Identifier Locator Addressing with IPv6

Network virtualization without encapsulation

draft-herbert-nvo3-ila-01

Tom Herbert <tom@herbertland.com>

Background: Why IPv6 in the DC

- Nominally, to scale number of hosts
- Limits of IPv4 addressing
  - Hierarchical address allocation can exhaust 10/8
  - Renumbering becomes a greater effort then transitioning to IPv6

Forward looking benefits

- Capability to address more than just physical hosts (e.g. ILA)
- Flow label for ECMP
  - HW no longer needs to parse L4 protocols
  - Allows us to deploy protocols other than just TCP and UDP (like IPIP, GRE/IP, MPLS/IP, etc.)
- Extension headers
  - Assuming reasonable HW support in switches...

Deployment @Facebook

- Migration for internal traffic >95% complete
- Most services access network through libraries simplified migration
- Issues encountered
  - SW: Java still has issues!
  - SW: Performance, routing code in Linux stack
  - HW: Old rack switches had problems with ND
  - HW: Performance, functional issues w/ NIC offloads

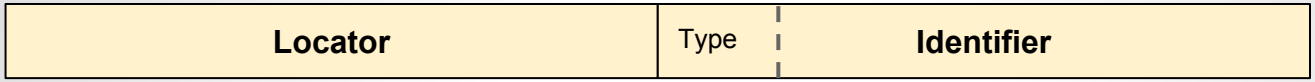
Motivations for ILA

- Address per task
  - Each task gets its over port number space
  - Greatly simplifies task scheduling, service lookup
- Network virtualization
  - Network virtualization without encapsulation
  - Container virtualization and task mobility
  - Alt. implementation of virtual networking for VMs

Solution

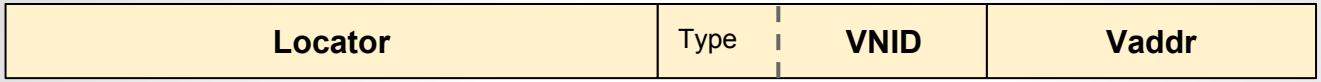
- Split IPv6 address into identifier (who) and locator (where) ala ILNP
- Each task gets its own unique identifier
- Mapping identifiers to locators
- If task migrates between hosts, its locator changes but its identifier does not
- When not migrating, data path is essentially same as before

Address split



- Locator
  - 64 bits identifier of physical hosts
  - Routable
  - Not used as connection endpoint
- Identifier
  - 64 bit logical endpoint address of virtual node
  - Routable to an translator (NVE)
  - Used as connection endpoint
  - Typed to allow different modes

Network virtualization use cases



- Embed VNID in ILA address
  - Potentially eliminate need for encapsulation
  - No place to put security to authenticate VNID, so intra VN use might be limited
- Allows VM to common DC service, or Internet w/o stateful NAT or encapsulation
- Allow two VMs to communicate directly under policy w/o NAT

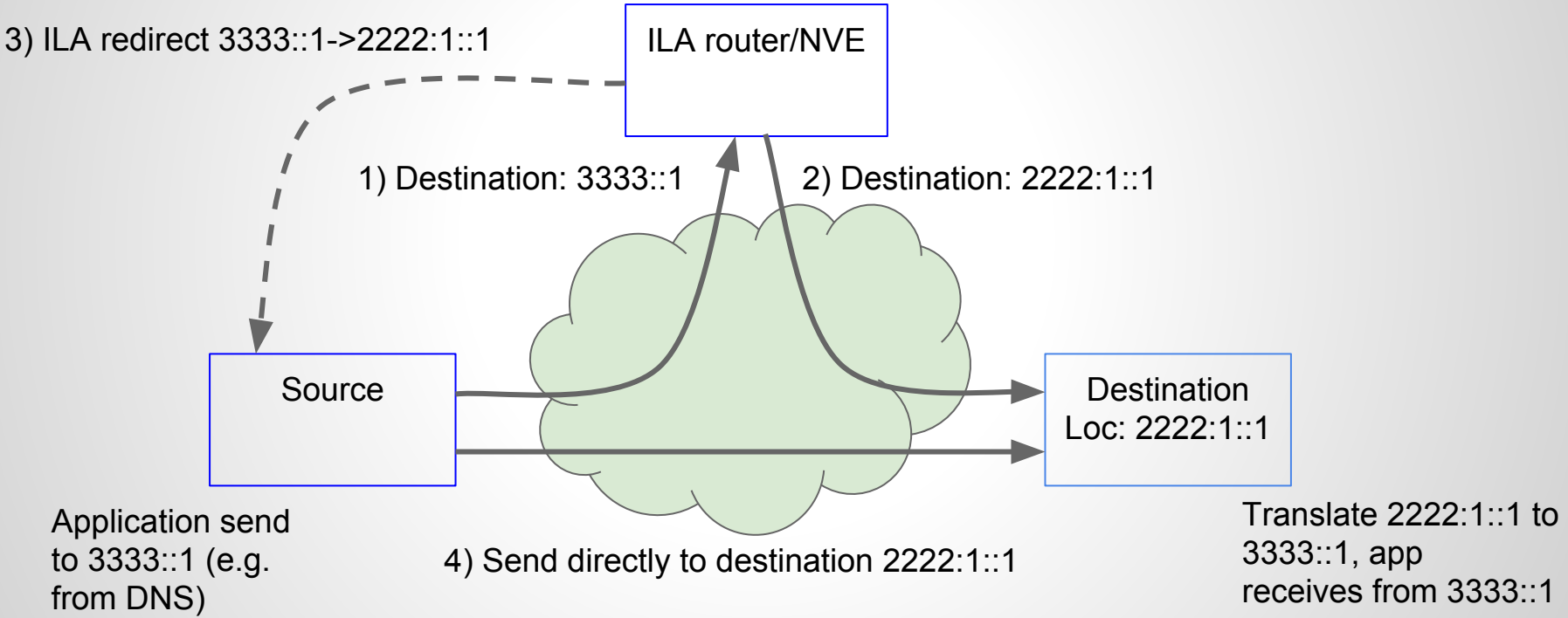
Details

- Need to map identifiers to locators
  - Same problem of mapping Vaddr to Paddr in NV
  - Use NVO3 control plane to distribute mappings
- User visible representation of ILA address (used to connect, in DNS, etc.)
- Generate identifier for each task, duplicate address detection

Architecture

- NVo3 architecture can be applied
  - NVEs in network can perform translation
  - Route packet to NVE which translates and forwards
  - “ILA redirect” can be sent back to host to eliminate triangular routing
- Translations
  - Unicast: Set destination to ILA address
  - Multicast: Need to modify source address

Flow example



Status

- I-D 01 posted
- Integrated into Linux stack
- Commencing canary testing
- Design for address per task assignment
- Design/implementation of ILA router