Jacking-up the Internet Architecture by separating Location and Identity

Luigi Iannone
Senior Research Scientist
Deutsche Telekom Laboratories
Road Map

- Why we need a new Internet Architecture?
- How Loc/ID Split works?
- Ok, let’s split it, but what’s the gain?
- How to modify the stack of a software router?
Road Map

- Why we need a new Internet Architecture?
- How Loc/ID Split works?
- Ok, let’s split it, but what’s the gain?
- How to modify the stack of a software router?
Internet’s Scaling Issues

“It is commonly recognized that today’s Internet routing and addressing system is facing serious scaling problems.”

**BGP’s FIB inflation**

- **Cause of BGP Forwarding Information Base (FIB) Explosion:**
  - PI (Provider Independent) prefix assignment
  - Multi-homing
  - Traffic-Engineering
  - Security
  - ...

[Graph showing the increase in active BGP entries (FIB) from 1994 to 2011, with significant growth in AS6447 (Oregon).]

Source: [http://bgp.potaroo.net/index-bgp.html](http://bgp.potaroo.net/index-bgp.html)
Stable Core exposed to Dynamic Edge

- BGP Update Churn
  - Growing slower than FIB’s size
  - Can have peaks of thousands per seconds
  - Increases need processing power
PA vs. PI Addresses

Provider Aggregatable Addresses

Provider

10.0.0.0/16

Customer

10.0.0.0/24

Provider Independent Addresses

Provider

10.0.0.0/16

10.0.0.0/16

Customer

11.0.0.0/24
Multi-Homing

L. Iannone - Deutsche Telekom Laboratories
Traffic-Engineering (e.g., load balancing)
Some other reasons

- Security:
  - Remember the youtube incident?
  - Big CDNs de-aggregate to avoid prefix hijacking

- Moore’s Law
  - “The number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years”
  - True for high-volume productions, not for low-volume highly specialized hardware like routers

- Rekhter’s Law
  - “Addressing can follow topology or topology can follow addressing. Choose one.”
  - Due to:
    - Overloaded IP Address Semantic
    - The single connection design of the Internet
Let’s be clear:
- There is no hard scaling limit

We need:
- Improve Scalability
- Reduce OpEx (make a cheaper Internet)
Road Map

- Why we need a new Internet Architecture?
- How Loc/ID Split works?
- Ok, let’s split it, but what’s the gain?
- How to modify the stack of a software router?
After more than two years of discussion at the IRTF (Internet Research Task Force)...

“The Research Group has rough consensus that separating identity from location is desirable and technically feasible. However, the Research Group does NOT have consensus on the best engineering approach to such an identity/location split.”

From RFC 6115: Recommendation for a Routing Architecture

- Along with a plethora of proposals:
  - LISP
  - Routing Architecture for the Next Generation Internet (RANGI)
  - Internet Vastly Improved Plumbing (Ivip)
  - Hierarchical IPv4 Framework (hIPv4)
  - Name Overlay (NOL) Service for Scalable Internet Routing
  - Compact Routing in a Locator Identifier Mapping System (CRM)
  - Layered Mapping System (LMS)
  - Two-Phased Mapping
  - Global Locator, Local Locator, and Identifier Split (GLI-Split)
  - Tunneled Inter-Domain Routing (TIDR)
  - Identifier-Locator Network Protocol (ILNP)
  - Enhanced Efficiency of Mapping Distribution protocols in Map-and-Encap Schemes (EEMDP)
  - Evolution
  - Name-Based Sockets
  - Routing and Addressing in Networks with Global Enterprise Recursion (IRON-RANGER)
  - Hierarchical Architecture for Internet Routing (HAIR)
Is it a new idea?

• Not really it has been around for a while (early 90s and may be even before)
  • J. Saltzer, On the Naming and Binding of Network Destinations, RFC 1498, IETF Network Working Group, August 1993.
  • R. Hiden, New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG, RFC 1955, IETF Network Working Group, June 1996.

• More recent work also use the same idea for different purposes:
Split Locators and Identifiers with Map & Encap

- Bindings between ID and Locators: *Mappings*
  - $\text{EID}_x \rightarrow \text{RLOC}^1_{\text{EID}_x}$, $\text{RLOC}^2_{\text{EID}_x}$
  - $\text{EID}_y \rightarrow \text{RLOC}^1_{\text{EID}_y}$, $\text{RLOC}^2_{\text{EID}_y}$
**LISP?**

**Locator ID Separation Protocol**

- Nothing to do with LISP the programming language…….

…. well if you interpret parenthesis as a level of encapsulation then LISP is:

(IP (UDP (LISP (IP (Original Transport Level Packet))))))
LISP Protocol Details: Sending Side

LISP-Database:
- Contains mappings “owned” locally
- Used to select source RLOC

\[ EID_s - \text{Prefix} \Rightarrow \left( RLOC^{1}_{EID_s}, RLOC^{2}_{EID_s} \right) \]
Where does LISP find the Mappings?

Mapping Distribution System:
- Queried to retrieve mappings
- Used to select Destination RLOC

EID\_d - Prefix \(\Rightarrow\) \((RLOC\_1^{EID\_d}, RLOC\_2^{EID\_d})\)
Where does LISP Store the Mappings?

LISP-Cache:
- Queried before the Mapping system
- Mapping system queried only in case of miss
- Used to select Destination RLOC
LISP Protocol Details: Receiver Side

Consistency Checks:
- Check DB: Am I the correct RLOC for the destination EID?
## Few words on Mapping Systems

### Existing LISP Related Mapping Distribution Protocols:

<table>
<thead>
<tr>
<th>Mapping System</th>
<th>Distribution Model</th>
<th>Propagated Information</th>
<th>Aggregation</th>
<th>Sensitive to Churn</th>
</tr>
</thead>
<tbody>
<tr>
<td>LISP-NERD</td>
<td>Push</td>
<td>Entire Mapping Database</td>
<td>No</td>
<td>No (updates on a fixed time schedule)</td>
</tr>
<tr>
<td>LISP-CONS</td>
<td>Hybrid Push/Pull</td>
<td>EID-Prefix</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LISP-EMACS</td>
<td>Pull</td>
<td>-</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LISP-ALT</td>
<td>Hybrid Push/Pull</td>
<td>EID-Prefix</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LISP-DHT</td>
<td>Pull</td>
<td>EID-Prefix</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LISP-TREE</td>
<td>Pull</td>
<td>EID-Prefix</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Road Map

- Why we need a new Internet Architecture?
- How Loc/ID Split works?
- Ok, let’s split it, but what’s the gain?
- How to modify the stack of a software router?
What do we gain with this split?

- **FIB:**
  - shrunk since stab AS do not inject anything in the DFZ
- **PI Addresses:**
  - Just a matter of changing RLOCs
- **Multi-Homing:**
  - Just a list of RLOCs
- **TE:**
  - Just a matter of giving priorities to RLOCs
- **Churn:**
  - Flapping Edge AS are not anymore in the DFZ

- **Additional Benefits:**
  - Mobility Support
  - Path Diversity
How much Path-Diversity do we gain?

Simulations of Multihomed ASes

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Routes</th>
<th>Peers</th>
<th>Pairs</th>
<th>M-h Stubs</th>
<th>M-h Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routeviews</td>
<td>5,750,380</td>
<td>32</td>
<td>496</td>
<td>6.402</td>
<td>29.575</td>
</tr>
</tbody>
</table>

This work has been carried out in collaboration with B. Quoitin INL - UCLouvain
How Traffic Engineering is improved?

- **IDIPS:**
  - Path Selection Service
  - Allows evaluating paths from both network and business perspectives

- **LISP:**
  - Inter-domain Tunneling Service
  - Allows easy and lightweight traffic differentiation

- **LISP + IDIPS:**
  - Flexible inter-domain TE

This work has been carried out in collaboration with D. Saucez INL - UCLouvain in the framework of the AGAVE Project
How much can we reduce the FIB?

- Simulations
  - Synthetic topologies generated with GHITTLE
  - Hierarchical with business relationship
  - 14965 ASes

This work has been carried out in collaboration with B. Quoitin INL - UCLouvain
Large Scale LISP Evaluation

- LISP Emulation
  - Based on PCAP Traces
  - Used BGP Granularity for mappings

Contacted Prefixes

![Graph showing contacted prefixes over time]

- **Total**
- **Bi-directional**
- **Out**
- **In**

Prefixes/Minute

Time of day

APR. 2009

Total

Bi-directional

Out

In
Cache Entries for Vanilla LISP

The graph shows the number of entries per minute and the cache size per minute over the course of a day for different timeout values:

- **1800s timeout** (red line)
- **180s timeout** (blue line)
- **60s timeout** (light blue line)

**Number of entries per minute**

- X-axis: Time of day (02h to 02h)
- Y-axis: Number of entries per minute (0 to 100K)

**Cache size per minute**

- X-axis: Time of day (02h to 02h)
- Y-axis: Cache size (0 to 14MB)

**AUG. 2009**

Different timeout values affect the cache usage, with longer timeouts leading to higher cache sizes and vice versa.
Hit vs. Miss Ratio

Number of hits/misses per minute

Time of day

AUG. 2009
Entries Lifetime

CDF of entry lifetime (seconds):
- 60s timeout
- 180s timeout
- 1800s timeout

Entry lifetime (seconds):
- 1 10 100 1000 10000 100000

CDF values:
- 0
- 0.1
- 0.2
- 0.3
- 0.4
- 0.5
- 0.6
- 0.7
- 0.8
- 0.9
- 1
Will ISPs adopt this technology?

Main Assumptions:
- Incentives to adopt LISP is inversely proportional to the number of “Entries”
  - i.e., “entries” represent the cost
- An entry is the information concerning a Prefix
  - Could be a BGP entry
  - Could be a Mapping in the cache
  - Could be an entry in the overall mapping system

Cost Evolution Formalization

- Model:

\[
C_T = X_L C_{(L \rightarrow B)}(X_L) + (1 - X_L) C_{(B \rightarrow L)}(X_L)
\]

\[
C_T(X_L) = X_L \left[ \alpha X_L + (1 - X_L) + \gamma X_L C_I + \omega X_L \right] + (1 - X_L) \left[ \beta X_L + (1 - X_L) \right]
\]

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_T)</td>
<td>Total cost for a generic ISP (i.e., the average number of entries stored per-ISP)</td>
<td>([0 - \infty])</td>
</tr>
<tr>
<td>(C_{(L \rightarrow B)})</td>
<td>Cost (i.e., the number of entries stored per-ISP) for a Loc/ID Split adopter ISP</td>
<td>([0 - \infty])</td>
</tr>
<tr>
<td>(C_{(B \rightarrow L)})</td>
<td>Cost (i.e., the number of entries stored per-ISP) for a legacy BGP ISP</td>
<td>([0 - \infty])</td>
</tr>
<tr>
<td>(X_L)</td>
<td>% of ISPs that are in Loc/ID Split enabled ISPs</td>
<td>([0 - 1])</td>
</tr>
<tr>
<td>(C_I)</td>
<td>% of ISPs necessary to maintain the Connectivity Infrastructure</td>
<td>([0 - 1])</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>Cache Aggregation Factor</td>
<td>([0 - 1])</td>
</tr>
<tr>
<td>(\beta)</td>
<td>Legacy ISP Proxy Aggregation Factor</td>
<td>([0 - 1])</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>Connectivity Infrastructure Aggregation Factor</td>
<td>([0 - 1])</td>
</tr>
<tr>
<td>(\omega)</td>
<td>Mapping Distribution System Load Factor</td>
<td>([0 - 1])</td>
</tr>
</tbody>
</table>
Total Cost Evolution without Mapping System and naive adopters

Increasing value of \((\alpha + yC)\)
Total Cost
with Mapping System and naive adopters

![Graph showing cost comparison between Pull Model and Push model.]

- Cost (% entries) values:
  - Pull Model: ~0.093
  - Push model: ~1.063

- X-axis: % Loc/ID Split Adopters
- Y-axis: Cost (% entries)
Road Map

- Why we need a new Internet Architecture?
- How Loc/ID Split works?
- Ok, let’s split it, but what’s the gain?
- How to modify the stack of a software router?
How to implement an open-source LISP

▪ Use an open-source OS: FreeBSD

▪ Using Virtual Interfaces
  ▪ Drawback: Static address assignment

▪ Modify NAT or Firewalls:
  ▪ Drawback: Static rules, API toward the control plane difficult to implement

▪ Native implementation:
  ▪ Advantage: Clean Architecture design
  ▪ Answers to: “What would a protocol stack look like if LISP (or any other Loc/ID split approach) would be a fundamental piece of the Internet Architecture?”
Main Modules:
- Encap/Decap
- Mapping Database and Cache
- Mapping Distribution Protocol Support

Mapping Socket:
- Support for any Mapping distribution protocol
- Flexible
- Extensible
LISP DB + LISP Cache = MapTable

- MapTable is a Radix Tree
  - Two MapTables installed: IPv4 & IPv6
- Merges both LISP-Database and LISP-Cache
  - Different tag
- Search Key: EID-Prefix
- RLOCs: stored as chained list
  - Ordered by Priority and Weight
Normal protocol stack operation

```
Transport Layer
ip6_output()
```

Diagram:
- Data-Link
- `ip6_output()`
LISP enabled Protocol stack operation

Diagram:
- `lisp_output()`
- `lisp6_output()`
- `ip_output()`
- `ip6_output()`
- `Transport Layer`
- `Data-Link`
LISP enabled Protocol stack operation
Measurement Setup

- **Protocols:**
  - IPv4
  - IPv6

- **Scenarios:**
  - Routing no LISP stack
  - Routing with LISP stack
  - GRE Tunnel no LISP Stack
  - GRE Tunnel with LISP Stack
  - LISP

- **Tools:**
  - Tcpdump (in and out interfaces of Enc & Dec)
  - Iperf (Src & Dst)
Latency on the Encapsulator
Latency on the Decapsulator
REASONS WHY PEOPLE WHO WORK WITH COMPUTERS SEEM TO HAVE A LOT OF SPARE TIME...

Web Developer
- "It's uploading"

Sysadmin
- "It's rebooting"

Hacker
- "It's scripted"

3D Artist
- "It's rendering"

IT Consultant
- "It's your problem now"

Programmer
- "It's compiling"