

# Multihoming L3 Shim Approach

draft-nordmark-multi6dt-shim-00.txt

Erik Nordmark

[erik.nordmark@sun.com](mailto:erik.nordmark@sun.com)

Multi6 design team: J. Arkko, I. van Beijnum, M.  
Bagnulo, G. Houston, E. Nordmark, M.  
Wasserman, J. Ylitalo

# Overview

- No separate ID name space
- Placement of the L3 shim
- Deferred context establishment
- Assumptions about the DNS
- Issue: receive side demultiplexing
- Open Issues

# No Separate ID name space

- ULID – upper-layer ID
  - The 128-bit quantity which is used above the shim layer
- The set of locators (from AAAA records) are candidates for being the ULID
- The ULID is what's seen by TCP, applications etc

# Placement of the L3 shim

- Above the IP routing sublayer, below the IP endpoint sublayer
  - Below fragmentation, IPsec
- [TBD add figure that is in draft]

# Deferred Context Establishment

- Three events occurring at different times
  - Initial contact e.g., some TCP connection to a peer
  - Deciding to setup multi6 context state
    - Based on local policy – port numbers, #packets sent, etc
  - Rehoming the connection after a failure
- Also need to handle failures during the initial contact
  - Base case: punt to the application layer to try different ULID
  - Possible to optimize by having shim do something?

# Assumptions about the DNS

- None
  - A FQDN might be for a service or for a host
  - The FQDN lookup returns a set of potential ULIDs which will be tried by the application until one is working
  - Then the peer will pass its set of locators during the (deferred) context establishment
- Desire to optimize failure during initial contact (by having the multi6 shim try different ones instead of the ULP/application) makes this more complex

# Receive side demultiplexing issue

- Receiver needs to be able to correctly rewrite IP address fields before passing to ULP
- Example: ULID A communicates with ULID B and C
  - Later discovers that ULID B has locators B and C, and ULID C has locators B and C i.e., its the same host
  - Locator B fails
  - The peer will receive packets from locator A to locator C
    - Some of which need to be rewritten to ULID B and others which need no rewrite

# RSD: prevent receive side confusion

- Each locator is only used with a single ULID
- Means that a host with e.g. 3 prefixes would have 3 ULIDs and 9 locators
  - Each locator is used with only one ULID
- The locator will uniquely identify the ULID at the receiver
- Example: Prefixes P1, P2
  - ULIDs P1|IID1 and P2|IID2
  - Extra locators P2|IID21 and P1|IID12
  - P2|IID21 is remapped to received to ULID P1|IID1

# RSD: carry additional info

- Some “context tag” in each packet that needs to be rewritten by receiver
  - The tag exchanged during context establishment
- Where in the packet does it go?
  - Reusing flow label field?
  - A new extension header?
- Former has some complexity due to overloading, but not packet overhead
- Latter implies an extra 8 bytes in the packets after a locator failure

# Open Issues (from mailing list)

- Carrying flow label across the shim
  - Has impact on protocols which do explicit signaling of flows
- Flexibility when using flow label for receive side demux
- Check against “things-to-think-about”
- ICMP error demultiplexing needs to be discussed
  - The sender needs to be able to demux based on the first 64 bytes of the packet that it sent

# Open Issues (2)

- More clarity about ULAs
  - Intent was to say that centrally assigned ULAs are interesting because applications could use a maintained reverse tree for such ones
  - Handling unreachable ULIDs during initial contact will make all ULAs “work”
    - Performance is an issue
    - Perhaps default address selection avoids the performance pitfall?