution scales to the point that manual network prefix configuration becomes infeasible, the prefix can also be pushed automatically from a master router which is in a higher administrative domain using the Router Renumbering mechanism[5]. This is illustrated on figure 4 and allows for LAN mobility - it retains hosts' connectivity even if the ISP is changed, or if it decides to deploy a new naming scheme. The Renumbering Protocol is supposed to eliminate the nuisance from manually configuring network prefixes in each routing device. However, it is still in a proposition phase and the only available authentication is based on IPsec, which is still considered difficult to implement.

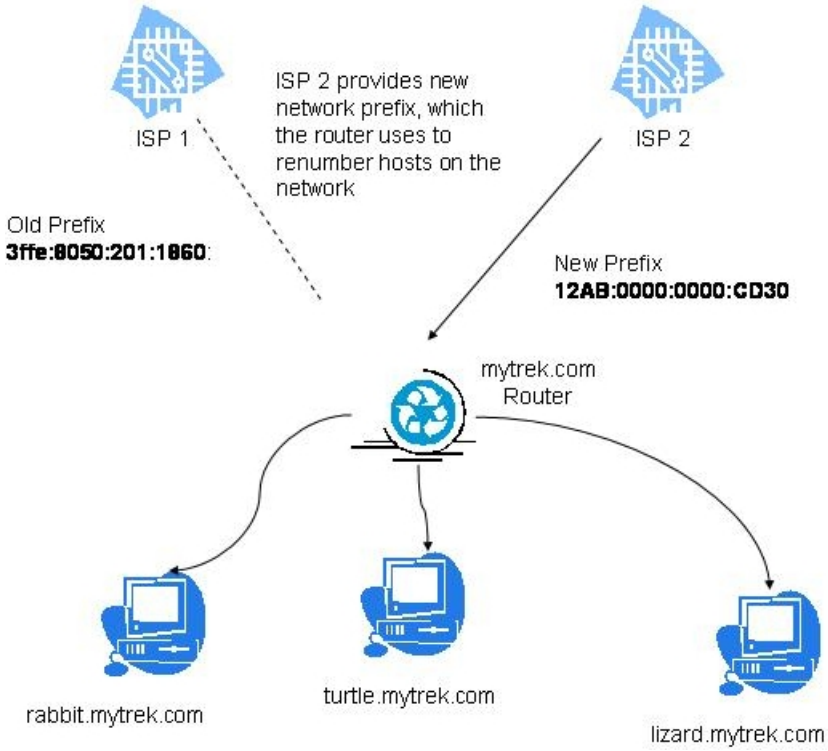

Figure 4: Router Renumbering
[7]

Eventually, SLAAC alone does not solve the problem of identifying hosts from outside the network without a central naming service. And since Router Advertisements do not support DDNS updates, additional software must be installed on each client so that it can update its DNS record. The previous section mentioned two tools for this task but it can still be cumbersome to install and configure them in large-scale infrastructures. It is possible to deploy new OS images that already contain these components though, so that the machine can have connectivity without human intervention after the install completed.

5 Migrating sites to IPv6 - Locator/ID Separation Protocol

The previous chapter described some of the possibilites that a datacenter can use to assign addresses within their own private network. In most cases, the switch-over to IPv6 should be manageable within one physical site, as long as connectivity to the rest of the corporate network and to the outside world remains. In this chapter, an alternative approach is presented that can help companies maintain the connectivity in between their IPv6 sites and the rest of the world.

It was decided by the research group to extend upon this specific subject because of the unique features it can provide for large datacenter setups. The assignment and management of IP address space within datacenter sites was already covered to some extent in the first part of this paper, so the intra-site connectivity is interesting to cover as well. Another motivation to look at the state of the protocol was the research project conducted by OS3 alumni Atilla de Groot [9]. He looked at the state of the Linux implementation of LISP back in 2009, but a lot of IETF additions have been made to the protocol since that time. But first, a bit of a technical background of the protocol that has been researched will be given.

5.1 Locator/ID Separation Protocol

The Locator/ID Separation Protocol (LISP) is a protocol that is currently being developed by a workgroup of Internet Engineering Task Force (IETF) [6]. The main reason for the development of this protocol, is that the current addressing method used with IPv4 and IPv6 doesn't work well on an ever increasing scale. Currently, both protocols do address a node using a single IP identifier, which contains information on both their location on the Internet and their ID within the destinations network. LISP introduces a new overlay network that runs over the existing Internet, but it does remain compatible with the current Internet infrastructure.

Figure 5: LISP logo

This concept would work fine if the Internet had a hierarchical design, but the wide variety of IP address ranges spread all over the world makes it a memory intensive and hard to maintain topology. In general, IP address ranges are aggregated in the best way possible in order to decrease the storage of this information, but the eventual transition IPv6 will make this problem even worse. The top-level routers will have to maintain both ranges in order to keep v4 and v6 connectivity available. Right now, these tables contain reachability information for several thousand autonomous systems that exist around the globe.

The IETF decided to further develop this protocol in order to achieve better scalability of the rapidly growing forwarding tables that Border Gateway Protocol (BGP) routers have to maintain. The BGP protocol is used de facto when routing traffic in between autonomous sytems, so the solution for this problem can benefit all Internet providers and exchanges at once. The figure below displays the growth that the tables have had since 1988;

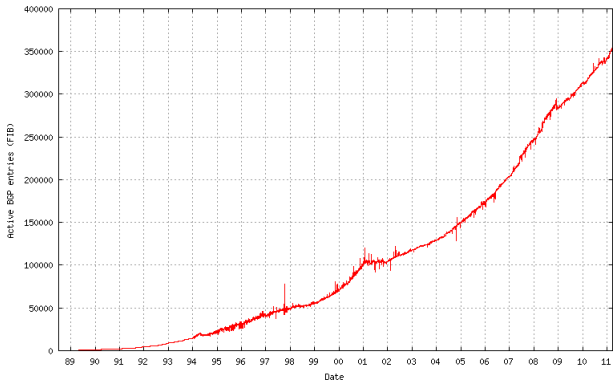

Figure 6: Rapid growth of BPG tables
[7]

While it is hard to say how large these tables will be in a couple of years, it is undeniable that the table sizes have been growing exponentionally. The tables require more and more of expensive router memory, even for the routers that are never accessed from the connected AS. Just as an indication, the full BGP table was almost 63MB in March 2011 [8]. It should be noted that the situation right now is still manageable with IPv4, but the transition to Ipv6 will be most interesting in this regard.

This is where LISP comes in; its purpose is not only to decrease the size of routing tables though and to keep global routing manageable. It also brings benefits to parties that require IP address mobility, scalability, and availability. The following chapters will focus on how this new type of spreading routing information can help with the reachability and IPv6 implementation within datacenters.

5.2 Scaling problems with single identifier

As mentioned before, IPv4 and IPv6 use a single identifier to address a node on the Internet. When you register for an Internet connection with a service provider, you will either receive one of these IPs' (usually the case with IPv4 since they're becoming rare) or you can lease a block of sequential addresses, known as a subnet (IPv4) or a prefix (IPv6). This unique number or network aggregate is then used to identify and address you on the Internet.

You could consider this to be too specific information, especially when looking at the amount of IP prefixes and autonomous systems that are already around. While the LISP protocol is still dependant on BGP information to get packets from A to B, it is capable of severely decreasing the very specific BPG information that is slowly filling up BGP routers now. It does this by delegating

the routing of the end point identifiers by a node closer by or even within the destination network. This approach offers a lot of promising new possibilities, of which some will be covered in the remainder of this paper.

5.3 Splitting up the address space

Addresses in LISP are split up in two parts;

Routing Locators (RLOCs) + Endpoint Identifiers (EIDs)

The endpoint identifier (EID) of a node consists of two parts in the LISP network. The first part contains a reference to the IP address location of the network, which is a network plus its subnet or a prefix. The second part contains an identifier that is used internally to distinguish the node. This can be the IP address that the node is currently using, but this is not a neccesity. As a matter of fact, it can be any type of EID since LISP is protocol agnostic (i.e. if IPv7 ever gets released, it will be usable as an EID).

It should also be noted that this EID is not a new component that you have to introduce into your network. An IP that you are currently using can be used as an EID as well, or you can route the traffic directly outside of the LISP overlay network.

Routing Locators (RLOC) are routers or servers that point the traffic that is entering a network to its final destination. They form a key component in the LISP overlay, since they allow the receiving site to traffic engineer the traffic, which takes some effort to achieve normally. Consider the following set-up;

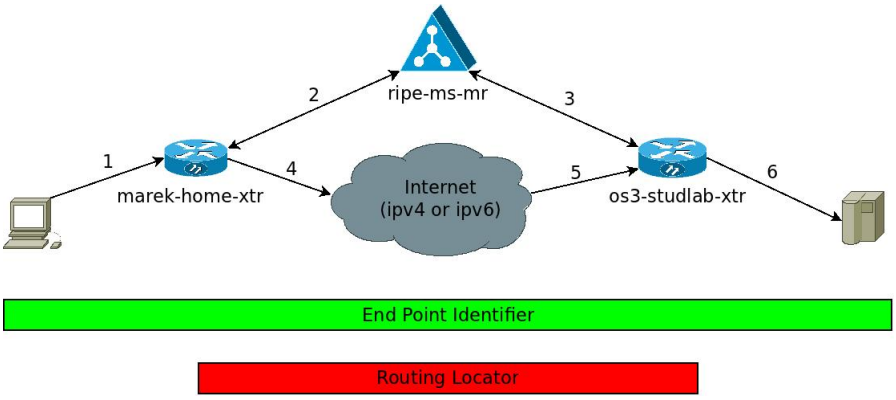

Figure 7: Theoretical LISP scenario, displaying the address retrieval procedure

Here we see a theoretical LISP topology with two sites. Both of them have routers that perform the Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR) functions of the LISP network. This means that they can receive and transmit LISP encapsulated packages. Both functionalities are often done within one Internet gateway, in this case the ITR and the ETR are called an XTR together. The figure also shows that the EID is present at all times, and that it gets encapsulated in a packet with a RLOC address in the header. This concept is explained below in the following overview of events;

1. Marek's PC contacts the home router with an IP address for OS3. This IP has been retrieved from DNS.

2. The router checks it's own cache for the RLOC of the OS3 network. It cannot find it, so it queries i.e. a RIPE map resolver for the RLOC. This server also performs the role of map server where nodes can register in order to be found through LISP. You only need one registration with a "Tier 1" map server in order to be accesible through LISP.

3. Incidentally, the OS3 router has registred with the RIPE map resolver/server as well, so the RIPE server can resolve this query. Please note that it is not required for the RIPE server to be the middle man in this case, since in most cases it will have to query one of its peers to see if the destination did register there. In this case, we're lucky, and Marek's router will receive the RLOC information about OS3 first-hand from RIPE. It is also unknown if RIPE would provide this type of service, but they do participate in the LISP beta network right now (more on this later).

4. Marek's router contacts the RLOC given by RIPE, and receives information about where the RLOC wants the traffic to go. This part is where traffic engineering in LISP can be applied, more on this later. Sending traffic to this new destination happens in the conventional way through the Internet.

5. The packet arrives at the OS3 XTR, which can direct the traffic from there based on its EID.

Traffic flowing the other way arround would follow the same process.

5.4 Datacenter examples

In case a network is not LISP compatible, a Proxy XTR can be used to encapsulate the traffic into the LISP network. This PITR (sometimes called PXTR) will attract IP traffic destined for a particular network through BGP route announcements. Because LISP can help in decreasing BGP table sizes, the PITR should announce large IP blocks and divide them further from there. This could for example be done for all customers of a service provider that have LISP capable routers or for all universities in The Netherlands collectively. Nothing stops you from announcing your own AS into the cloud, and then encapsulating the regular IPv4/IPv6 traffic to LISP.

A simple example is displayed below;

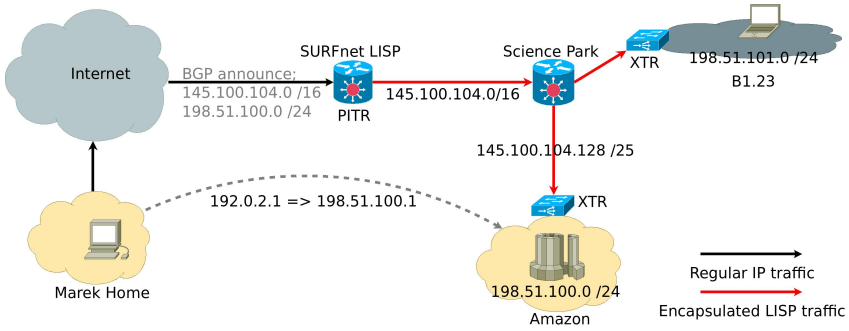

Figure 8: Theoretical scenario; the packetflow from Marek gets redirected to the Amazon cloud

The picture displays how the IPv4 subnet `198.51.100.0/24` could be forwarded to through the existing OS3 network. By getting the traffic through the PITR, the traffic can traverse to another XTR where it exits the LISP network (in this example, the traffic is part of the LISP network between the PITR and the bottom left XTR). In the figure, only IPv4 is used, but this can be applied with IPv6 as well. When introducing IPv6 into your network, you could consider to make the servers dual stack and to tunnel the IPv6 traffic to your server over existing IPv4 infrastructure. The same can be done over the public Internet, which will remain partially IPv4 for quite some more time. An illustration of this is shown below;

The figure above displays how a datacenter topology could look like. When a user wants to visit the webservers displayed on the top, his/her PC or residential gateway will first ask the LISP MR/MS about the RLOC of the webpage. The answer can consist of two different records, one for IPv4 and IPv6 each. The node can then make a pick between an address and access the page. Again, this process can be made invisible to the user if necassery.

The technology to do comparable tricks has been around for a while already, but the beauty of LISP is that it scales well between LISP sites and non-LISP sites and that no specific adjacencies have to be made. It can be implemented (partially) over the course of a longer period, without significant investments. As an example, Facebook has been experimenting with LISP for a while now. Their currently topology does not utilize LISP to the fullest, but they have been very positive about the low cost of investment and the rapid deployment of it

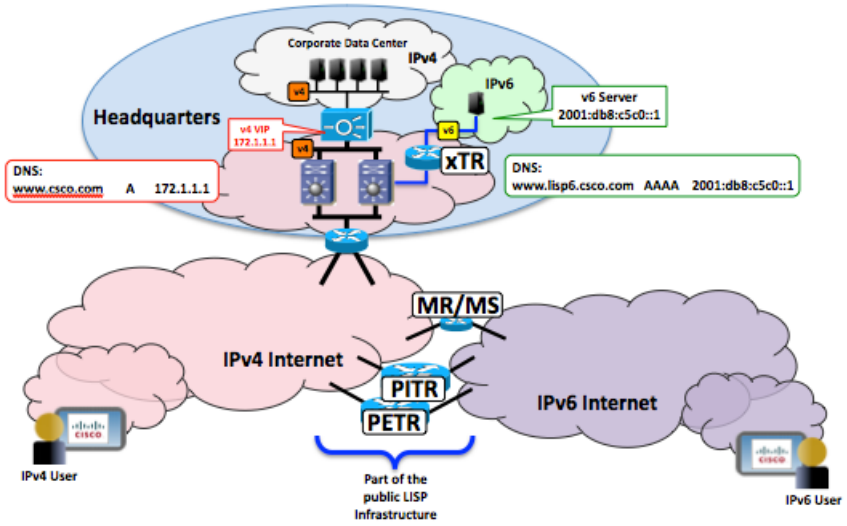

Figure 9: IPv6 islands can be interconnected with LISP [12]

[11] [10]. In the case of Facebook, LISP provides some IPv6 connectivity to their social network service, which can be expanded or modified fully at will.

5.5 Current Implementation of LISP

As already discussed, LISP is an open standard that is in development in various ways right now.

5.5.1 OpenLISP

This is an implementation of LISP daemons on FreeBSD, developed by the University of Louvain. It is uncertain what the current state of development is, more information is available on the OpenLISP website [13]. The latest release dates from september 2010 and is compatible with FreeBSD up until version 8.2.

5.5.2 LISP for Linux

A prelimenary implementation of LISP for Linux has been under development by Mathieu Peresse. The features include a modified kernel source, a lisp userspace daemon (lispd) and a modified version of 'iproute' that can handle LISP packets. Again, more information plus the sources can be found on GitHub [15].

5.5.3 LISP Internet Groper (LIG)

LIG is a tool that can be used to request information about LISP sites through the public beta network [16]. With the '-m' flag, a mapping server can be entered, after which an IP network can be queried for its RLOCs. It has been

developed by David Meyer and is a great tool to get first insights into the operation of LISP. More information, including the sources, is available on GitHub [14]. Please note that LIG does not appear to work correctly from behind NAT environments.

5.5.4 LISP on Cisco routers

The implementation of LISP on Cisco platforms is very well developed compared to the open source implementations mentioned above. Stable versions of LISP are available in IOS, IOS-XE and NX-OS, which is one of the reasons why these devices are dominating the LISP public beta network right now. Cisco also has a couple of valuable papers about LISP implementation strategies on their website [17].

An overview of these implementations can be found on the LISP4 community page [18].

5.6 Simple Multihoming with multiple links and addresses

Another interesting and easy to deploy option is multihoming. This option is best illustrated by a practical example using the LISP beta network and a webpage using both IPv4 and IPv6 addresses. In this example, the APNIC mapping server is queried for the APAN LISP site. This LISP namespace is retrieved through DNS and it has a test IP range assigned (153.16.0.0/16). This address could be used as the main portal for a webpage if a provider can lease it to you in the future, much like with the previous examples given. After the query is sent to the APNIC mapping server, the following results are returned;

```
1  marek@london:~$ lig -m apnic-alt.rloc.lisp4.net apan-xtr.lisp4.net
2  Send map-request to apnic-alt.rloc.lisp4.net for apan-xtr.lisp4.net
       ...
3
4  Received map-reply from 203.181.249.170 with rtt 0.50400 secs
5
6  Mapping entry for EID '153.16.67.1':
7  153.16.67.0/24, via map-reply, record ttl: 1440, auth, not mobile
8    Locator                                  State     Priority/Weight
9    203.181.249.170                          up        1/100
10   2001:200:e000:17::170                    up        2/100
```

APAN's site returns just two IP's, but there could be many more, spread all over the world.

6 Conclusion

At the end of the paper, the research group came up with the following conclusions regarding the management and transition to IPv6 in large companies;

6.1 Stateless/Stateful addressing

Regarding IPv6 addressing within a single corporate site, the research group draws the following conclusions;

- **DHCPv6 seems obsolete**, since stateless IPv6 can provide for the basic functionalities needed with IP addressing. Stateless IPv6 configuration does not require separate DHCPv6 machines to be active and the announcement of the prefix can be done as a very simple service or even by networking equipment. Stateless IPv6 also makes it very easy to move sites, since only the RADVD prefix needs to change and not the full IP address.
- **(D)DNS can be used for addressing** within stateful and stateless networks, in order to get connectivity from the outside world to your internal network. DDNS offers minimal overhead, and it is possible to initiate this from either DHCPv6 (if you have good reasons to use it) or from the client itself. DDNS can register the hostname of a client directly into a BIND9 DNS server and it can keep it's name up-to-date there.
- **Addressing hierarchy** can be achieved by subnetting your IPv6 space into smaller chunks (i.e. /64's), and then by configuring routers or layer 3 switches to announce the prefix for the chunk on the LAN. In this way, you have some information about the physical location of the node, while still maintaining a manageable and flexible address space.
- **Nodes can become dual stack IPv4 and IPv6**, using i.e. regular DHCP for their IPv4 stack and RADVD for their IPv6 stack. The installation of the IPv6 stack and RADVD can be done in a scalable fashion, while your network keeps operating at IPv4.
- **Dual-stack offers the easiest migration scenario**, since there is always a fall-back possible and both IP addresses can keep addressable if you maintain the recommendations mentioned above.

6.2 Connecting IPv4 and IPv6 sites in a scalable way

In the second part of the report, several transition and interoperability scenario's involving LISP were presented. While there are many manual configurations possible to achieve some of the results that are stressed in the chapter, LISP offers these options in a scalable and transparent way.

- **LISP can offer freedom** regarding the location of your servers. By referring subnets of your network through the PITR, you can maintain a consistent namespace but change the destination of this address around.
- **Multihoming is possible with LISP**, since multiple addresses can be pointed towards i.e. a DSL connection or corporate gateway. These addresses are not limited to one protocol family though, so IPv4 and IPv6 addresses can be combined in order to achieve full connectivity. Another advantage of the multihoming is that you are independant from your current ISP, since the traffic can be pointed towards cheaper or more reliable Internet connection, irregardless of their IP address assignments.
- **Traffic engineering enables load-balancing** at any point of your network. By assigning weights and priorities to the RLOCs that are responsible for your network, basic traffic shaping can take place. This will take the basic metrics of priority and weight into account and it can be a transparent or hidden process. Combined with the bullet mentioned above and using multiple Internet connections, very reliable connectivity can be achieved.
- **Current open source LISP software lacks features and active development**. The Cisco implementations on the other hand are already feature complete and actively developed. It is expected that the open source implementations will catch up somewhere this year and that other network manufacturers will start participating in LISP as well. The first tests with this are already taking place on the (currently Cisco dominated) LISP test network.
- **Implementation costs are minimal**, but active development of the open source will be absolutely crucial for the success of LISP.

We hope that this paper gives a little bit of insight into the possibilities and risks that we see with corporate IPv6 deployment. Even though we did deviate slightly from the original research question, we've been happy to learn more about protocols that are on the drawing board right now, and we hope that you can appreciate this.

- Aleksandar and Marek

`aleksandar.kasabov@os3.nl` and `marek.kuczynski@os3.nl`

References

[1] *HOWTO Design a fault-tolerant DHCP + DNS solution* http://shebangme.blogspot.com/2010/09/howto-design-fault-tolerant-dhcp-dns.html (Reviewed on 2011-03-15)

[2] *DHCP for IPv6* RFC 3315, 2003

[3] *Marya Steenman & Jan van Lith - Implementing IPv6 in an Organization* https://www.os3.nl/_media/2004-2005/rp1/report05.pdf?id=archive%3Aresearch_projects&cache=cache (Reviewed on 2011-03-22)

[4] *IPv6 Stateless Address Autoconfiguration* RFC 4862, 2007

[5] *Router Renumbering for IPv6* RFC 2894, 2000

[6] *LISP IETF workgroup* https://datatracker.ietf.org/wg/lisp/charter/ (Reviewed on 2011-03-26)

[7] *BGP table sizes* http://bgp.potaroo.net/as1221/bgp-active.html (Reviewed on 2011-03-26)

[8] *Full BGP table from AS65000* http://bgp.potaroo.net/as1221/bgptable.txt (Reviewed on 2011-03-26)

[9] *Atilla de Groot - LISP+ALT thesis* http://staff.science.uva.nl/~delaat/sne-2008-2009/p06/report.pdf (Reviewed on 2011-03-26)

[10] *LISP at Facebook* http://www.nanog.org/meetings/nanog50/presentations/Tuesday/NANOG50.Talk9.lee_nanog50_atlanta_oct2010_007_publish.pdf (Reviewed on 2011-03-26)

[11] *Facebook's LISP experiences* http://www.networkworld.com/news/2010/061110-facebook-ipv6.html (Reviewed on 2011-03-26)

[12] *Cisco IPv6 migration through LISP* http://lisp4.cisco.com/ipv6_lisp-transition_WP.pdf (Reviewed on 2011-03-26)

[13] *OpenLISP info page* http://www.openlisp.org/ (Reviewed on 2011-03-26)

[14] *LISP Internet Groper (LIG)* https://github.com/davidmeyer/lig (Reviewed on 2011-03-26)

[15] *LISP on Linux* https://github.com/aless/linux-2.6-lisp/wiki/Status (Reviewed on 2011-03-26)

[16] *LISP betanetwork* http://www.lisp4.net/beta-network/ (Reviewed on 2011-03-26)

[17] *Cisco on LISP* http://lisp4.cisco.com/lisp_over.html (Reviewed on 2011-03-26)

[18] *LISP implementations* http://www.lisp4.net/implementations/ (Reviewed on 2011-03-26)

[19] *IPv6 Router Advertisements Option for DNS Configuration* RFC 5006

[20] *Networking Challenges and Resultant Approaches for Large Scale Cloud Construction* Cisco Systems, Inc.

[21] *Locator Identifier Separation Protocol* `http://en.wikipedia.org/wiki/Locator/Identifier_Separation_Protocol`