A Simple BGP-Based Routing Service for the Aeronautical Telecommunications Network (with AERO and OMNI)

IETF 111 rtgwg session (July 28, 2021)

Fred L. Templin (The Boeing Company)

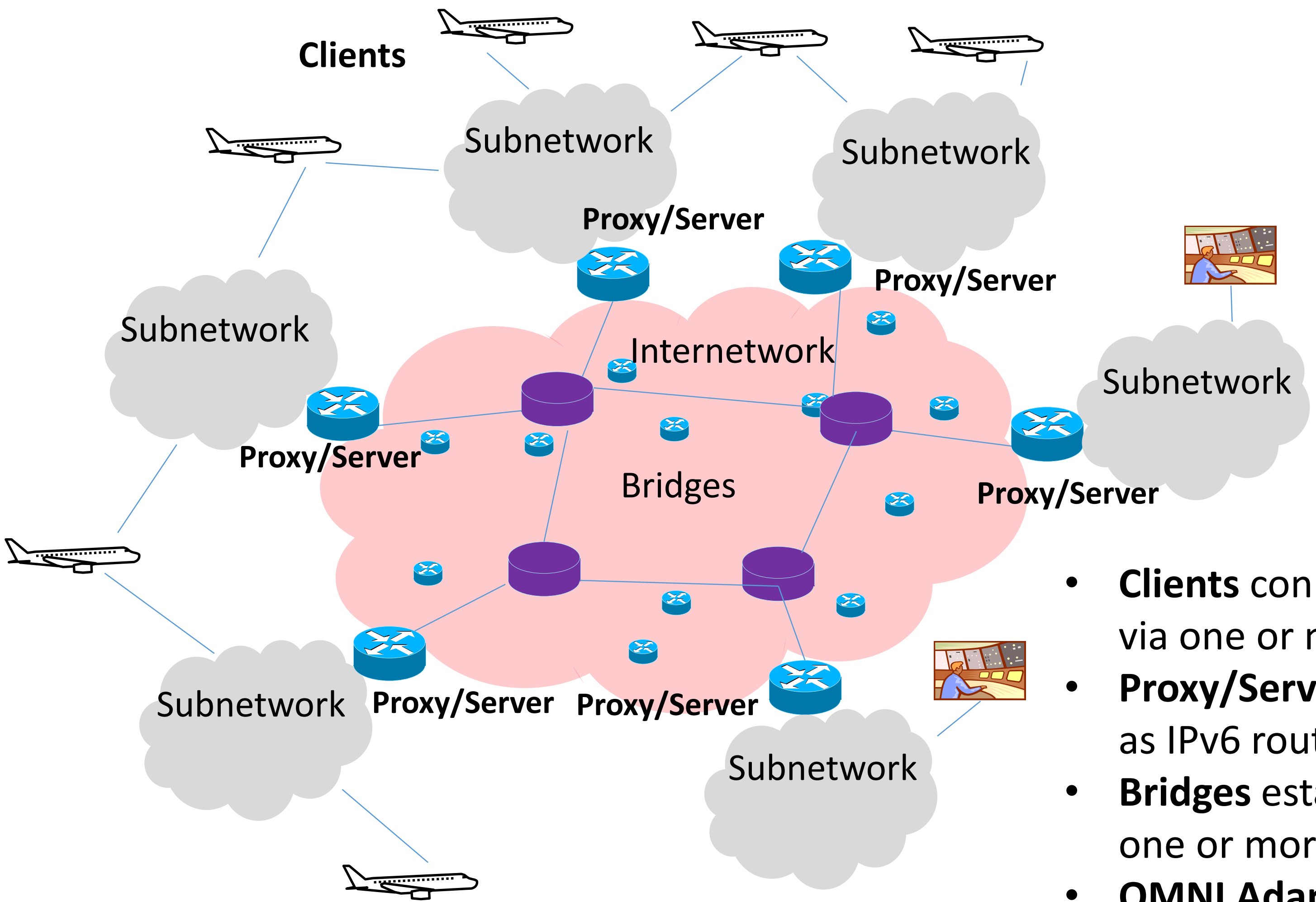
fred.l.templin@boeing.com

fltemplin@acm.org

Document Status

- “A Simple BGP-based Mobile Routing System for the Aeronautical Telecommunications Network”
  - BGP-based “spanning tree” configured over one or more Internetworking “segments” based on Non-Broadcast, Multiple Access (NBMA) interface model and IPv6 Unique Local Address (ULA) prefixes
  - ASBRs of each segment in a “hub-and-spokes” arrangement, with peering between adjacent segment hubs
  - IETF rtgwg working group item since August 30, 2018 - coordinated with International Civil Aviation Organization (ICAO) Aeronautical Telecommunications Network (ATN)
  - https://datatracker.ietf.org/doc/draft-ietf-rtgwg-atn-bgp/
  - **Work ready for IETF rtgwg WGLC**
- “Automatic Extended Route Optimization (AERO)”
  - Route optimization extensions that establish “shortcuts” to avoid strict spanning tree paths
  - Mobility/multilink/multinet/multihop support based on agile “hub-and-spokes” Client↔Proxy/Server model
  - https://datatracker.ietf.org/doc/draft-templin-6man-aero/
  - **Work ready for IETF adoption**
- “Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces”
  - Single NBMA network interface exposed to the IP layer with fixed 9KB MTU, but configured as an overlay over multiple underlying (physical or virtual) interfaces with heterogeneous MTUs
  - OMNI Adaptation Layer (OAL) – minimal mid-layer encapsulation that maps IP layer to underlying interfaces
  - https://datatracker.ietf.org/doc/draft-templin-6man-omni/
  - **Work ready for IETF adoption**

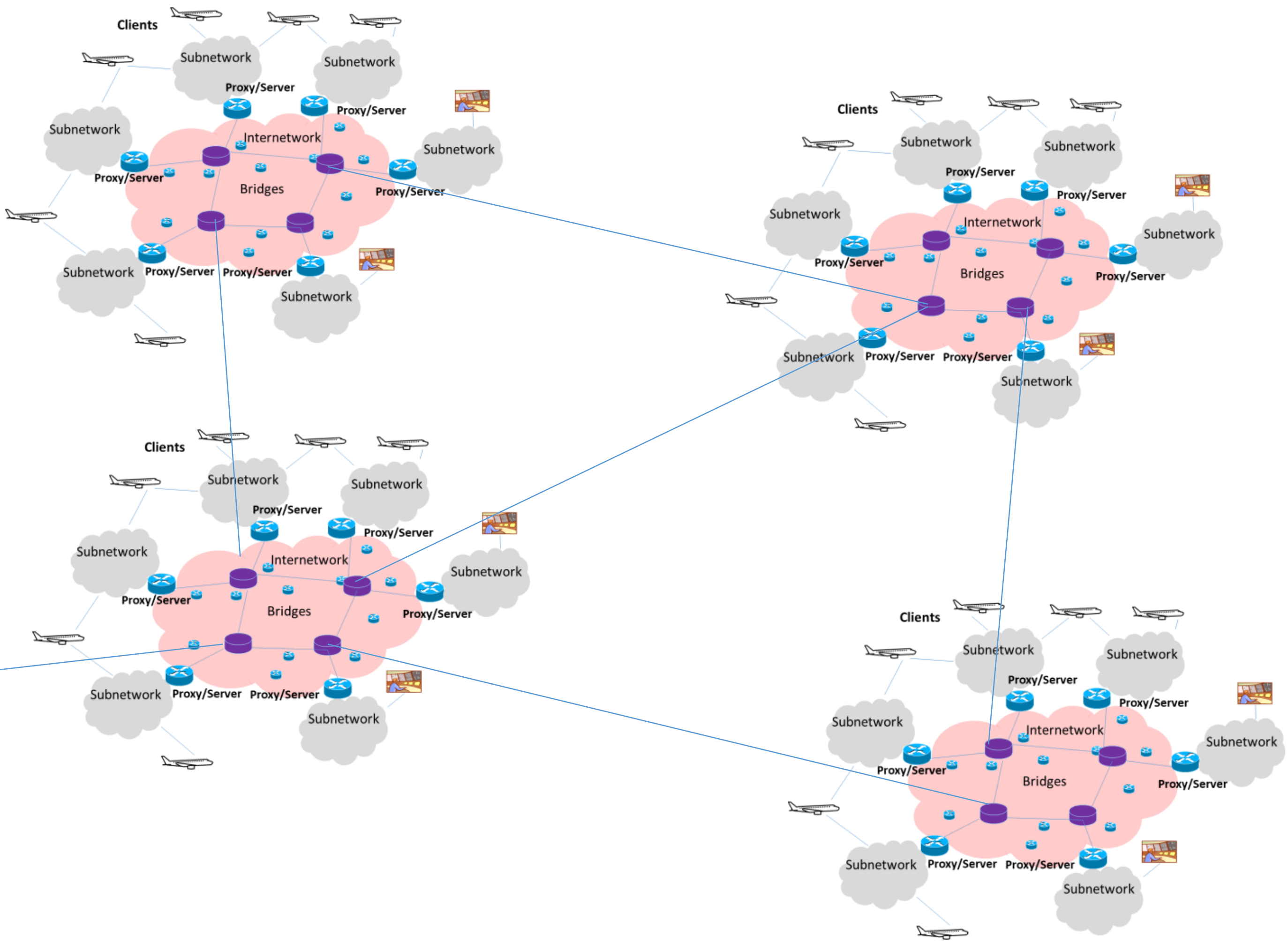
AERO/OMNI in Single Internetwork

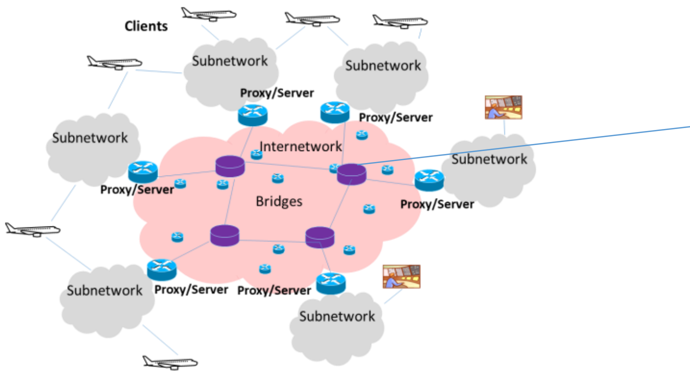


- **Clients** connect to access Subnetworks via one or more data links
- **Proxy/Servers** at Subnetwork edges act as IPv6 routers or IPv6 ND Proxies
- **Bridges** establish "spanning tree" over one or more Internetwork w/**IPv6 ULAs**
- **OMNI Adaptation Layer (OAL)** IPv6 encapsulation sees spanning tree as L2 service (same as bridged campus LAN)

AERO/OMNI Operational Big-Picture

- Arbitrarily-many **Internetwork segments** joined as **Non-Broadcast, Multiple Access (NBMA) link**
- **Segment Routing** over concatenated Internetworks using **OAL forwarding**
- Example Internetwork segments include IPv4 Internet, IPv6 Internet, corporate enterprise networks, cellular operator networks, etc.
- **NOTE: Clients may connect to multiple distinct segments (not shown)**





Overlay Multilink Network Interface (OMNI)

```
+------------------------------+
|     Upper Layer Protocol     |
|                              |
+------------------------------+
|           IP (L3)            |
|                              |
+------------------------------+
|        OMNI Interface        |
|   (OMNI Adaptation Layer)    |
+------------------------------+
|  L2  |  L2  |        |  L2  |
|(IF#1)|(IF#2)| .....  |(IF#n)|
+------+------+        +------+
|  L1  |  L1  |        |  L1  |
|      |      |        |      |
+------+------+        +------+
```

```
                       +----------------+
                       | AERO Bridge B1 |
                       | Nbr: S1, S2, P1|
                       |(X1->S1; X2->S2)|
                       |     MSP M1     |
                       +-------+--------+
  +--------------+             |            +--------------+
  | AERO P/S S1  |             |            | AERO P/S S2  |
  | Nbr: C1, B1  |             |            | Nbr: C2, B1  |
  | default->B1  |             |            | default->B1  |
  |    X1->C1    |             |            |    X2->C2    |
  +-------+------+             |            +------+-------+
          |       OMNI link    |                   |
  X===+===+====================+===================+===+===X
      |                                                |
+-----+--------+                              +--------+-----+
|AERO Client C1|                              |AERO Client C2|
|    Nbr: S1   |                              |    Nbr: S2   |
| default->S1  |                              | default->S2  |
|    MNP X1    |                              |    MNP X2    |
+------+-------+                              +-----+--------+
       |                                            |
      .-.                                          .-.
   ,-(  _)-.                                    ,-(  _)-.
.-(_  IP   )-.   +-------+      +-------+    .-(_  IP   )-.
(__   EUN     )--|Host H1|      |Host H2|--(__   EUN     )
  `-(______)-'   +-------+      +-------+     `-(______)-'
```

- AERO/OMNI Clients, Proxy/Servers and Bridges configure **OMNI Interfaces (node's connection to the OMNI link)**
- Client OMNI Interfaces configured over **multiple underlying interfaces**; connect **End-User Networks (EUN)**
- OMNI interfaces use **IPv6 encapsulation** to span entire concatenated Internetworking path in a single NBMA L3 hop **(all AERO/OMNI nodes are neighbors on the link)**

AERO/OMNI Client Underlying Interfaces

- Clients connect to first-hop Proxy/Servers via underlying interfaces in one of the following ways:
- **Direct-Connect (aka Point-to-Point)** – Client’s underlying interface connects directly to Proxy/Server at L2 over P2P media without traversing any L3 hops
- **VPN** – Client’s underlying interface uses network-layer security to securely establish a virtual link to Proxy/Server over one or more L3 hops
- **Access Network (ANET)** – Client’s underlying interface connects to a secured access network (e.g., enterprise/operator network) with L2/L1 security provisions on the path to the Proxy/Server, which then connects to an external Internetwork
- **Internetwork (INET)** – Client’s underlying interface connects directly to an external Internetwork, where the Proxy/Server may be one or more L3 hops away. Client may be located behind one or several Network Address Translators (NATs)
- **Clients may have diverse underlying interface types, which may connect to multiple distinct Internetworks (as opposed to all via the same Internetwork). For example, an INET interface connected to the public Internet, an ANET interface connected to a cellular operator network, a Direct interface connected to a dedicated enterprise network Proxy/Server, etc.**

OMNI Encapsulation and Segment Routing

```
+------------------------------------------------------------+
|            Network Layer (Original IP packets)             |
+--+---------------------------------------------------------+
   |         OMNI Interface (virtual sublayer nexus)         |
   +--------------------------+------------------------------+
                              |       OAL Encaps/Decaps      |
                              +------------------------------+
                              |        OAL Frag/Reass        |
                  +-----------+-------------+----------------+
                  | UDP Encaps/Decaps/Compress |
             +----+---+-----------+-------+--+  +---------+
             | IP E/D |                | IP E/D |     | IP E/D |
         +---+------+-+-----+   +--+---+----+     +----+---+--+
         |L2 E/D|   |L2 E/D|    |L2 E/D|                  |L2 E/D|
+-------+-----+---+-----+---+-----+--------------+------+
|                  Underlying Interfaces                     |
+------------------------------------------------------------+
```

- OMNI Interface inserts mid-layer encapsulations between IP layer (L3) and Underlying Interfaces (L2) – **minimal encapsulation with effective header compression**
- Mid-layer **segment routing** used as necessary to traverse spanning tree over disjoint Internetworking "segments"
- **Route optimization** avoids strict spanning tree paths whenever possible

```
. . . . . . . . . . . . . . . . . . . . . . . . .
.                                               .
.             .-(::::::::)                      .
.          .-(::::::::::::)-.   +-+             .
.         (:::: Segment A :::)--|B|---+         .
.          `-(::::::::::::)-'   +-+   |         .
.             `-(::::::)-'            |         .
.                                     |         .
.             .-(::::::::)            |         .
.          .-(::::::::::::)-.   +-+   |         .
.         (:::: Segment B :::)--|B|---+         .
.          `-(::::::::::::)-'   +-+   |         .
.             `-(::::::)-'            |         .
.                                     |         .
.             .-(::::::::)            |         .
.          .-(::::::::::::)-.   +-+   |         .
.         (:::: Segment C :::)--|B|---+         .
.          `-(::::::::::::)-'   +-+   |         .
.             `-(::::::)-'            |         .
.                                     |         .
.                ..(etc)..            x         .
.                                               .
.                                               .
.  <-    Segment Routing Topology (SRT)   ->    .
. . . . . . . . . . . . . . . . . . . . . . . . .
```

AERO/OMNI Addressing

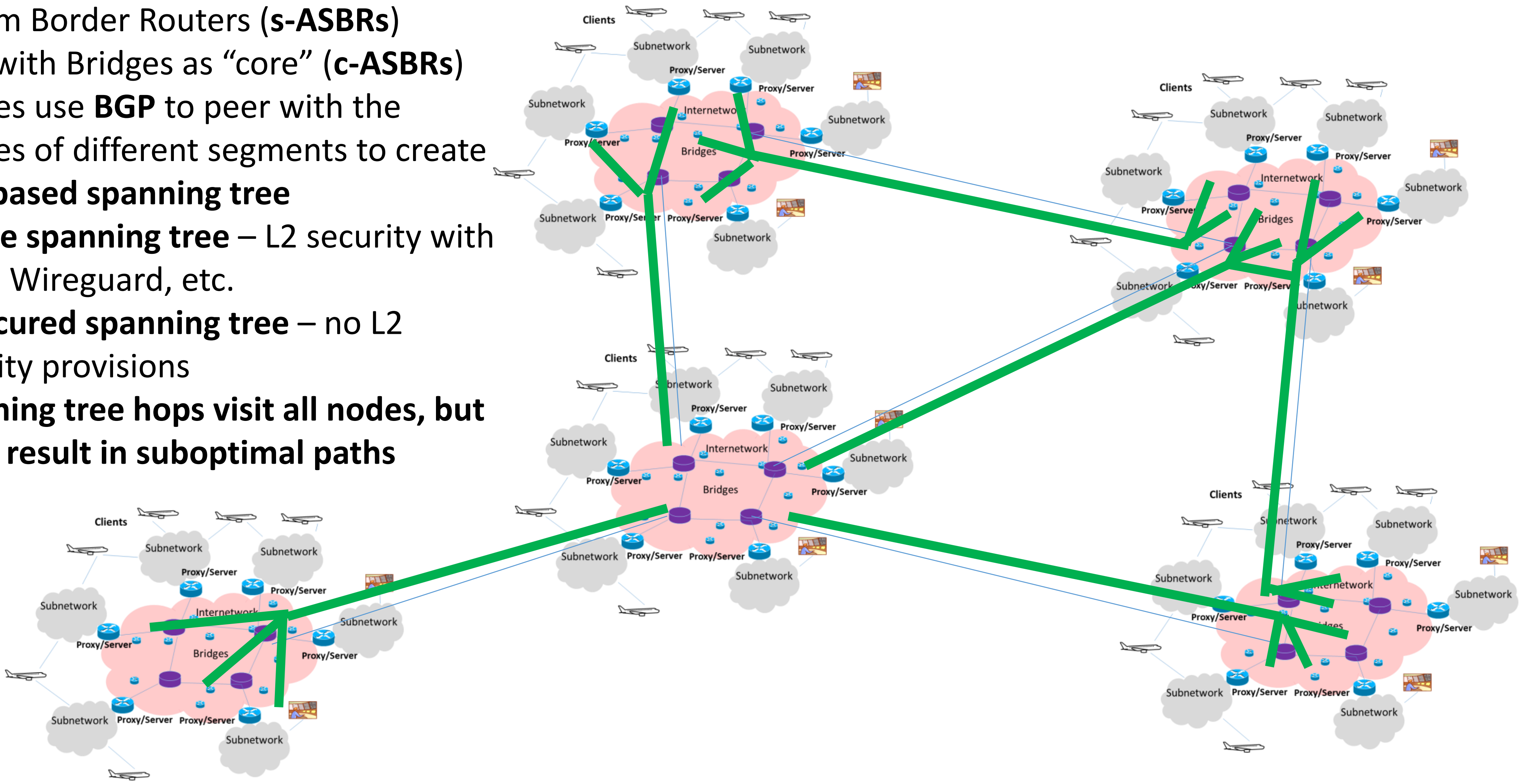
- Each Client advertises **IP Global Unicast Address (GUA)** prefix (aka **Mobile Network Prefix (MNP))** to downstream-attached End-User Networks (EUNs) (e.g., **2001:db8:1:2::/64**)
  - EUN applications use MNP addresses as the source/destination addresses of original IP packets
- Each AERO/OMNI node configures an **IPv6 Link-Local Address (LLA)** for use in IPv6 Neighbor Discovery (IPv6 ND) messaging
  - Clients configure **MNP-LLAs** (e.g., for the delegated MNP 2001:db8:1:2::/64, the MNP-LLA is **fe80::2001:db8:1:2/128**)
  - Proxy/Servers and Bridges configure Administrative LLAs (**ADM-LLAs**) with the 32 least significant bits set according to SRT node numbering/subnetting (e.g., for the SRT subnet 2001:1000/24, ADM-LLAs within that subnet are assigned as **fe80::2001:1001/112**, **fe80::2001:1002/112, fe80::2001:1003/112**, etc.)
- Each AERO/OMNI node configures an **IPv6 Unique-Local Address (ULA)**
  - Clients, Proxy/Servers and Bridges configure ULAs by concatenating the lower 64 bits of their LLAs with the OMNI link 64 bit ULA prefix (e.g., for the ULA prefix fd00:0102:0304:0506::/64 the MNP-ULA taken from the above MNP-LLA is **fd00:0102:0304:0506:2001:db8:1:2/128**)
  - **Clients, Proxy/Servers and Bridges use ULAs as OAL header source/destination addresses**

AERO/OMNI Addressing (continued)

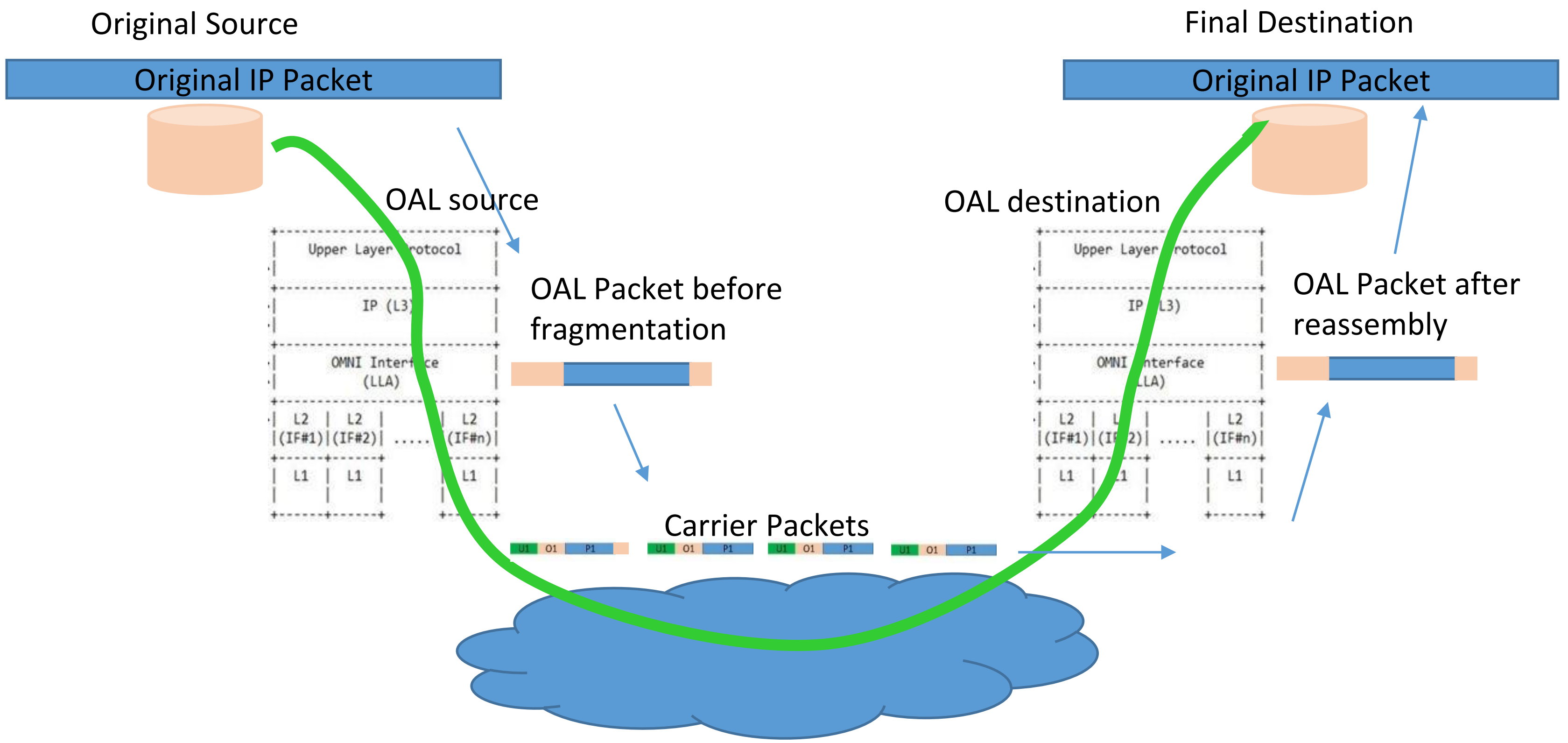
- AERO/OMNI Proxy/Servers configure IPv4 and/or IPv6 **Anycast addresses** to advertise OMNI link reachability into access networks
- Potential Clients that connect to the access network can send network join IPv6 ND messages with an Anycast encapsulation destination address to discover the nearest Proxy/Server for the desired OMNI link
- For IPv4, the Anycast address is **192.88.99.1** (formerly reserved as 6to4 relay anycast, but reclaimed for AERO/OMNI)
- For IPv6, the Anycast address is based on the "6to4-expansion" of 192.88.99.1 as **2002:c058:6301:MNP[64]:Link-Suffix[16]**

ATN-BGP Routing System

- Proxy/Servers are “stub” Autonomous System Border Routers (**s-ASBRs**)
- Peer with Bridges as “core” (**c-ASBRs**)
- Bridges use **BGP** to peer with the Bridges of different segments to create **ULA-based spanning tree**
- **Secure spanning tree** – L2 security with IPsec, Wireguard, etc.
- **Unsecured spanning tree** – no L2 security provisions
- **Spanning tree hops visit all nodes, but often result in suboptimal paths**

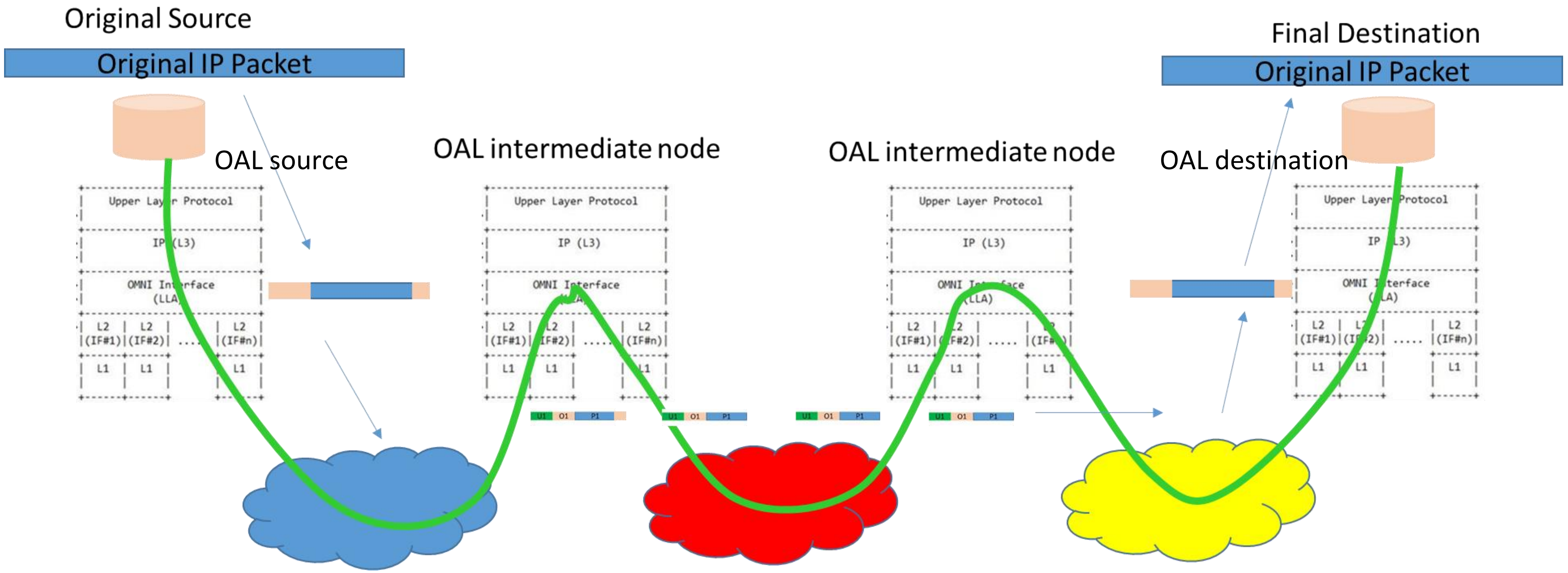


OMNI Interface and OMNI Adaptation Layer (OAL)



- **OAL source** OMNI interface admits original IP packets up to **9180 bytes (OMNI MTU)**
- OAL source encapsulates in IPv6 header with **ULA addresses**, includes trailing checksum, then fragments if necessary and includes *NET header(s) to create "**Carrier Packets**" **guaranteed small enough to traverse all paths**
- Carrier packets traverse one or more Internetworks to **OAL destination** which securely reassembles, verifies checksum, then forwards the original IP packet to final destination
- **Original source can tune packet sizes to achieve optimal performance based on "hard/soft" MTU feedback**

Automatic Extended Route Optimization (AERO)



- Original source and final destination on different Internetwork segments
- **OAL source produces carrier packets, and OAL destination reassembles**
- **OAL intermediate nodes (aka Bridges) join network segments in spanning tree**
- Supports global mobility and route optimization, with dynamic MTU tuning

IPv6 ND Messaging (all ND message types)

- AERO/OMNI nodes use IPv6 ND messages for OMNI link neighbor coordination
- IPv6 ND messages include “**OMNI option**” to:
  1. **Assert protocol conformance**
  2. **Synchronize windows**
  3. **Exchange configuration information**

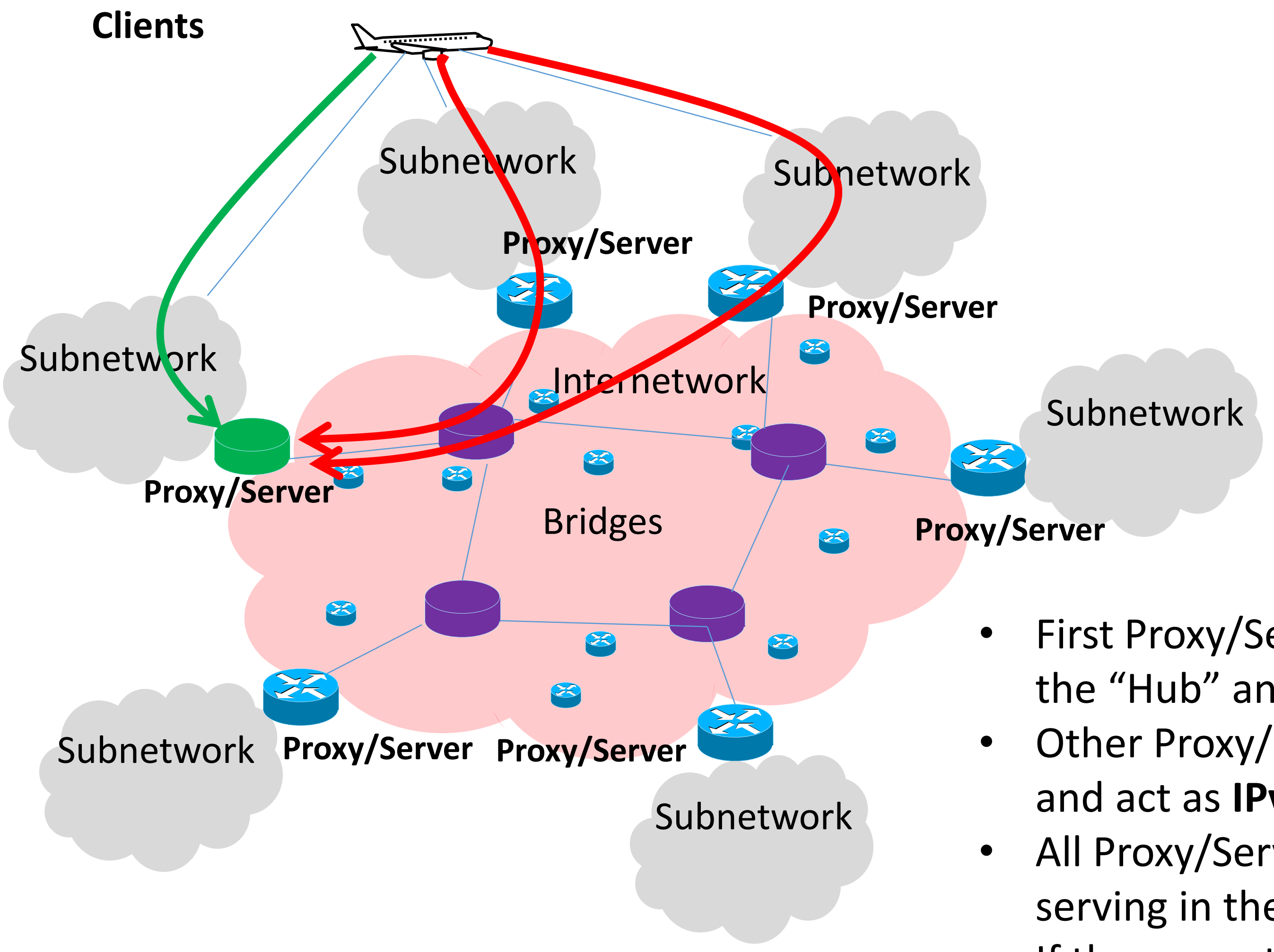
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     |    Preflen    |  S/T-omIndex  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Acknowledgment Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S|A|R|O|P|     |                                               |
|Y|C|S|P|N| Res |                    Window                     |
|N|K|T|T|G|     |                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                          Sub-Options                          ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

OMNI Option

IPv6 ND Messaging (RS/RA)

- Clients and Proxy/Servers exchange Router Solicitation/Advertisement (RS/RA) messages either directly or via a Proxy middlebox:
  - When a Client sends an RS message to Proxy/Server “A” with “All Routers” multicast or the ADM-LLA “A” as the destination, “A” assumes the **ROUTER** role and returns a unicast RA while injecting the Client’s MNP into the AERO/OMNI routing system
  - When a Client sends an RS message to Proxy/Server “B” with the ADM-LLA “A” as the destination, “B” assumes the **PROXY** role and forwards the RS to “A” which returns a unicast RA to the Client via “B”
  - Proxy/Server “A” then becomes the “hub” (aka default router / mobility anchor point) in a hub-and-spokes arrangement with all other Proxy/Servers “B*” as the “spokes”. Client can later switch to a different hub by sending new RS messages

Hub-and-Spokes Proxy/Servers



- First Proxy/Server contacted becomes the "Hub" and acts as **Default Router**
- Other Proxy/Servers become "Spokes" and act as **IPv6 ND Proxys**
- All Proxy/Servers equally capable of serving in the Hub role
- If the current Hub goes down, Client sends new RSes to select a new Hub.

IPv6 ND Messaging (NS/NA)

- Source AERO/OMNI nodes send Neighbor Solicitation (Address Resolution) messages (NS(AR)) into the secured spanning tree to discover the underlying interface configurations of target Clients
- **The secured spanning tree delivers the NS(AR) to the target Client’s hub Proxy/Server, which returns an Neighbor Advertisement (NA(AR))** with multilink link-layer addressing information for the target
- **When the source AERO/OMNI node receives the NA(AR), it can select (source, target) underlying interface pairs and send NS (Neighbor Unreachability Detection) (NS(NUD)) messages to populate forward-path state** in nodes on the path to the target
- **When the target receives the NS(NUD), it returns an NA(NUD) to populate reverse-path stat** and confirm that the bidirectional state information has been established
- Also during the NS/NA(NUD) exchange, both **send/receive Identification “windows” are established** with acceptable carrier packet Identification value ranges

AERO/OMNI Forwarding

- NS/NA(NUD) exchange selects (source, target) underlying interface pair
- Source uses OAL encaps/frag to produce carrier packets and forward them into **First-Hop Segment (FHS)**
- **Segment Routing Topology (SRT) spanning tree** traverses Intermediate segments
- **Last-Hop Segment (LHS)** delivers packets to Target
- **FORWARDING BASED ON:**
  1. **OAL IPv6 ADDRESSES, OR**
  2. **MULTLINK FOWARDING VECTOR INDICES (MFVIs)**

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                *NET Header                     |
|             src = 192.0.2.100                  |
|              dst = 192.0.2.1                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              OAL IPv6 Header                   |
|   src = [ULA*]::2001:db8:1:2                   |
|     dst= [ULA*]::3000:0000                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        CRH-32 (if necessary)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        OAL Fragment Header                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Original IP Header                    |
|        (first-fragment only)                   |
|         src = 2001:db8:1:2::1                  |
|   dst = 2001:db8:1234:5678::1                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                |
~                                                ~
~ Original Packet Body/Fragment                  ~
~                                                ~
|                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

OAL Carrier Packet

```
                        +---------------+        (:::)-.
                        |Source Client  |<-->.-(::EUN:::)
                        +---------------+        `-(::::)-'
                        |OMNI interface|
                        +----+----+----+
            +-----------|IF#1|IF#2|IF#n|------ +
           /            +----+----+----+        \
          /                    |                 \
         /                     |                  \
        v                      v                   v
      (:::)-.                (:::)-.             (:::)-.
 .-(::*NET:::)         .-(::*NET:::)        .-(::*NET:::)
   `-(::::)-'            `-(::::)-'           `-(::::)-'
    +------+              +------+             +------+
... |FHS#1| .........     |FHS#2| .........    |FHS#n| ...
    +--|---+              +--|---+             +--|---+
       |                     |                    |
        \                    v                    /
         \                                       /
          v                (:::)-.              v
                       .-(:::::::)
                   .-(::: Segment :::)-.
                 (:::::   Routing   ::::)
                   `-(:: Topology ::)-'
                       `-(:::::::-'
                  /          |          \
                 /           |           \
                v            v            v
    +------+              +------+             +------+
... |LHS#1| .........     |LHS#2| .........    |LHS#m| ...
    +--|---+              +--|---+             +--|---+
        \                    |                    /
         v                   v                   v
                    <-- Target Clients -->
```

AERO/OMNI Multilink Forwarding Vectors

- OAL header includes IPv6 ULA (src,dst) for “traditional” IPv6-based forwarding of carrier packets over strict spanning tree paths, but **this may result in longer paths and more encapsulation overhead than are necessary**
- AERO/OMNI introduce a new forwarding table known as the **Multilink Forwarding Information Base (MFIB) that contains Multilink Forwarding Vectors (MFVs) identified by unique 32-bit Multilink Forwarding Vector Indexes (MFVIs)**
- When an OAL source sends an NS(NUD) toward a target, it includes an OMNI option with **Multilink Forwarding Parameters** to establish MFVs and **reverse-path “B” MFVIs** in AERO/OMNI nodes along the path
- When the target returns an NA(NUD) toward the OAL source, it follows the reverse MFV path created by the NS(NUD) and establishes **forward-path “A” MFVIs**
- With MFVs/MFVIs populated, large portions of the OAL header and fragment header can be omitted and an **OAL Compressed Header (OCH)** can be included instead. Additionally, any unnecessary hops can be “skipped” to avoid strict spanning tree traversal as an effective **Route Optimization**

Multilink Forwarding Parameters

- OMNI option sub-option (Type=3)
- Includes **First Hop Segment (FHS)** and **Last Hop Segment (LHS)** Parameters with associated **Forward/Mode/Type (FMT)** and **Segment Routing Topology (SRT)** codes
- FHS/LHS parameters include:
  - Client UDP Port/INADDR
  - Proxy/Server MSID/INADDR
  - Bridge MSID/INADDR
- MFVI List - records A/B MVFIs established at each FHS/LHS hop
- A/B indexes, plus 2-bit “Job” code

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| S-Type=3|     Sub-length=N      |FHS Cli omIndex|     omType    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Provider ID  | Link  | Resvd | FMT |    SRT    |               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                 ~
~                  FHS Client UDP Port/INADDR                   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                 FHS Proxy/Server MSID/INADDR                  ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    FHS Bridge MSID/INADDR                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|LHS Cli omIndex|     omType    |  Provider ID  | Link  | Resvd |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| FMT |    SRT    |                                             ~
+-+-+-+-+-+-+-+-+-+                                             ~
~                  LHS Client UDP Port/INADDR                   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                 LHS Proxy/Server MSID/INADDR                  ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    LHS Bridge MSID/INADDR                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~            Tunnel Window Synchronization Parameters           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
~        Multilink Forwarding Vector Index (MFVI) List          ~
~              (5 consecutive 4-octet MFVIs)                    ~
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  A  |  B  |Job|
+-+-+-+-+-+-+-+-+
```

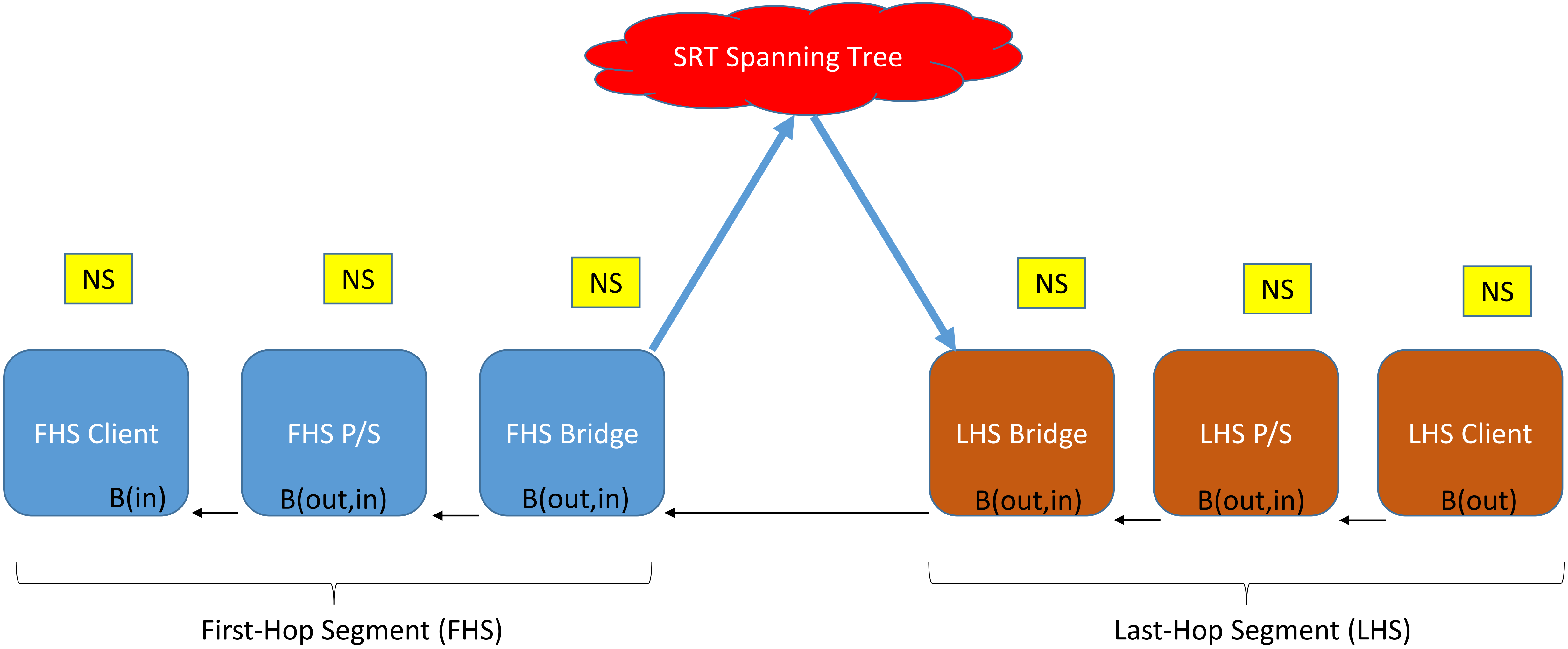
AERO/OMNI Multilink Forwarding Vectors (2)

- Each AERO/OMNI node creates MFV that records OAL (src,dst) ULAs, and two 32-bit MFVIs:
  - **one local "A" (forward-path) and one local "B" (reverse-path) MFVI**
  - **any next-hop "A" MFVIs**
  - **any previous-hop "B" MFVIs**
- The local A/B MFVIs MUST be assured node-local unique (e.g. through 32-bit random number generation while testing for local uniqueness), but need not be unique among other AERO/OMNI nodes
  - **MFVIs ARE NOT IPv{4,6} ADDRESSES, MPLS LABELS, ASNs, SPIs, etc.**
  - **MFVIs ARE SIMPLY NODE-LOCAL UNIQUE 32-BIT INTEGERS**
- When an AERO/OMNI node processes an NS(NUD), it creates an MVF and generates a local "B" MFVI while recording its information in the NS(NUD) Multilink Forwarding Parameters, then forwards the NS(NUD) to the next hop toward the target
- When the next hop AERO/OMNI node receives the NS(NUD), it creates an MFV, caches the previous-hop "B" MFVI, generates a local "B" MFVI and forwards as above
- When the final hop AERO/OMNI node receives the NS(NUD), it discovers the full map of previous-hop nodes along with the "B" values for all previous-hops

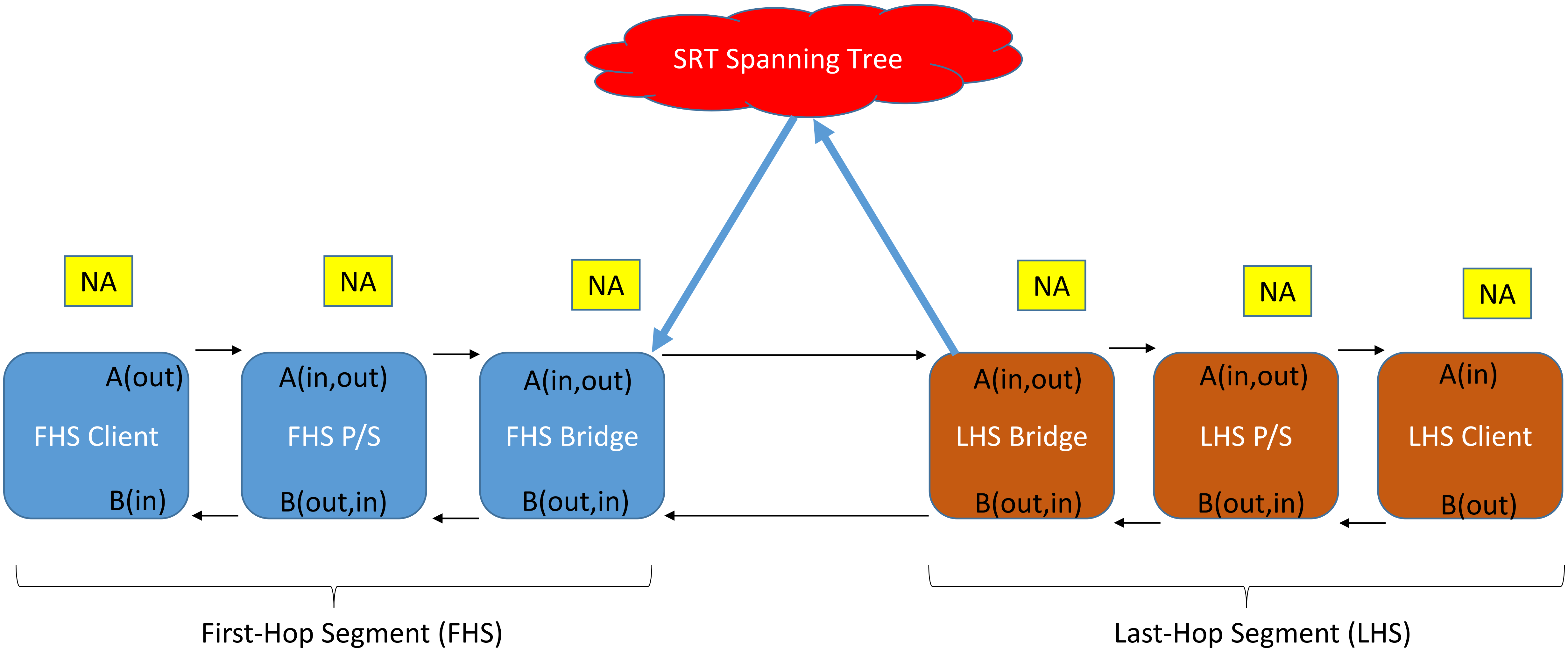
AERO/OMNI Multilink Forwarding Vectors (3)

- When the final-hop AERO/OMNI node receives the NS(NUD), it creates an MFV and assigns a unique local “A” MFVI. The node then generates an NA(NUD) and returns it toward the original source by following the chain of “B” MFVI previous-hops established by the NS(NUD)
- When a previous hop AERO/OMNI node receives the NA(NUD), it locates the MFV based on the cached “B” MFVI, caches the next hop “A” MFVI, and assigns a unique local “A” MFVI
- When the first-hop AERO/OMNI node receives the NA(NUD), it discovers the full map of next-hop nodes along with the “A” values for all hops on the path to the target

NS(NUD) – (Initialize MFV; Build "B" MFVI)



NA(NUD) – (Follow "B" MFVI; Build "A" MFVI)



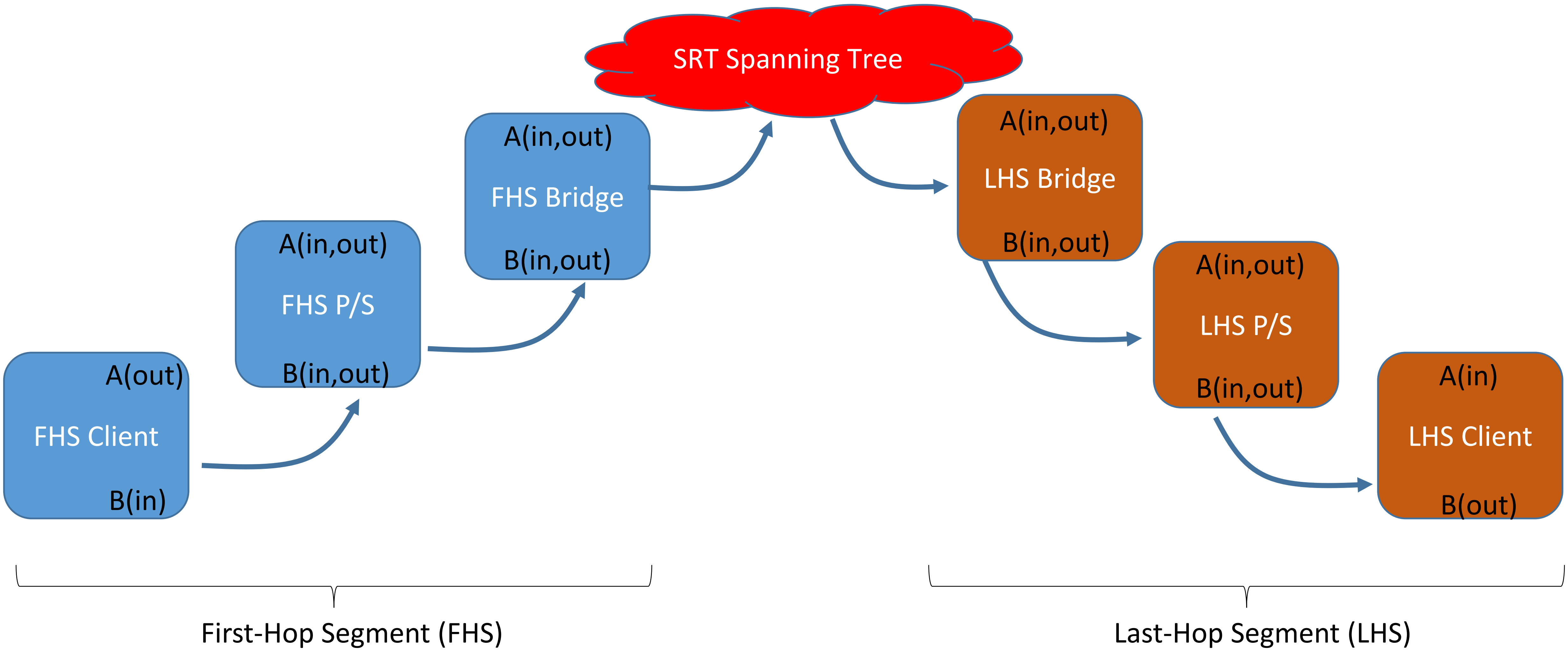
AERO/OMNI MFV-based Forwarding

- After MVFs and A/B MFVIs have been established, OAL peers can begin sending carrier packets while omitting significant portions of the OAL header, fragment header and CRH-32 routing header using the **OMNI Compressed Header, Type 0 and Type 1 (OCH-0/1)**
- OCH-0/1 includes the A or B MFVI for the next-hop AERO/OMNI node according to "directionality" established in the NS/NA(NUD) exchange (i.e., the NS source includes the "A" MFVI, and the NA responder includes the "B" MFVI)
- **When the AERO/OMNI node receives a carrier packet with an OCH-0/1, it uses the MFVI to locate the MFV which includes next/previous hop information**
- **The AERO/OMNI node then rewrites the MFVI value in the OCH-0/1 compressed header and forwards to the next/previous hop**
- This MFV-based forwarding is based on a single 32-bit MFVI and can be used to avoid strict spanning tree paths

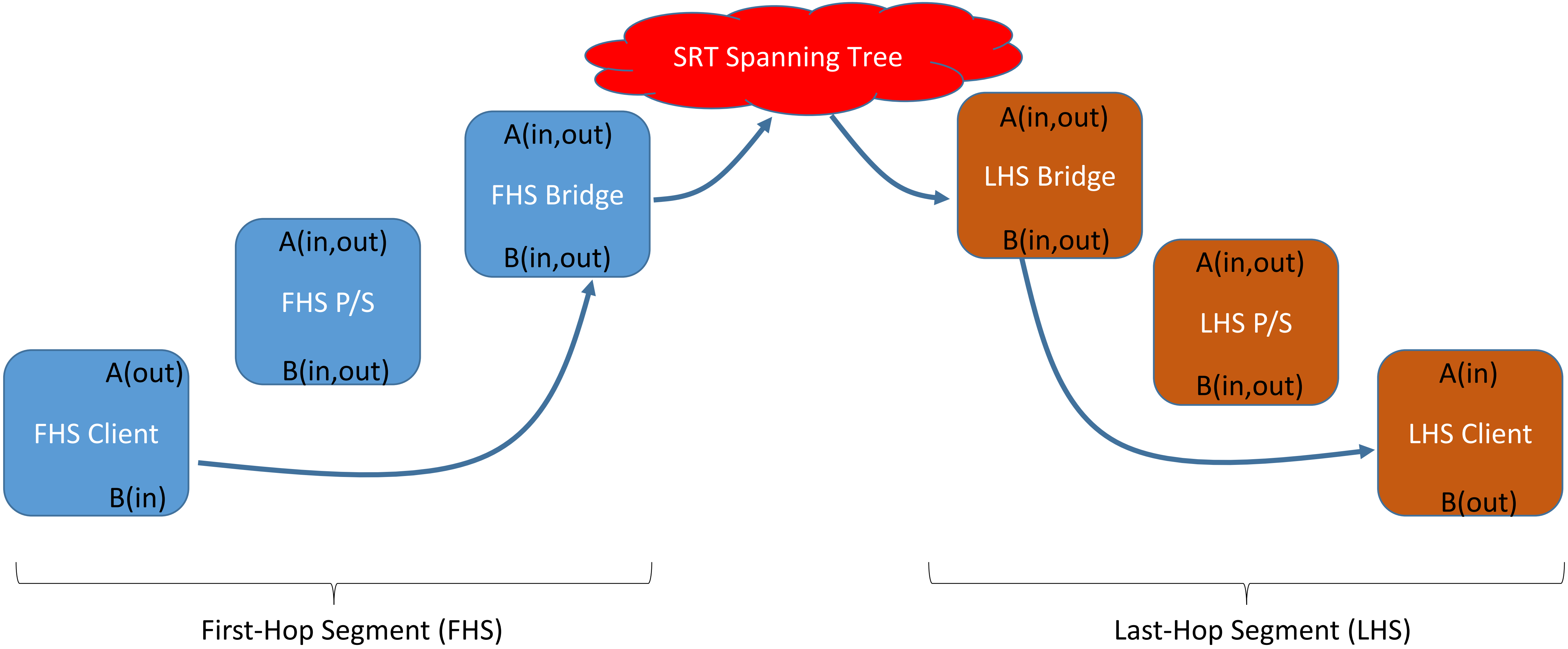
AERO/OMNI MFV Maintenance

- After MVFs and A/B MFVIs have been established, OAL peers exchange additional NS/NA(NUD) messages before ReachableTime expires to keep the soft state alive
- If ReachableTime expires, AERO/OMNI nodes garbage-collect the MFVs, and future NS/NA(NUD) exchanges needed to re-establish new MVFs
- The exact MFVIs from expired MFVs need not be re-used when initializing new MFVs – all that is required is that MFVIs are unique within node-local scope until their ReachableTime expires
- Since there are 2^{32} possible MVFIs and since the number of MVFIs in use at a given time will be much smaller, random-generation should produce a unique new MFVI after a very small number of attempts (usually only 1)

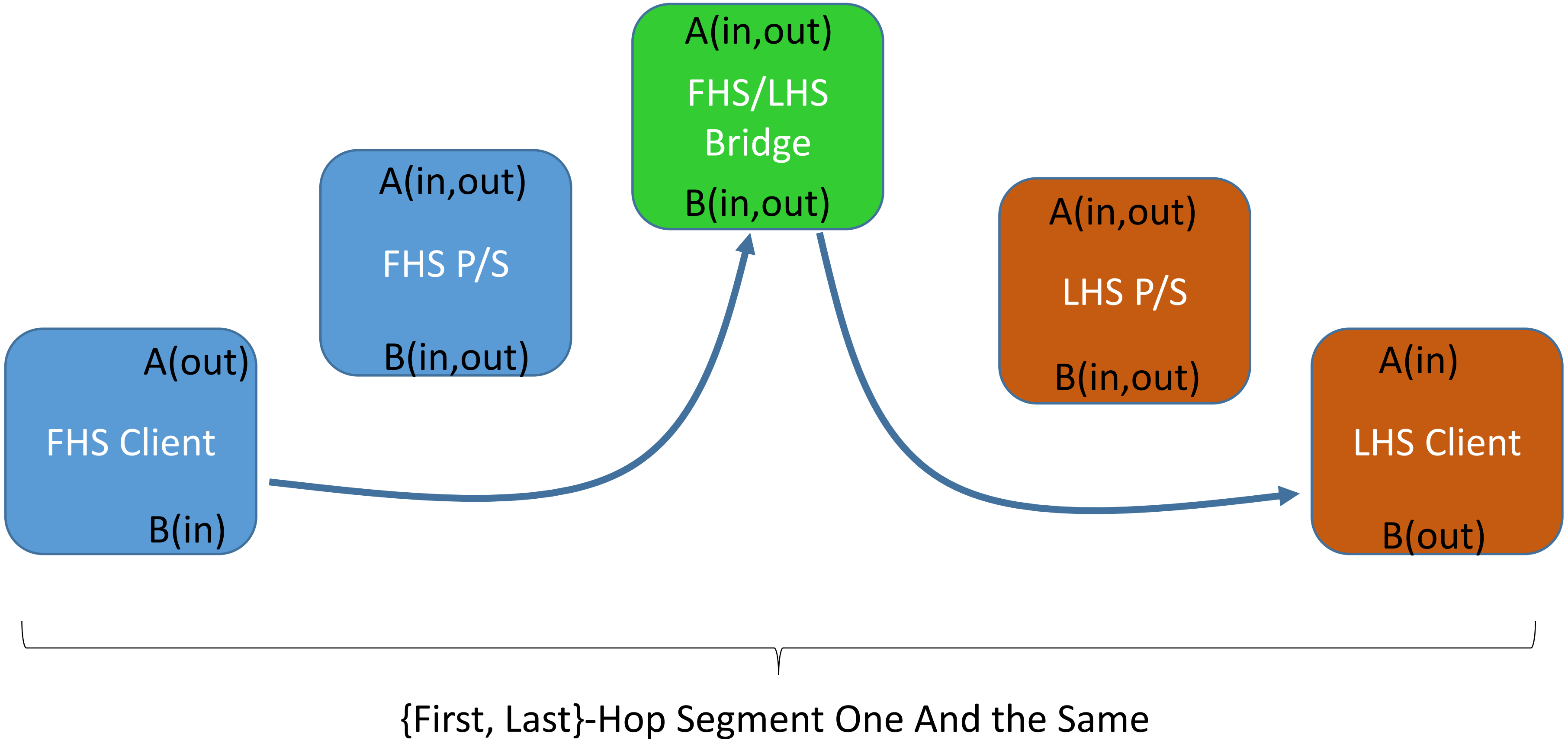
FHS/LHS Peer Exchanges Before Route Optimization



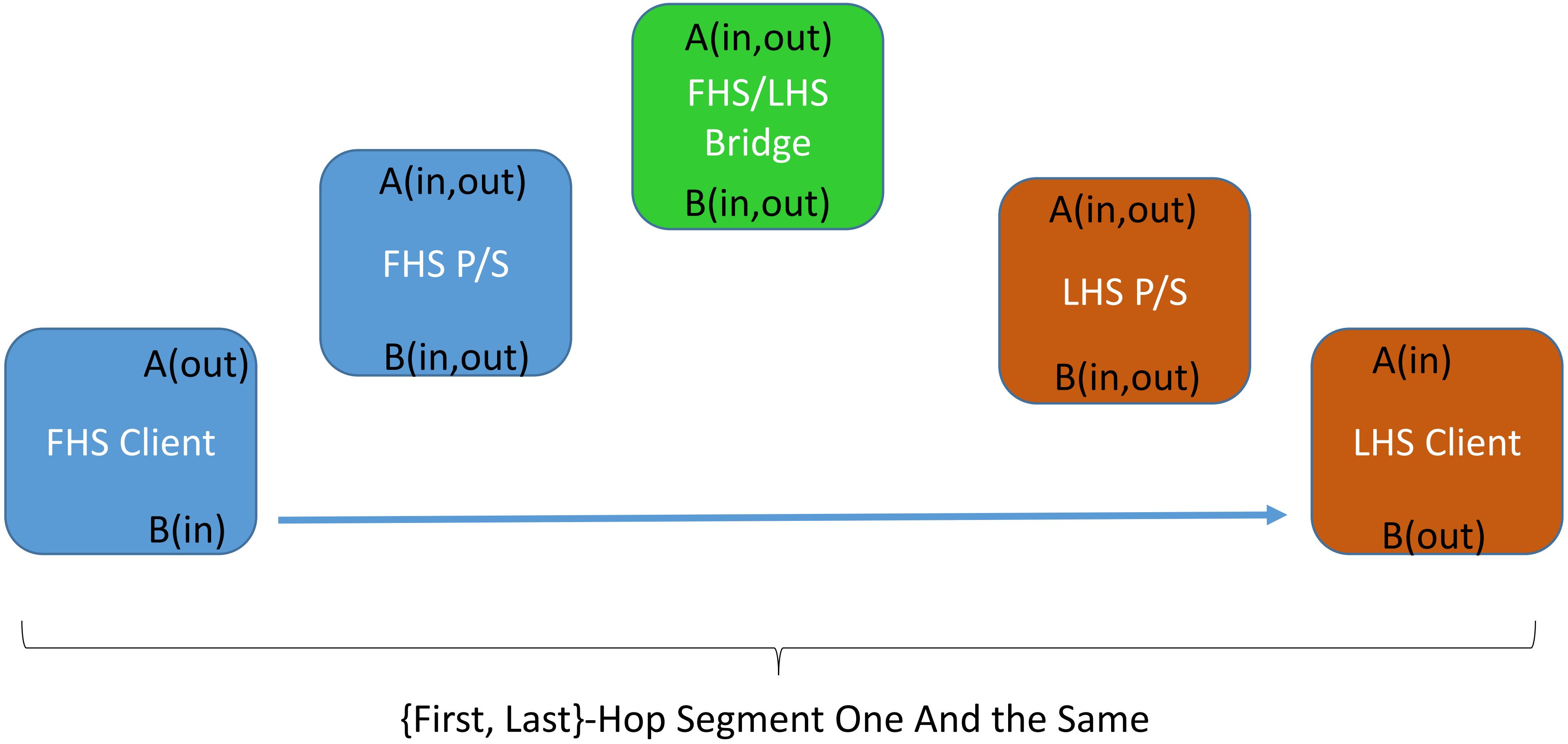
FHS/LHS Peer Exchanges After Route Optimization



Peer Exchanges After Route Optimization When FHS/LHS One and the Same (Before Client/Client)



Peer Exchanges After Route Optimization When FHS/LHS One and the Same (After Client/Client)



NAT Traversal Issues

- When source/target Client underlying interface connects via an open Internetwork, there is a very good chance it may be located behind one or more IP Network Address Translators (NATs)
- Each Client naturally establishes NAT mappings when it performs an RS/RA exchange with its FHS Proxy/Server, but these mappings generally not viable for Client exchanges with other AERO/OMNI nodes
- When AERO/OMNI route optimization is applied, route optimization peer nodes must perform NS/NA(NUD) and “bubble” exchanges to establish NAT mappings for itself in the NAT(s) on the path to the Client
- NAT traversal procedures are the same as in RFC4380 and RFC6081

IPv6 ND Messaging (unsolicited NAs)

- Hub Proxy/Servers announce Client state changes (e.g., mobility-related address changes, link quality changes, traffic selector changes) by sending unsolicited NAs (uNAs) to all nodes that it has recently sent an NA(AR)
  - When nodes receive uNAs, they update their state information for the Client which may require new NS/NA(NUD) path state exchanges
- New Hub Proxy/Servers send uNAs to inform old P/S that Client has "Departed"
  - When the old Hub Proxy/Server receives uNAs, it records the new Hub Proxy/Server ADM-LLA, withdraws MNP route and forwards uNAs to nodes that it has recently sent an NA(AR)
- Target nodes request selective link-layer retransmissions by sending uNAs to source with the Identifications (and ordinal numbers) of any missing fragments
  - When the source node receives the uNAs, it retransmits requested fragments if it still has them in its retransmission cache
  - Retransmission window should be brief, and determines link persistence

Document Adoption Call

- Three inter-dependent, but separable documents (first already IETF rtgwg working group item; calling for adoption of other two)
- “A Simple BGP-based Mobile Routing System for the Aeronautical Telecommunications Network”
  - https://datatracker.ietf.org/doc/draft-ietf-rtgwg-atn-bgp/
  - **Work ready for IETF rtgwg WGLC**
- “Automatic Extended Route Optimization (AERO)”
  - https://datatracker.ietf.org/doc/draft-templin-6man-aero/
  - **Adopt as IETF rtgwg working group item?**
- “Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces”
  - https://datatracker.ietf.org/doc/draft-templin-6man-omni/
  - **Adopt as IETF rtgwg working group item?**