FA-IINAS: Functional Addressing (FA) for internets with Independent Network Address Spaces (IINAS)(*)

Toerless Eckert, Futurewei USA (tte@cs.fau.de)

1/08/2021

SPRING

(*) Will pay for better name/abbreviation: Coffee, Pizza, Sushi!

Motivation

- Draft introduced IETF111 RTGWG / INTAREA / OPSAREA
  - What can we do better with variable length addresses ?
  - Draft focusses on answers for complex limited domain networks (industrial etc..) – Novel (NAT style) Address prefix rewrite rules
    - See IETF111 slides and draft
  - Draft mentions simpler benefits uses for current SP networks
    - But no time to detail in draft – hence this slide deck

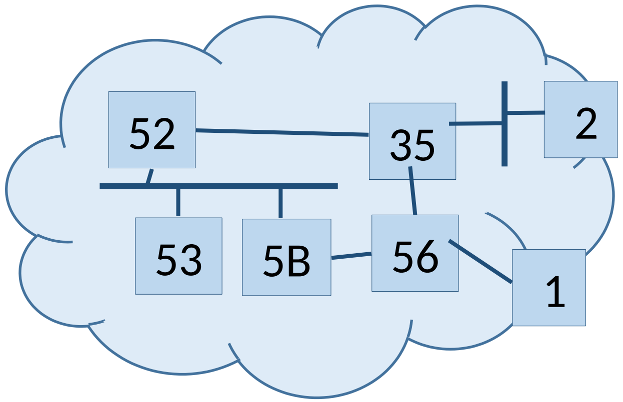
The SPRING question

- Imagine customers (6G, Industrial, cities, IoT) without 20 year IPv6 vs. MPLS preference
- Imagine we had an RFC trying to compare both of them
  - *Would we even get consensus given the strong preferences contributors have either way.*
- IMHO most obvious feedback:
  - Why does IETF ONLY continue to enhance two parallel forwarding planes ?
  - Both have downsides the other solves better.
  - One common forwarding plane with best of both worlds would be logical!
  - Avoids duplication of expertise, development, HW
  - Elimitates limiations
- IMHO: This is the SPRING question
  - We have a unified SR architecture
  - But we do not have a unified forward plane option
    - Nothing bad to first use MPLS/IPv6. They are both great phase 1 options! To merge on common architecture.
- Current IETF WG options detrimental to consider merged/common forwarding option!
  - Both 6man and MPLS can only evolve ”so far” – maintain legacy (architecture)
  - Imagine QUIC would have been worked out in TCPM WG.
  - That legacy is DIFFERENT from operator requirements
    - Thin waist, low operational churn / complexity / limitations, agile incremental innovation
- SPRING should have an interest to see better/unified forwarding plane options explored

Address structure

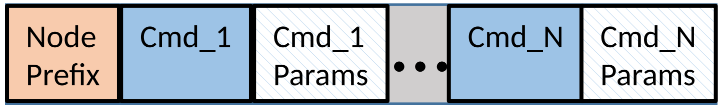
Address allocation within domain:

- Per-node prefix disjoint – nodes OWN prefix and any longer address
- Address space managed by IGP
- E.g.: granularity of prefix length is 4 bit
  - Makes human operator life simple when dealing with hex address prefixes
- Addresses may or may not be mapped to IPv6 addresses
  - TBD: When it helps to simplify the deployment!



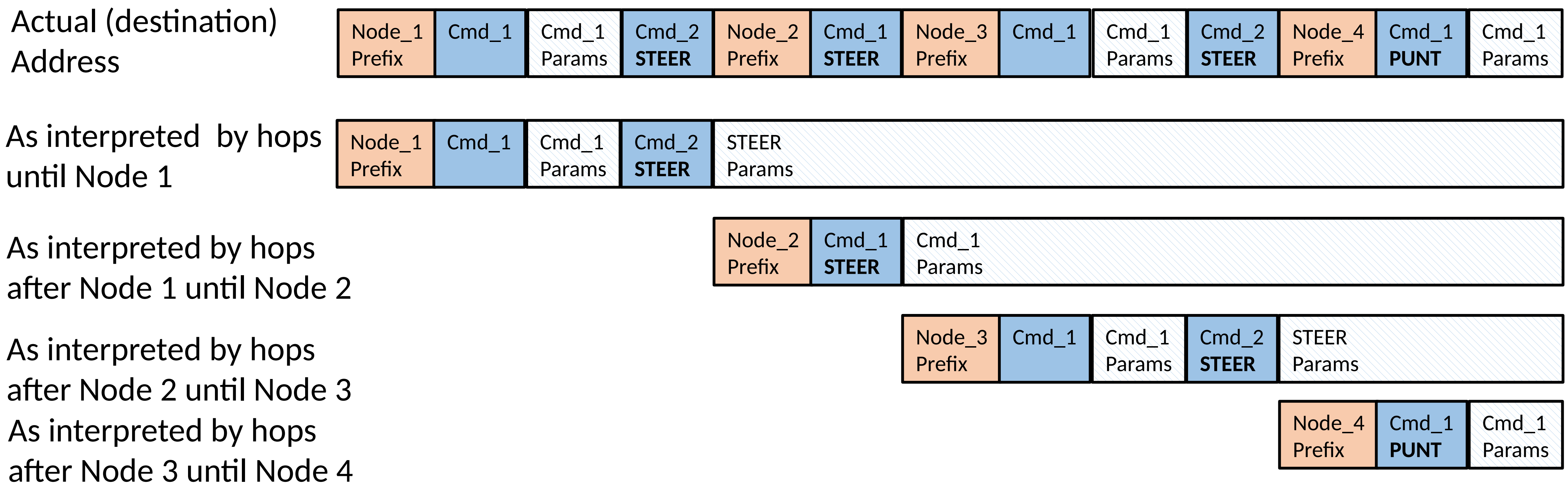
Address Prefix followed by sequence of commands with optional parameters

- Example: 4 bit long. ‘0’ = receive, ‘1’ = receive into VPN-parameter , ‘2’ = oam-punt, …
- Could / should be superset of existing SR commands
  - That are not just IPv6/MPLS encap specific (like PSP).
- Command address space extensible (like address prefixes):
  - Common commands 0..E (4 bit),
    less common commands Fxx – longer prefixes
- Always optional, only included in address when needed
  - No limit to 64 bits, no need to waste 64 bits when no commands needed



Steering

- Just another command, e.g. ‘3’ – ’loose steer’, ‘4’ - ‘strict steer’ (no IGP reroute).
- Example with loose steering
  - Intermediate hops forward based on Node prefix of address (as usual)
  - When Node prefix is owned by node, Cmd chain is executed
- Example compact encoding:
  - 4 * 16 bit node prefixes, 6 * 4 bit commands, 3 * 8 bit ‘Cmd_1 Params’: 112 bit !



Address encoding / interpretation

- How to achieve sequential interpretation along the path ?
  A. MPLS style: “Address prefix stripping’
     ‘Steering’ command discards prefix (up to its own command code),
     keeping only parameter suffix

     Address are as shown in prior slide - getting shorter towards destination

  B. SRH style: ‘Address Interpretation Offset’ (AIO) field in packet
     ‘Steering’ command increases this offset. Fixed length, e.g.: 8 bit (unit nibble).

Address With AIO

AIO	Node_1 Prefix	Cmd_1	Cmd_1 Params	Cmd_2 **STEER**	Node_2 Prefix	Cmd_1 **STEER**	Node_3 Prefix	Cmd_1	Cmd_1 Params	Cmd_2 **STEER**	Node_4 Prefix	Cmd_1 **PUNT**	Cmd_1 Params

Address encoding / interpretation

- New routing risk because of variable length prefixes ?!
  - What happens if encoded Node_1 length does not match routing table ?
    - Problem not had in MPLS / SRH – fixed 20/32 and 128 bit units of decoding.
  - Include new packet field ‘Prefixlength’ (PL)
  - Minimum check: when node thinks address (at AIO offset) is its own prefix:
    - Compare own lookup prefix-length with packet PL field
    - Upon mismatch, stop processing packet, raise error
  - AIO+PL =~ 16 bit (10 + 6 bit ?!)

Address
With AIO/PL

Address encoding / interpretation

Command code consistency across network is important!

Options

1. Fixed by standard (strongest consistency)
2. Preconfigured across network (fairly strong consistency)
3. Dynamically user assigned, managed by IGP (eventual consistency)

Support all three options.

- Standardize address ranges / prefix length.
  - Removes inconsistent prefix length without requiring PL field for commands
  - Best option for commands (but would limit flexibility too much for node prefixes)
- Example standardization with 4 and 12 bit command codes
  - 4 bit for most important/often-used commands:
  - Fixed by standards 0 - 6, F00-F6F
  - Consistently preconfigured 7 - A, F70-FAF
  - Dynamically signaled B - E, FB0 – FEF
  - Standard easily extended with >= 16 bit codes later.

Example header *(motivational!)*

32 bit fixed header + address(es)

```
Strip & enhance IPv6 header:

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|VE |ECN| DestAddrLen   | SrcAddrLen    | Hop Limit     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Destination Address / ”Delivery Instructions”...
+-+-+-+-+-+-+-+-+-+-+-+ . . .
| Source Address / „Return Instructions” ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Version = not 4 or 6 !  VE – 2 bit to identify addtl. E.g.: extension headers

Functionality covered by Destination Address:
   Next protocol, SRH segment list/destination functions/TLV

Source address optional (length can be zero)
   May not be used in limited domains operating like MPLS

Moved to TBD QoS/”Service-Level-Objectives” extension header (with more features)
   DSCP, Flow Label - Not universally used, waste in base header.
```

Functionality beyond addressing

- IMHO this is a vision for a 'best of both world' future addressing

- BUT: Many more differences in MPLS vs. IPv6 forwarding
- Example: Inband (IPv6) vs. OOB payload + header (MPLS) MTU discovery
  - Inband is an application pain.
  - OOB is only easy without CsC (inter-provider tunneling)
    - Sell MPLS transport network connectivity to 5G/6G radio tower operator Which also wants to do MPLS
  - Should we not want to innovate here independent of addressing ?

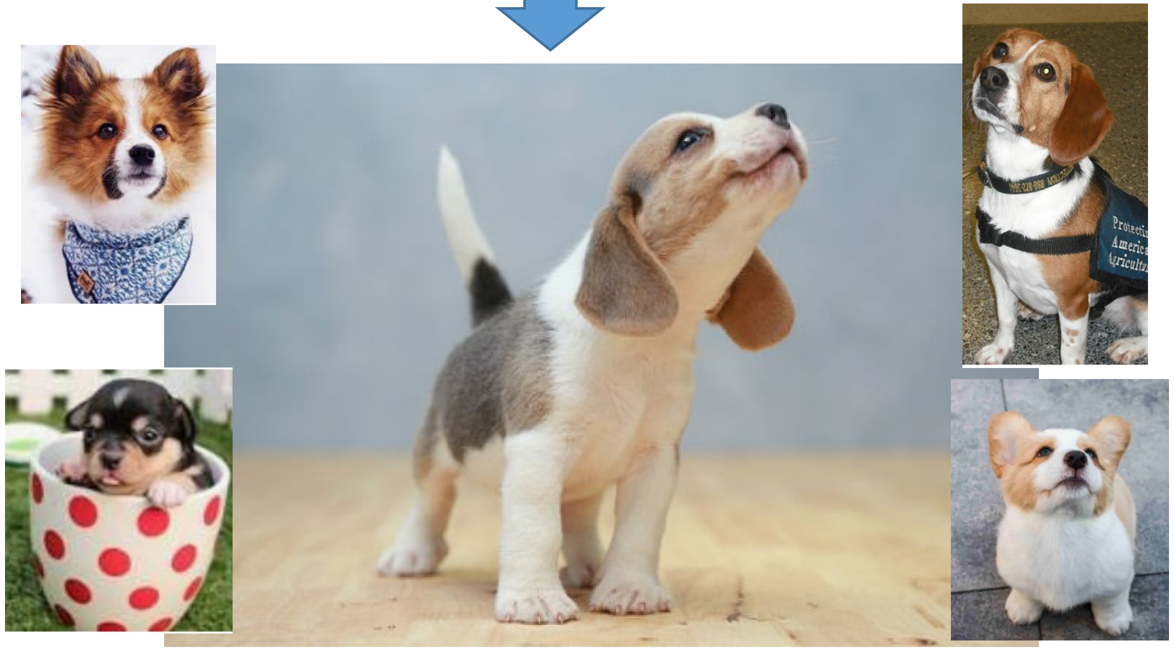
- Backward compatibility ?!
  - The only question is: HOW MUCH
  - Can easily map SRv6 and/or MPLS addressing into new packet encoding
    - And even keep/reuse all or either control plane (FA-INAAS with extended MPLS or SRv6 control plane)
    - Add more addresses for new semantics
  - Design so that we _could_ support all SRv6/SRH and MPLS functionality
    - BUT: standardize/introduce old features only when needed – to eliminate bloat.

Domesticate addressing

NAT: IPv4, adressing, RFC1918, ULA,…
26++ IPv4/IPv6 transition (NAT) mechanisms:
https://en.wikipedia.org/wiki/IPv6_transition_mechanism

But benefits from **functional structures in IPv6** already (scopes/zones, unicast prefix multicast, RP,…)

Experiences with **address processing** of MPLS and **source-route** processing in SR-MPLS/SRH



FA-IINAS: **Multi-purpose functional address processing**

The End