Forming An IPv6-only Core for Today’s Internet

SIGCOMM’07 IPv6 Workshop
2007.08.31 Kyoto, Japan

Acknowledgement

• Bruno Quoitin
  – reviewed our previous works and suggested LISP, CRIO and works of Cedric de Launois for the metric of multihoming
  – reviewed our current work and commented what is significant, what is important
• Ang Li (UCI)
  – completed the early work for IPv4-IPv6 address mapping and packet translation
• Prof. Xing Li (Tsinghua University)
  – proposed using address mapping to scale IPv4-IPv6 packet translation within a domain
  – authorized using CERNET2 platform to do experiments
• Prof. Jianping Wu (Tsinghua University)
  – proposed building IPv6-only CERNET2 as a strategic design of deployment (but w/o technological verification)
Outline

• Introduction
• A Big Picture for IPv6-only Core Internet
• Mapping Lookup Service for IPv6-only Core
• Evaluations
• Conclusion

Today’s Internet: IPv4

• The problem of scaling
  – Global IPv4 routing table is growing dramatically
    » draft-iab-raws-report
  – More are coming: multi-homing / traffic-engineering

• PI multi-homing
  – PA is not possible due to lack of addresses, unless ...

• ID/Loc separation
  – LISP
    » draft-farinacci-lisp
  – CRIQ
    » Zhang, ICNP’06
Is IPv6 the Future?

- IPv6-only backbones are deployed
  - e.g. CNGI-CERNET2, 6WiN
- Global routing
  - Well aggregated up to now
  - PA multi-homing
    - has been quantitatively proved superior to PI mode
      - Cedric de Launois et al., Computer Networks, 50(8), 2006
      - with a new path diversity metric for multi-homing
- ID/Loc separation
  - shim6
- Problems:
  - no application contents, no mature services for transactions, no affiliation
    - IPv6-only networks are almost useless for users
    - Content resources are still stored in IPv4 end systems
    - People still visit each other over IPv4, even NAT troubles end-to-end
    - Networks are insistent of not migrating from IPv4 to IPv6

Motivation: Combining Advantages of both IPv4 and IPv6

- IPv6
  - has good nature in aggregation and multi-homing
  - plays role of locator
- IPv4
  - has resources and applications
  - plays role of identifier
- Conditions
  - Packet alternation between two different protocols
    - with either encapsulation or translation
  - Address mapping
    - Either encapsulation or translation involves address mapping.
    - Challenge: How do we make it with as less as possible states, and scalable?
Previous Works

- Carrying IPv4 traffic with IPv6-only backbone
  - Chen et al., NOMS 2006
    - Approaches
      - proposed a novel, stateless mapping scheme
        - a prefix-specific mapping
      - implemented the “edge router” - the box that does stateless translation
        - ever deployed and tested in real networks
          - [http://v6s.6test.edu.cn/](http://v6s.6test.edu.cn/)
          - [http://v4s.6test.edu.cn/](http://v4s.6test.edu.cn/)
          - [http://202.38.118.4/](http://202.38.118.4/) - CERNET2 6PlanetLab platform manager
    - Limitations
      - A solution for transition instead of more scalable routing in multi-homing environment
        - not feasible index service for address mapping has been proposed
        - mappings are basically applied within AS

IPv6-only Core Internet

- An architecture
  - IPv4 customer networks; IPv6 core providers
- A mapping lookup service
  - distributed index system
  - self-organized
- Evaluations
  - concern: current IPv6 connectivity is quite poor, then whether customer networks, if they change to connect IPv6 providers only, can get the benefits in multi-homing now?
Big Picture of Internet with IPv6-only Core

- Foundation stones
  - Edge Router
    - regular router for native IPv6
    - translator at boundary
    - “entry ER” vs. “exit ER”
  - Prefix-specific address mapping
    - /32 assumption
  - Mapping lookup service (MLS)

Example
On Multi-homing

• With multiple address mappings
  – Customer gains the tolerance of network failure
    • e.g. host in C5 having access to a peer in C3
      – entry ER: E6 who knows the mapping
        » 3.3.0.0/16 => 2001:2001:ff03:300::/56, w = .7
        » 3.3.0.0/16 => 2001:2002:ff03:300::/56, w = .3
      – exit ER: E3 is preferred, E4 is alternative
    • Link E3-C3 is broken, then
      – ICMPv6 “destination unreachable”
      – E6 can be aware of that
      – E6 use the backup mapping
  – Advantage
    • without need of a “shim” layer in the stack

On Multi-homing

• With multiple address mappings
  – Customer gains the load balance
    • e.g. host in C5 having access to a peer in C3
      – entry ER: E6 who knows the mapping
        » 3.3.0.0/16 => 2001:2001:ff03:300::/56, w = .7
        » 3.3.0.0/16 => 2001:2002:ff03:300::/56, w = .3
      – exit ER: E3 with p = .7; E4 with p = .3 (p: Probability)
  – Further exploration: traffic engineering by customers
    • can ER6 tune the mapping according to ER-to-ER measurement?
      » proposed by Bruno Quoitin
Why Translation Rather than Encapsulation?

- Pros
  - 3.3.111.22 can be accessed by
    - native IPv6 nodes via translation
    - native IPv4 nodes via double translations
  - 2001:2101:ff01:0101:2300:: can be accessed by
    - native IPv6 nodes directly
    - native IPv4 nodes (w/ destination address 1.1.1.35) via translation

- Cons
  - lose 'some' end-to-end

- but not so bad as NAT-PT, because
  - the mapping is stateless
  - each Edge Router can do the right job

Challenge: Look Up the Mapping

- Possible Solutions
  - BGP extension
    - likely to carry all IPv4 entries (but without AS path) in IPv6 RIB
  - Auto-discovery
    - LISP suggests ICMPv6 extension can help
  - Distributed database
    - LISP suggests using DNS and DHT but not well done yet
  - Our try: self-organized distributed index system
    - Considerations
      - load-balance among MLS (Mapping Lookup Service) facilities
      - lowest impact to IPv6 global routing
      - incremental deployment
MLS Overlay

- Self-organization
  - Assumption:
    - providers who serve IPv4 customers are **willing** to contribute a server in MLS
  - Distribution of mapping information:
    - each server stores mapping for a contiguous IPv4 address block
  - Growth:
    - split the IPv4 address space once a new server joins
      » CAN’s idea

![Mapping Overlay Growth](image-url)
Mapping Lookup in *Historical Neighbour Table*

– for retrieval, registration, update and withdrawn, it is necessary to find where the mapping for a certain identifier is (or should be) stored.

**Mapping Lookup**

- Recursive trying *longest-match* algorithm

  $x$: the target IPv4 address;
  
  $i$: current or randomly selected MLS server, with IPv6 address $S_i$
  
  $X_i$: the block that $i$ takes care of;
  
  $A_i$: $i$’s neighborhood, with each neighbor’s prefix and the block it takes care of mapping for.

  $$\text{FindPref}(x, i) \{$$

  \[ \text{If } (x \land X_i = X) \text{ Then} \{
  \begin{align*}
  & \text{If } (i \text{ stores matched entry(-ies) }) \text{ Then} \\
  & \quad \text{Find the longest-matched one, the } p(x); \\
  & \quad \text{Return all the entries } p(x) = \{p_i, w_i}\} \\
  & \quad \text{Else Return NULL;}
  \end{align*}
  \]

  $$\text{Find } j \in A_i \text{, so that } X_j \text{ matches } x \text{ or } X_j \text{ has the shortest prefix}$$

  $$\text{Return FindPref}(x, j);$$

  $$\}$$
On the initial selection of MLS server

- select the local MLS server
  - if ER has no idea about other MLS servers
  - MLS servers have special suffix for statelessness
    » e.g. ::1:4664
- randomly select MLS server among all servers
  - if ER get a cache of MLS server list
  - it is quite ok should the list is incomplete
  - how does an ER get the list know?
    - from its local MLS server
    - from BGP table, provided each MLS contributor announces their /32
      prefix with a special community number,
    » e.g. <ASN>:4664

Comparison to Inverse DNS

- The lookup algorithm is similar to inverse DNS but
  - in smaller granularity
  - applying longest match for the ID/Loc mapping
    e.g.
    202.38.112.0/20 --> 2001:250::/32
    202.38.118.0/24 --> 2001:da8::/32
    then
    \[ p(202.38.118.4) = 2001:da8:ffca:2676:400:: \]
- Disadvantages
  - Authorization/authentication is to be designed but has been mature in DNS
Redundancy Consideration

- Backup (secondary) server
  - like backup DNS server mechanism

- Backup mapping by neighbors in IPv4 address space
  - like Pastry’s “leaf-set” idea
  - still in open debate ...

Performance Consideration

- Concerns
  - Retrieving mapping in MLS is a time-consuming job!

- Strategies
  - ER local caches
    - cache for mapping entries just retrieved
      - accelerates packet translation and delivery, not always retrieves the MLS overlay
      - optimization
        » most frequently used entries first (a good heuristics?)
  - cache for the MLS server just visited
    - accelerates retrieval without always search along the MLS tree
Performance Consideration

- Concerns
  - Retrieving MLS is a time-consuming job!
- Possible strategies
  - MLS overlay relaying: fast delivery
    - Intuition: who maintains mapping for a destination can relay for it
      - e.g. i* (with IPv6 address S_i*) maintains p(x)
      - entry ER can translate destination IPv4 address with S_i*’s prefix p* instead of p(x) before entry ER get p(x) from S_i*
  - Concerns
    - additional traffic load of the MLS overlay
    - detours of path
    - transit policy of relaying AS
    - Just drop the packet if no mapping found in local cache

Evaluation

- What if today’s dual stack providers stop their IPv4 backbones?
  - Methodology
    - using current IPv6 BGP table:
      - suppose 733 IPv6 AS as the Core
    - using current IPv4 AS relationship (from CAIDA):
      - 305 AS in among the IPv6 core also provide IPv4 connection
      - 3991 IPv4 AS directly connect to providers above, as the customers
  - Metric
    - Path Diversity: defined by Cedric De Launois, ICNP’06
Evaluation

• Guess it before doing calculations
  – IPv6 PA multi-homing must be better than PI multi-homing
    • It is true for same topology running either IPv6 or IPv4, as has been proved by Launois et al.
  – Current IPv6 providers are connected far less dense than among IPv4 core providers
  – then, is it attractive for customer networks to connect to IPv6 core providers?
Observations

- Single-homed customers gain little when providers migrate to IPv6-only
  - explanation: current IPv6 interconnectivity is quite poor because IPv6 deployment is immature currently
- Dual-homed
  - IPv4
    - less than 40% have path diversity over 0.2
  - IPv6
    - more than 70% over 0.2
- **Remark**
  - Even when the IPv6 deployment in early stage, migrating to IPv6-only core is beneficial to multi-homed customers
Further Calculation

- Local cache: how big is enough?
  - hope it not close to the size of IPv4 FIB
- Methodology
  - using real traffic data of destination
    - CERNET international outbound
    - 24 hours: ~ 8 billion packets (excluding unreachable destinations according to IPv4 global routing table)
- Metric
  - cache-hit ratio in stable state
  - packet count before stabilization

**Cache Size vs. Cache Hit Ratio**

![Graph showing cache size vs. cache hit ratio](image)
Conclusions

• Contributions
  – An architecture of using IPv4 as ID while IPv6 as Loc
    • share the PA multi-homing benefits to IPv4 communities
  – Quantitative evaluation for building IPv6-only backbone networks, in the term of path diversity for customers

• New works are raised
  – How do we specify/use the weight for mapping?
  – A feasible and scalable mapping lookup service?
What If We Design IPv6 Over Again?

myhost: ~ m32$ telnet 1020:3040:5060:7080

Universal ID/Loc Mapping Facility

0x0000: 6000 0000 002c 0640 2001 0200 0000 ff10
0x0010: 020d 93ff fe89 ab93 2001 0da8 0200 9002
0x0020: 1020 3040 5060 7080 c108 0017 0bb7 4aa3
0x0030: 0000 0000 b002 ffff 703c 0000 0204 05a0
0x0040: 0103 0300 0101 080a 2b9a 619d 0000 0000
0x0050: 0402 0000

What If We Design IPv6 Over Again?

myhost: ~ m32$ telnet 202.38.118.4

Universal ID/Loc Mapping Facility

0x0000: 6000 0000 002c 0640 2001 0200 0000 ff10
0x0010: 020d 93ff fe89 ab93 2001 0da8 0200 9002
0x0020: 0000 0000 ca26 7640 c1e 0017 909b d196
0x0030: 0000 0000 b002 ffff 1edb 0000 0204 05a0
0x0040: 0103 0300 0101 080a 2b9a 67ea 0000 0000
0x0050: 0402 0000
Thanks!

- Questions?