Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 3, 2022

F. Templin, Ed.
Boeing Research & Technology

July 2, 2021

Automatic Extended Route Optimization (AERO)
draft-templin-6man-aero-22

Abstract

This document specifies an Automatic Extended Route Optimization (AERO) service for IP internetworking over Overlay Multilink Network (OMNI) interfaces. AERO/OMNI use an IPv6 link-local address format that supports operation of the IPv6 Neighbor Discovery (IPv6 ND) protocol. Prefix delegation/registration services are employed for network admission and to manage the IP forwarding and routing systems. Secure multilink operation, mobility management, multicast, traffic selector signaling and route optimization are naturally supported through dynamic neighbor cache updates. AERO is a widely-applicable mobile internetworking service especially well-suited to aviation services, intelligent transportation systems, mobile end user devices and many other applications.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Automatic Extended Route Optimization (AERO) fulfills the requirements of Distributed Mobility Management (DMM) [RFC7333] and route optimization [RFC5522] for aeronautical networking and other network mobility use cases including intelligent transportation

systems and enterprise mobile device users. AERO is a secure internetworking and mobility management service that employs the Overlay Multilink Network Interface (OMNI) [I-D.templin-6man-omni] Non-Broadcast, Multiple Access (NBMA) virtual link model. The OMNI link is a virtual overlay configured over one or more underlying Internetworks, and nodes on the link can exchange original IP packets as single-hop neighbors. The OMNI Adaptation Layer (OAL) supports multilink operation for increased reliability and path optimization while providing fragmentation and reassembly services to support Maximum Transmission Unit (MTU) diversity. In terms of precedence, this specification may provide first-principle insights into a representative mobility service architecture as context for understanding the OMNI specification.

The AERO service comprises Clients, Proxy/Servers and Relays that are seen as OMNI link neighbors as well as Bridges that interconnect diverse Internetworks as OMNI link segments through OAL forwarding at a layer below IP. Each node's OMNI interface uses an IPv6 link-local address format that supports operation of the IPv6 Neighbor Discovery (IPv6 ND) protocol [RFC4861]. A node's OMNI interface can be configured over multiple underlying interfaces, and therefore appears as a single interface with multiple link-layer addresses. Each link-layer address is subject to change due to mobility and/or multilink fluctuations, and link-layer address changes are signaled by ND messaging the same as for any IPv6 link.

AERO provides a secure cloud-based service where mobile node Clients may use Proxy/Servers acting as default routers and mobility anchor points while fixed nodes may use any Relay on the link for efficient communications. Fixed nodes forward original IP packets destined to other AERO nodes via the nearest Relay, which forwards them through the cloud. Mobile node Clients discover shortest paths to OMNI link neighbors through AERO route optimization. Both unicast and multicast communications are supported, and Clients may efficiently move between locations while maintaining continuous communications with correspondents and without changing their IP Address.

AERO Bridges peer with Proxy/Servers in a secured private BGP overlay routing instance to establish a Segment Routing Topology (SRT) spanning tree over the underlying Internetworks of multiple disjoint administrative domains as a single unified OMNI link. Each OMNI link instance is characterized by the set of Mobility Service Prefixes (MSPs) common to all mobile nodes. Relays provide an optimal route from (fixed) correspondent nodes on the underlying Internetwork to (mobile or fixed) nodes on the OMNI link. To the underlying Internetwork, the Relay is the source of a route to the MSP; hence uplink traffic to the mobile node is naturally routed to the nearest Relay.

AERO can be used with OMNI links that span private-use Internetworks and/or public Internetworks such as the global Internet. In the latter case, some end systems may be located behind global Internet Network Address Translators (NATs). A means for robust traversal of NATs while avoiding "triangle routing" and critical infrastructure traffic concentration is therefore provided.

AERO assumes the use of PIM Sparse Mode in support of multicast communication. In support of Source Specific Multicast (SSM) when a Mobile Node is the source, AERO route optimization ensures that a shortest-path multicast tree is established with provisions for mobility and multilink operation. In all other multicast scenarios there are no AERO dependencies.

AERO provides a secure aeronautical internetworking service for both manned and unmanned aircraft, where the aircraft is treated as a mobile node that can connect an Internet of Things (IoT). AERO is also applicable to a wide variety of other use cases. For example, it can be used to coordinate the links of mobile nodes (e.g., cellphones, tablets, laptop computers, etc.) that connect into a home enterprise network via public access networks with VPN or non-VPN services enabled according to the appropriate security model. AERO can also be used to facilitate terrestrial vehicular and urban air mobility (as well as pedestrian communication services) for future intelligent transportation systems [I-D.ietf-ipwave-vehicular-networking][I-D.templin-ipwave-uam-its]. Other applicable use cases are also in scope.

Along with OMNI, AERO provides secured optimal routing support for the "6M's" of modern Internetworking, including:

1. Multilink - a mobile node's ability to coordinate multiple diverse underlying data links as a single logical unit (i.e., the OMNI interface) to achieve the required communications performance and reliability objectives.

2. Multinet - the ability to span the OMNI link over a segment routing topology with multiple diverse network administrative domains while maintaining seamless end-to-end communications between mobile Clients and correspondents such as air traffic controllers, fleet administrators, etc.

3. Mobility - a mobile node's ability to change network points of attachment (e.g., moving between wireless base stations) which may result in an underlying interface address change, but without disruptions to ongoing communication sessions with peers over the OMNI link.

4. Multicast - the ability to send a single network transmission that reaches multiple nodes belonging to the same interest group, but without disturbing other nodes not subscribed to the interest group.

5. Multihop - a mobile node vehicle-to-vehicle relaying capability useful when multiple forwarding hops between vehicles may be necessary to "reach back" to an infrastructure access point connection to the OMNI link.

6. MTU assurance - the ability to deliver packets of various robust sizes between peers without loss due to a link size restriction, and to dynamically adjust packets sizes to achieve the optimal performance for each independent traffic flow.

The following numbered sections present the AERO specification. The appendices at the end of the document are non-normative.

2. Terminology

The terminology in the normative references applies; especially, the terminology in the OMNI specification [I-D.templin-6man-omni] is used extensively throughout. The following terms are defined within the scope of this document:

IPv6 Neighbor Discovery (IPv6 ND)
a control message service for coordinating neighbor relationships between nodes connected to a common link. AERO uses the IPv6 ND messaging service specified in [RFC4861].

IPv6 Prefix Delegation
a networking service for delegating IPv6 prefixes to nodes on the link. The nominal service is DHCPv6 [RFC8415], however alternate services (e.g., based on IPv6 ND messaging) are also in scope. A minimal form of prefix delegation known as "prefix registration" can be used if the Client knows its prefix in advance and can represent it in the source address of an IPv6 ND message.

Access Network (ANET)
a node's first-hop data link service network (e.g., a radio access network, cellular service provider network, corporate enterprise network, etc.) that often provides link-layer security services such as IEEE 802.1X and physical-layer security (e.g., "protected spectrum") to prevent unauthorized access internally and with border network-layer security services such as firewalls and proxys that prevent unauthorized outside access.

ANET interface

a node's attachment to a link in an ANET.

Internetwork (INET)

a network topology with a coherent IP routing and addressing plan and that provides a transit backbone service for its connected end systems. INETs also provide an underlay service over which the AERO virtual link is configured. Example INETs include corporate enterprise networks, aviation networks, and the public Internet itself. When there is no administrative boundary between an ANET and the INET, the ANET and INET are one and the same.

INET interface

a node's attachment to a link in an INET.

*NET

a "wildcard" term referring to either ANET or INET when it is not necessary to draw a distinction between the two.

*NET interface

a node's attachment to a link in a *NET.

*NET Partition

frequently, *NETs such as large corporate enterprise networks are sub-divided internally into separate isolated partitions (a technique also known as "network segmentation"). Each partition is fully connected internally but disconnected from other partitions, and there is no requirement that separate partitions maintain consistent Internet Protocol and/or addressing plans. (Each *NET partition is seen as a separate OMNI link segment as discussed below.)

*NET address

an IP address assigned to a node's interface connection to a *NET.

*NET encapsulation

the encapsulation of a packet in an outer header or headers that can be routed within the scope of the local *NET partition.

OMNI link

the same as defined in [I-D.templin-6man-omni]. The OMNI link employs IPv6 encapsulation [RFC2473] to traverse intermediate nodes in a spanning tree over underlying *NET segments the same as a bridged campus LAN. AERO nodes on the OMNI link appear as single-hop neighbors at the network layer even though they may be separated by many underlying *NET hops; AERO nodes can employ Segment Routing [RFC8402] to navigate between different OMNI links, and/or to cause packets to visit selected waypoints within the same OMNI link.

OMNI Interface

a node's attachment to an OMNI link. Since OMNI interface addresses are managed for uniqueness, OMNI interfaces do not require Duplicate Address Detection (DAD) and therefore set the administrative variable 'DupAddrDetectTransmits' to zero [RFC4862].

OMNI Adaptation Layer (OAL)

an OMNI interface service that subjects original IP packets admitted into the interface to mid-layer IPv6 header encapsulation followed by fragmentation and reassembly. The OAL is also responsible for generating MTU-related control messages as necessary, and for providing addressing context for spanning multiple segments of a bridged OMNI link.

original IP packet

a whole IP packet or fragment admitted into the OMNI interface by the network layer prior to OAL encapsulation and fragmentation, or an IP packet delivered to the network layer by the OMNI interface following OAL decapsulation and reassembly.

OAL packet

an original IP packet encapsulated in OAL headers and trailers before OAL fragmentation, or following OAL reassembly.

OAL fragment

a portion of an OAL packet following fragmentation but prior to *NET encapsulation, or following *NET encapsulation but prior to OAL reassembly.

(OAL) atomic fragment

an OAL packet that can be forwarded without fragmentation, but still includes a Fragment Header with a valid Identification value and with Fragment Offset and More Fragments both set to 0.

(OAL) carrier packet

an encapsulated OAL fragment following *NET encapsulation or prior to *NET decapsulation. OAL sources and destinations exchange carrier packets over underlying interfaces, and may be separated by one or more OAL intermediate nodes. OAL intermediate nodes re-encapsulate carrier packets during forwarding by removing the *NET headers of the previous hop underlying network and replacing them with new *NET headers for the next hop underlying network.

OAL source

an OMNI interface acts as an OAL source when it encapsulates original IP packets to form OAL packets, then performs OAL fragmentation and *NET encapsulation to create carrier packets.

OAL destination
an OMNI interface acts as an OAL destination when it decapsulates carrier packets, then performs OAL reassembly and decapsulation to derive the original IP packet.

OAL intermediate node
an OMNI interface acts as an OAL intermediate node when it removes the *NET headers of carrier packets received from a first hop, then re-encapsulates the carrier packets in new *NET headers and forwards them to the next hop. OAL intermediate nodes decrement the Hop Limit of the OAL IPv6 header during re-encapsulation, and discard the packet if the Hop Limit reaches 0. OAL intermediate nodes do not decrement the Hop Limit/TTL of the original IP packet.

underlying interface
a *NET interface over which an OMNI interface is configured.

Mobility Service Prefix (MSP)
an aggregated IP Global Unicast Address (GUA) prefix (e.g., 2001:db8::/32, 192.0.2.0/24, etc.) assigned to the OMNI link and from which more-specific Mobile Network Prefixes (MNPs) are delegated. OMNI link administrators typically obtain MSPs from an Internet address registry, however private-use prefixes can alternatively be used subject to certain limitations (see: [I-D.templin-6man-omni]). OMNI links that connect to the global Internet advertise their MSPs to their interdomain routing peers.

Mobile Network Prefix (MNP)
a longer IP prefix delegated from an MSP (e.g., 2001:db8:1000:2000::/56, 192.0.2.8/30, etc.) and delegated to an AERO Client or Relay.

Mobile Network Prefix Link Local Address (MNP-LLA)
an IPv6 Link Local Address that embeds the most significant 64 bits of an MNP in the lower 64 bits of fe80::/64, as specified in [I-D.templin-6man-omni].

Mobile Network Prefix Unique Local Address (MNP-ULA)
an IPv6 Unique-Local Address derived from an MNP-LLA.

Administrative Link Local Address (ADM-LLA)
an IPv6 Link Local Address that embeds a 32-bit administratively-assigned identification value in the lower 32 bits of fe80::/96, as specified in [I-D.templin-6man-omni].

Administrative Unique Local Address (ADM-ULA)
an IPv6 Unique-Local Address derived from an ADM-LLA.

AERO node

a node that is connected to an OMNI link and participates in the AERO internetworking and mobility service.

AERO Client ("Client")

an AERO node that connects over one or more underlying interfaces and requests MNP delegation/registration service from AERO Proxy/Servers. The Client assigns an MNP-LLA to the OMNI interface for use in IPv6 ND exchanges with other AERO nodes and forwards original IP packets to correspondents according to OMNI interface neighbor cache state.

AERO Proxy/Server ("Proxy/Server")

a node that provides a proxying service between AERO Clients and external peers on its Client-facing ANET interfaces (i.e., in the same fashion as for an enterprise network proxy) as well as default forwarding and mobility anchor point services for coordination with correspondents on its INET-facing interfaces. (Proxy/Servers in the open INET instead configure only an INET interface and no ANET interfaces.) The Proxy/Server configures an OMNI interface and assigns an ADM-LLA to support the operation of IPv6 ND services, while advertising all of its associated MNPs via BGP peerings with Bridges.

AERO Relay ("Relay")

a Proxy/Server that provides forwarding services between nodes reached via the OMNI link and correspondents on other links/networks. AERO Relays configure an OMNI interface and assign an ADM-LLA the same as Proxy/Servers, and also run a dynamic routing protocol to discover any non-MNP IP GUA routes in service on its other links/networks. The Relay advertises the MSP(s) to its other links/networks, and redistributes routes discovered on other links/networks into the OMNI link routing system the same as for Proxy/Servers.

AERO Bridge ("Bridge")

a BGP hub autonomous system node that also provides OAL forwarding services for nodes on an OMNI link. Bridges forwards carrier packets between OMNI link segments as OAL intermediate nodes while decrementing the OAL IPv6 header Hop Limit but without decrementing the network layer IP TTL/Hop Limit. Bridges peer with Proxy/Servers and other Bridges to form a spanning tree over all OMNI link segments and to discover the set of all MNP and non-MNP prefixes in service. Bridges process carrier packets received over the secured spanning tree that are addressed to themselves, while forwarding all other carrier packets to the next hop also via the secured spanning tree. Bridges forward carrier packets received over the unsecured spanning tree to the next hop either

via the unsecured spanning tree or via direct encapsulation if the next hop is on the same OMNI link segment.

Hub Proxy/Server

a single Proxy/Server selected by the Client that provides a designated router and mobility anchor point service for all of the Client's underlying interfaces. Clients normally select the first FHS Proxy/Server they coordinate with to serve in the Hub role, as all FHS Proxy/Servers are equally capable candidates to serve in that capacity.

First-Hop Segment (FHS) Proxy/Server

a Proxy/Server for an underlying interface of the source Client that forwards packets sent by the source Client over that interface into the segment routing topology. FHS Proxy/Servers act as intermediate forwarding nodes to facilitate RS/RA exchanges between a Client and its Hub Proxy/Server.

Last-Hop Segment (LHS) Proxy/Server

a Proxy/Server for an underlying interface of the target Client that forwards packets received from the segment routing topology to the target Client over that interface.

Segment Routing Topology (SRT)

a multinet OMNI link forwarding region between the FHS Proxy/ Server and LHS Proxy/Server. FHS/LHS Proxy/Servers and SRT Bridges span the OMNI link on behalf of source/target Client pairs. The SRT maintains a spanning tree established through BGP peerings between Bridges and Proxy/Servers. Each SRT segment includes Bridges in a "hub" and Proxy/Servers in "spokes", while adjacent segments are interconnected by Bridge-Bridge peerings. The BGP peerings are configured over both secured and unsecured underlying network paths such that a secured spanning tree is available for critical control messages while other messages can use the unsecured spanning tree.

link-layer address

an IP address used as an encapsulation header source or destination address from the perspective of the OMNI interface. When an upper layer protocol (e.g., UDP) is used as part of the encapsulation, the port number is also considered as part of the link-layer address.

network layer address

the source or destination address of an original IP packet presented to the OMNI interface.

end user network (EUN)

an internal virtual or external edge IP network that an AERO Client or Relay connects to the rest of the network via the OMNI interface. The Client/Relay sees each EUN as a "downstream" network, and sees the OMNI interface as the point of attachment to the "upstream" network.

Mobile Node (MN)

an AERO Client and all of its downstream-attached networks that move together as a single unit, i.e., an end system that connects an Internet of Things.

Mobile Router (MR)

a MN's on-board router that forwards original IP packets between any downstream-attached networks and the OMNI link. The MR is the MN entity that hosts the AERO Client.

Route Optimization Source (ROS)

the AERO node nearest the source that initiates route optimization. The ROS may be a FHS Proxy/Server or Relay for the source, or may be the source Client itself.

Route Optimization responder (ROR)

the AERO node that responds to route optimization requests on behalf of the target. The ROR may be a Proxy/Server for a target MNP Client or a Relay for a non-MNP target.

Potential Router List (PRL)

a geographically and/or topologically referenced list of addresses of all Proxy/Servers within the same OMNI link. Each OMNI link has its own PRL.

Distributed Mobility Management (DMM)

a BGP-based overlay routing service coordinated by Proxy/Servers and Bridges that tracks all Proxy/Server-to-Client associations.

Mobility Service (MS)

the collective set of all Proxy/Servers, Bridges and Relays that provide the AERO Service to Clients.

Mobility Service Endpoint MSE)

an individual Proxy/Server, Bridge or Relay in the Mobility Service.

Multilink Forwarding Information Base (MFIB)

A forwarding table on each AERO/OMNI source, destination and intermediate node that includes Multilink Forwarding Vectors (MFV) with both next hop forwarding instructions and context for

reconstructing compressed headers for specific underlying interface pairs used to communicate with peers.

Multilink Forwarding Vector (MFV)
An MFIB entry that includes soft state for each underlying interface pairwise communication session between peer OMNI nodes. MFVs are identified by both a next-hop and previous-hop MFV Index (MFVI), with the next-hop established based on an IPv6 ND solicitation and the previous hop established based on the solicited IPv6 ND advertisement response.

Multilink Forwarding Vector Index (MVFI)
A 4 octet value selected by an AERO/OMNI node when it creates an MFV, then advertised to either a next-hop or previous-hop. AERO/OMNI intermediate nodes assign two distinct local MFVIs for each MFV and advertise one to the next-hop and the other to the previous-hop. AERO/OMNI end systems assign and advertise a single MFVI. AERO/OMNI nodes also discover the remote MFVIs advertised by other nodes that indicate a value the other node is willing to accept.

Throughout the document, the simple terms "Client", "Proxy/Server", "Bridge" and "Relay" refer to "AERO Client", "AERO Proxy/Server", "AERO Bridge" and "AERO Relay", respectively. Capitalization is used to distinguish these terms from other common Internetworking uses in which they appear without capitalization.

The terminology of IPv6 ND [RFC4861] and DHCPv6 [RFC8415] (including the names of node variables, messages and protocol constants) is used throughout this document. The terms "All-Routers multicast", "All-Nodes multicast", "Solicited-Node multicast" and "Subnet-Router anycast" are defined in [RFC4291]. Also, the term "IP" is used to generically refer to either Internet Protocol version, i.e., IPv4 [RFC0791] or IPv6 [RFC8200].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Automatic Extended Route Optimization (AERO)

The following sections specify the operation of IP over OMNI links using the AERO service:

3.1. AERO Node Types

AERO Clients can be deployed as fixed infrastructure nodes close to end systems, or as Mobile Nodes (MNs) that can change their network attachment points dynamically. AERO Clients configure OMNI interfaces over underlying interfaces with addresses that may change due to mobility. AERO Clients register their Mobile Network Prefixes (MNPs) with the AERO service, and distribute the MNPs to nodes on EUNs. AERO Bridges, Proxy/Servers and Relays are critical infrastructure elements in fixed (i.e., non-mobile) INET deployments and hence have permanent and unchanging INET addresses. Together, they constitute the AERO service which provides an OMNI link virtual overlay for connecting AERO Clients.

AERO Bridges (together with Proxy/Servers) provide the secured backbone supporting infrastructure for a Segment Routing Topology (SRT) spanning tree for the OMNI link. Bridges forward carrier packets both within the same SRT segment and between disjoint SRT segments based on an IPv6 encapsulation mid-layer known as the OMNI Adaptation Layer (OAL) [I-D.templin-6man-omni]. The OMNI interface and OAL provide a virtual bridging service, since the inner IP TTL/Hop Limit is not decremented. Each Bridge also peers with Proxy/Servers and other Bridges in a dynamic routing protocol instance to provide a Distributed Mobility Management (DMM) service for the list of active MNPs (see Section 3.2.3). Bridges present the OMNI link as a set of one or more Mobility Service Prefixes (MSPs) and configure secured tunnels with Proxy/Servers, Relays and other Bridges; they further maintain forwarding table entries for each MNP or non-MNP prefix in service on the OMNI link.

AERO Proxy/Servers in distributed SRT segments provide default forwarding and mobility/multilink services for AERO Client mobile nodes. Each Proxy/Server also peers with Bridges in a dynamic routing protocol instance to advertise its list of associated MNPs (see Section 3.2.3). Hub Proxy/Servers provide prefix delegation/registration services and track the mobility/multilink profiles of each of their associated Clients, where each delegated prefix becomes an MNP taken from an MSP. Proxy/Servers at ANET/INET boundaries provide a forwarding service for ANET Clients to communicate with peers in external INETs while Proxy/Servers in the open INET provide an authentication service for INET Client IPv6 ND messages but limited forwarding services. Source Clients securely coordinate with target Clients by sending control messages via a First-Hop Segment (FHS) Proxy/Server which forwards them over the SRT spanning tree to a Last-Hop Segment (LHS) Proxy/Server which finally forwards them to the target.

AERO Relays are Proxy/Servers that provide forwarding services to exchange original IP packets between the OMNI link and other links/networks. Relays run a dynamic routing protocol to discover any non-MNP prefixes in service on other links/networks. The Relay redistributes OMNI link MSP(s) into other links/networks, and redistributes non-MNP prefixes via OMNI link Bridge BGP peerings.

3.2. The AERO Service over OMNI Links

3.2.1. AERO/OMNI Reference Model

Figure 1 presents the basic OMNI link reference model:

```
                          +----------------+
                          | 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      )
      `-(______)-'    +-------+     +-------+     `-(______)-'
```

Figure 1: AERO/OMNI Reference Model

In this model:

- the OMNI link is an overlay network service configured over one or more underlying SRT segments which may be managed by different

administrative authorities and have incompatible protocols and/or addressing plans.

- AERO Bridge B1 aggregates Mobility Service Prefix (MSP) M1, discovers Mobile Network Prefixes (MNPs) X* and advertises the MSP via BGP peerings over secured tunnels to Proxy/Servers (S1, S2). Bridges provide the backbone for an SRT spanning tree for the OMNI link.

- AERO Proxy/Servers S1 and S2 configure secured tunnels with Bridge B1 and also provide mobility, multilink, multicast and default router services for the MNPs of their associated Clients C1 and C2. (Proxy/Servers that act as Relays can also advertise non-MNP routes for non-mobile correspondent nodes the same as for MNP Clients.)

- AERO Clients C1 and C2 associate with Proxy/Servers S1 and S2, respectively. They receive MNP delegations X1 and X2, and also act as default routers for their associated physical or internal virtual EUNs. Simple hosts H1 and H2 attach to the EUNs served by Clients C1 and C2, respectively.

An OMNI link configured over a single *NET appears as a single unified link with a consistent underlying network addressing plan; all nodes on the link can exchange carrier packets via simple *NET encapsulation (i.e., following any necessary NAT traversal) since the underlying *NET is connected. In common practice, however, OMNI links are often configured over an SRT spanning tree that bridges multiple distinct *NET segments managed under different administrative authorities (e.g., as for worldwide aviation service providers such as ARINC, SITA, Inmarsat, etc.). Individual *NETs may also be partitioned internally, in which case each internal partition appears as a separate segment.

The addressing plan of each SRT segment is consistent internally but will often bear no relation to the addressing plans of other segments. Each segment is also likely to be separated from others by network security devices (e.g., firewalls, proxys, packet filtering gateways, etc.), and disjoint segments often have no common physical link connections. Therefore, nodes can only be assured of exchanging carrier packets directly with correspondents in the same segment, and not with those in other segments. The only means for joining the segments therefore is through inter-domain peerings between AERO Bridges.

The OMNI link spans multi-segment SRT topologies using the OMNI Adaptation Layer (OAL) [I-D.templin-6man-omni] to provide the network layer with a virtual abstraction similar to a bridged campus LAN.

The OAL is an OMNI interface sublayer that inserts a mid-layer IPv6 encapsulation header for inter-segment forwarding (i.e., bridging) without decrementing the network-layer TTL/Hop Limit of the original IP packet. An example OMNI link SRT is shown in Figure 2:

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

Figure 2: OMNI Link Segment Routing Topology (SRT)

Bridge, Proxy/Server and Relay OMNI interfaces are configured over both secured tunnels and open INET underlying interfaces within their respective SRT segments. Within each segment, Bridges configure "hub-and-spokes" BGP peerings with Proxy/Server/Relays as "spokes". Adjacent SRT segments are joined by Bridge-to-Bridge peerings to collectively form a spanning tree over the entire SRT. The "secured" spanning tree supports authentication and integrity for critical control plane messages. The "unsecured" spanning tree conveys ordinary carrier packets without security codes and that must be treated by destinations according to data origin authentication procedures. AERO nodes can employ route optimization to cause carrier packets to take more direct paths between OMNI link neighbors without having to follow strict spanning tree paths.

3.2.2. Addressing and Node Identification

AERO nodes on OMNI links use the Link-Local Address (LLA) prefix fe80::/64 [RFC4291] to assign LLAs used for network-layer addresses in link-scoped IPv6 ND and data messages. AERO Clients use LLAs constructed from MNPs (i.e., "MNP-LLAs") while other AERO nodes use LLAs constructed from administrative identification values ("ADM-LLAs") as specified in [I-D.templin-6man-omni]. Non-MNP routes are also represented the same as for MNP-LLAs, but may include a prefix that is not properly covered by an MSP.

AERO nodes also use the Unique Local Address (ULA) prefix fd00::/8 followed by a pseudo-random 40-bit OMNI domain identifier to form the prefix [ULA]::/48, then include a 16-bit OMNI link identifier '*' to form the prefix [ULA*]::/64 [RFC4291]. The AERO node then uses the prefix [ULA*]::/64 to form "MNP-ULAs" or "ADM-ULA"s as specified in [I-D.templin-6man-omni] to support OAL addressing. (The prefix [ULA*]::/64 appearing alone and with no suffix represents "default".) AERO Clients also use Temporary ULAs constructed per [I-D.templin-6man-omni], where the addresses are typically used only in initial control message exchanges until a stable MNP-LLA/ULA is assigned.

AERO MSPs, MNPs and non-MNP routes are typically based on Global Unicast Addresses (GUAs), but in some cases may be based on private-use addresses. A GUA block is also reserved for OMNI link anycast purposes. See [I-D.templin-6man-omni] for a full specification of LLAs, ULAs and GUAs used by AERO nodes on OMNI links.

Finally, AERO Clients and Proxy/Servers configure node identification values as specified in [I-D.templin-6man-omni].

3.2.3. AERO Routing System

The AERO routing system comprises a private Border Gateway Protocol (BGP) [RFC4271] service coordinated between Bridges and Proxy/Servers. The service supports carrier packet forwarding at a layer below IP and does not interact with the public Internet BGP routing system, but supports redistribution of information for other links and networks discovered by Relays.

In a reference deployment, each Proxy/Server is configured as an Autonomous System Border Router (ASBR) for a stub Autonomous System (AS) using a 32-bit AS Number (ASN) [RFC4271] that is unique within the BGP instance, and each Proxy/Server further uses eBGP to peer with one or more Bridges but does not peer with other Proxy/Servers. Each SRT segment in the OMNI link must include one or more Bridges in a "hub" AS, which peer with the Proxy/Servers within that segment as

"spoke" ASes. All Bridges within the same segment are members of the same hub AS, and use iBGP to maintain a consistent view of all active routes currently in service. The Bridges of different segments peer with one another using eBGP.

Bridges maintain forwarding table entries only for the MNP-ULAs corresponding to MNP and non-MNP routes that are currently active, while carrier packets destined to all other MNP-ULAs are dropped with a Destination Unreachable message returned due to the black-hole route. In this way, Proxy/Servers and Relays have only partial topology knowledge (i.e., they only maintain routing information for their directly associated Clients and non-AERO links) and they forward all other carrier packets to Bridges which have full topology knowledge.

Each OMNI link segment assigns a unique ADM-ULA sub-prefix of [ULA*]::/96 known as the "SRT prefix". For example, a first segment could assign [ULA*]::1000/116, a second could assign [ULA*]::2000/116, a third could assign [ULA*]::3000/116, etc. Within each segment, each Proxy/Server configures an ADM-ULA within the segment's SRT prefix, e.g., the Proxy/Servers within [ULA*]::2000/116 could assign the ADM-ULAs [ULA*]::2011/116, [ULA*]::2026/116, [ULA*]::2003/116, etc.

The administrative authorities for each segment must therefore coordinate to assure mutually-exclusive ADM-ULA prefix assignments, but internal provisioning of ADM-ULAs an independent local consideration for each administrative authority. For each ADM-ULA prefix, the Bridge(s) that connect that segment assign the all-zero's address of the prefix as a Subnet Router Anycast address. For example, the Subnet Router Anycast address for [ULA*]::1023/116 is simply [ULA*]::1000.

ADM-ULA prefixes are statically represented in Bridge forwarding tables. Bridges join multiple SRT segments into a unified OMNI link over multiple diverse network administrative domains. They support a virtual bridging service by first establishing forwarding table entries for their ADM-ULA prefixes either via standard BGP routing or static routes. For example, if three Bridges ('A', 'B' and 'C') from different segments serviced [ULA*]::1000/116, [ULA*]::2000/116 and [ULA*]::3000/116 respectively, then the forwarding tables in each Bridge are as follows:

```
A: [ULA*]::1000/116->local, [ULA*]::2000/116->B, [ULA*]::3000/116->C

B: [ULA*]::1000/116->A, [ULA*]::2000/116->local, [ULA*]::3000/116->C

C: [ULA*]::1000/116->A, [ULA*]::2000/116->B, [ULA*]::3000/116->local
```

These forwarding table entries rarely change, since they correspond to fixed infrastructure elements in their respective segments.

MNP (and non-MNP) ULAs are instead dynamically advertised in the AERO routing system by Proxy/Servers and Relays that provide service for their corresponding MNPs. For example, if three Proxy/Servers ('D', 'E' and 'F') service the MNPs 2001:db8:1000:2000::/56, 2001:db8:3000:4000::/56 and 2001:db8:5000:6000::/56 then the routing system would include:

D: [ULA*]:2001:db8:1000:2000/120

E: [ULA*]:2001:db8:3000:4000/120

F: [ULA*]:2001:db8:5000:6000/120

A full discussion of the BGP-based routing system used by AERO is found in [I-D.ietf-rtgwg-atn-bgp].

3.2.4. OMNI Link Forwarding

When the network layer forwards an original IP packet into an OMNI interface, the OMNI Adaptation Layer (OAL) encapsulates the packet to produce an OAL packet [I-D.templin-6man-omni]. This OAL source then fragments the OAL packet while including an identical Identification value for each fragment that must be within the window for the LHS Proxy/Server or the target Client itself. The OAL source also includes an identical Compressed Routing Header with 32-bit ID fields (CRH-32) [I-D.bonica-6man-comp-rtg-hdr] with each fragment if necessary as discussed in Section 3.2.7 and Section 3.14. The OAL source finally encapsulates each resulting OAL fragment in *NET headers to form an OAL carrier packet, with source address set to its own *NET address (e.g., 192.0.2.100) and destination set to the *NET address of the next hop OAL intermediate node or destination (e.g., 192.0.2.1).

The carrier packet encapsulation format in the above example is shown in Figure 3:

```
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |          *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 ~
        ~                               ~
        |                               |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Carrier Packet Format

In this format, the original IP header and packet body/fragment are encapsulated in an OAL IPv6 header prepared according to [RFC2473], the CRH-32 is a Routing Header extension of the OAL header, the Fragment Header identifies each fragment, and the *NET header is prepared as discussed in Section 3.6. The OAL source transmits each such carrier packet into the SRT spanning tree, where they are forwarded over possibly multiple OAL intermediate nodes until they arrive at the OAL destination.

The OMNI link control plane service distributes both Client MNP-ULA prefix information that may change dynamically due to regional node mobility and per-segment ADM-ULA prefix information that rarely changes. OMNI link Bridges and Proxy/Servers use the information to establish and maintain a forwarding plane spanning tree that connects all nodes on the link. The spanning tree supports a carrier packet virtual bridging service according to link-layer (instead of network-layer) information, but may often include longer paths than necessary. Each OMNI interface therefore also includes a Multilink Forwarding Information Base (MFIB) with Multilink Forwarding Vectors (MFVs) that can often provide "shortcuts" instead of always following strict spanning tree paths. As a result, the spanning tree is always

available but OMNI interfaces can often use the MFIB to greatly improve performance and reduce load on critical infrastructure elements.

3.2.5. Segment Routing Topologies (SRTs)

The 64-bit sub-prefixes of [ULA]::/48 identify up to 2^16 distinct Segment Routing Topologies (SRTs). Each SRT is a mutually-exclusive OMNI link overlay instance using a distinct set of ULAs, and emulates a bridged campus LAN service for the OMNI link. In some cases (e.g., when redundant topologies are needed for fault tolerance and reliability) it may be beneficial to deploy multiple SRTs that act as independent overlay instances. A communication failure in one instance therefore will not affect communications in other instances.

Each SRT is identified by a distinct value in bits 48-63 of [ULA]::/48, i.e., as [ULA0]::/64, [ULA1]::/64, [ULA2]::/64, etc. Each OMNI interface is identified by a unique interface name (e.g., omni0, omni1, omni2, etc.) and assigns an OMNI IPv6 anycast address used for OMNI interface determination in Safety-Based Multilink (SBM) as discussed in [I-D.templin-6man-omni]. Each OMNI interface further applies Performance-Based Multilink (PBM) internally.

The Bridges and Proxy/Servers of each independent SRT engage in BGP peerings to form a spanning tree with the Bridges in non-leaf nodes and the Proxy/Servers in leaf nodes. The spanning tree is configured over both secured and unsecured underlying network paths. The secured spanning tree is used to convey secured control messages between Hub, FHS and LHS Proxy/Servers, while the unsecured spanning tree forwards data messages and/or unsecured control messages.

Each SRT segment is identified by a unique ADM-ULA prefix used by all Proxy/Servers and Bridges in the segment. Each AERO node must therefore discover an SRT prefix that correspondents can use to determine the correct segment, and must publish the SRT prefix in IPv6 ND messages.

3.2.6. Segment Routing For OMNI Link Selection

Original IPv6 source can direct IPv6 packets to an AERO node by including a standard IPv6 Segment Routing Header (SRH) [RFC8754] with the OMNI IPv6 anycast address for the selected OMNI link as either the IPv6 destination or as an intermediate hop within the SRH. This allows the original source to determine the specific OMNI link SRT an original IPv6 packet will traverse when there may be multiple alternatives.

When an AERO node processes the SRH and forwards the original IPv6 packet to the correct OMNI interface, the OMNI interface writes the next IPv6 address from the SRH into the IPv6 destination address and decrements Segments Left. If decrementing would cause Segments Left to become 0, the OMNI interface deletes the SRH before forwarding. This form of Segment Routing supports Safety-Based Multilink (SBM).

3.2.7. OMNI Multilink Forwarding

OMNI interfaces include a supplemental forwarding table termed the Multilink Forwarding Information Base (MFIB) that provides shorter paths for carrier packet forwarding based on OMNI neighbor underlying interface pairs. The MFIB contains Multilink Forwarding Vectors (MFVs) indexed by 4-octet values known as MFV Indexes (MFVIs).

OMNI interface "OAL source", "OAL intermediate" and "OAL destination" nodes create MFVs/MFVIs when they process an IPv6 ND solicitation message with Job code "00" (Initialize; Build B) or a solicited advertisement with Job code "01" (Follow B; Build A) (see: [I-D.templin-6man-omni]). The OAL source of the solicitation (and OAL destination of the solicited advertisement) are considered to reside in the "First Hop Segment (FHS)", while the OAL destination of the solicitation (and OAL source of the solicited advertisement) are considered to reside in the "Last Hop Segment (LHS)".

When an OAL node processes a solicitation with Job code "00", it creates an MFV, records the solicitation's source and destination LLAs and assigns a "B" MFVI. When the "B" MVFI is referenced, the MVF presents the LLAs in (dst,src) order the opposite of how they appeared in the original solicitation.

When an OAL node processes a solicited advertisement with Job code "01", it locates the MFV created by the solicitation and assigns an "A" MFVI. When the "A" MFVI is referenced, the MFV presents the LLAs in (src,dst) order the same as they appeared in the original solicitation.

OAL nodes generate random 32-bit values as candidate A/B MFVIs which must first be tested for local uniqueness. If a candidate MFVI s already in use (or if the value is 0), the OAL node repeats the process until it obtains a unique non-zero value. (Since the number of MFVs in service at each OAL node is likely to be much smaller than 2**32, the process will generate a unique value after a small number of tries.) An MFVI generated by a first OAL node SHOULD NOT be tested for uniqueness on other OAL nodes, since the uniqueness property is node-local only.

OAL nodes maintain A/B MFVIs as follows:

- "B1" - a locally-unique MFVI maintained independently by each OAL node on the path from the FHS OAL source to the last LHS intermediate node before the OAL destination. The OAL node generates and assigns a "B1" MFVI to a newly-created MFV when it processes a solicitation message with Job code "00". When the OAL node receives future carrier packets that include this value, it can unambiguously locate the correct MFV and determine directionality without examining addresses.

- "A1" - a locally unique MFVI maintained independently by each OAL node on the path from the LHS OAL source to the last FHS intermediate node before the OAL destination. The OAL node generates and assigns an "A1" MFVI to the MVF that configures the corresponding "B1" MFVI when it processes a solicited advertisement message with Job code "01". When the OAL node receives future carrier packets that include this value, it can unambiguously locate the correct MFV and determine directionality without examining addresses.

- "A2" - the A1 MFVI of a remote OAL node discovered by an FHS OAL source or OAL intermediate node when it processes an advertisement message with Job code "01" that originated from an LHS OAL source. A2 values MUST NOT be tested for uniqueness within the OAL node's local context.

- "B2" - the B1 MFVI of a remote OAL node discovered by an LHS OAL source or OAL intermediate node when it processes a solicitation message with Job code "00" that originated from an FHS OAL source. B2 values MUST NOT be tested for uniqueness within the OAL node's local context.

When an FHS OAL source has an original IP packet to send to an LHS OAL destination, (i.e., one for which there is no existing NCE) it first selects a source and target underlying interface pair. The OAL source uses its cached information for the target underlying interface as LHS information then prepares a solicitation message with an OMNI Multilink Forwarding Parameters sub-option with Job code "00" and with source set to its own {ADM,MNP}-LLA. If the LHS FMT-Forward and FMT-Mode bits are both clear, the OAL source sets the destination to the ADM-LLA of the LHS Proxy/Server; otherwise, it sets the destination to the MNP-LLA of the target Client. The OAL source then sets window synchronization information in the OMNI header and creates a NCE for the selected destination {ADM,MNP}-LLA in the INCOMPLETE state. The OAL source next creates an MFV based on the solicitation source and destination LLAs, then generates a "B1" MFVI and assigns it to the MFV while also including it as the first B entry in the MFVI List. The OAL source then populates the solicitation Multilink Forwarding Parameters based on any FHS/LHS

information it knows locally. OAL intermediate nodes on the path to the OAL destination may populate addtional FHS/LHS information on a hop-by-hop basis.

If the OAL source is the FHS Proxy/Server, it then performs OAL encapsulation/fragmentation while setting the source to its own ADM-ULA and setting the destination to the FHS Subnet Router Anycast ULA determined by applying the FHS SRT prefix length to its ADM-ULA. The FHS Proxy/Server next examines the LHS FMT code. If FMT-Forward is clear and FMT-Mode is set, the FHS Proxy/Server checks for a NCE for the ADM-LLA of the LHS Proxy/Server. If there is no NCE, the LHS Proxy/Server creates one in the INCOMPLETE state. If a new NCE was created (or if the existing NCE requires fresh window synchronization), the FHS Proxy/Server then writes window synchronization parameters into the OMNI Multilink Forwarding Parameters Tunnel Window Synchronization fields. The FHS Proxy/Server then selects an appropriate Identification value and *NET headers and forwards the resulting carrier packets into the secured spanning tree which will deliver them to a Bridge interface that assigns the FHS Subnet Router Anycast ULA.

If the OAL source is the FHS Client, it instead includes an authentication signature if necessary, performs OAL encapsulation/fragmentation, sets the source to its own ADM-ULA and sets the destination to the ADM-ULA of the FHS Proxy/Server. The FHS Client then selects an appropriate Identification value and *NET headers and forwards the carrier packets to the FHS Proxy/Server. When the FHS Proxy/Server receives the carrier packets, it verifies the Identification, reassembles/decapsulates to obtain the solicitation then verifies the authentication signature. The FHS Proxy/Server then creates an MFV (i.e., the same as the FHS Client had done) while assigning the current B entry in the MFVI List (i.e., the one included by the FHS Client) as the "B2" MFVI for this MVF. The FHS Proxy/Server then generates a new unique "B1" MFVI, then both assigns it to the MFV and writes it as the next B entry in the OMNI Multilink Forwarding Parameters MFVI List (while also writing any FHS Client and Proxy/Server addressing information). The FHS Proxy/Server then checks LHS FMT-Forward/Mode to determine whether to create a NCE for the LHS Proxy/Server ADM-LLA and include Tunnel Window Synchronization parameters the same as above. The FHS Proxy/Server then re-encapsulates/re-fragments while setting the source to its own ADM-ULA and destination address to the FHS Subnet Router Anycast ULA. The FHS Proxy/Server finally includes an appropriate Identification value and *NET headers and forwards the carrier packets into the secured spanning tree the same as above.

Bridges in the spanning tree forward carrier packets not explicitly addressed to themselves, while forwarding those that arrived via the

secured spanning tree to the next hop also via the secured spanning tree and forwarding all others via the unsecured spanning tree. When an FHS Bridge receives a carrier packet over the secured spanning tree addressed to its ADM-ULA or the FHS Subnet Router Anycast ULA, it instead reassembles/decapsulates to obtain the solicitation. The FHS Bridge next creates an MFV (i.e., the same as the FHS Proxy/ Server had done) while assigning the current B entry in the MFVI List as the MFV "B2" index. The FHS Bridge also caches the solicitation Multilink Forwarding Parameters FHS information in the MFV, and also caches the first B entry in the MFVI List as "FHS-Client" when FHS FMT-Forward/Mode are both set to enable future direct forwarding to this FHS Client. The FHS Bridge then generates a "B1" MFVI for the MFV and also writes it as the next B entry in the solicitation's MFVI List.

The FHS Bridge then examines the SRT prefixes corresponding to both FHS and LHS. If the FHS Bridge has a local interface connection to both the FHS and LHS (whether they are the same or different segments), the FHS/LHS Bridge caches the solicitation LHS information and writes its ADM-ULA suffix and LHS INADDR into the solicitation OMNI Multilink Forwarding Parameters LHS fields. The FHS/LHS Bridge then re-encapsulates the solicitation with its own ADM-ULA as the source and with the ADM-ULA of the LHS Proxy/Server as the destination. If the FHS and LHS prefixes are different, the FHS Bridge instead re-encapsulates with its own ADM-ULA as the source and with the LHS Subnet Router Anycast ULA as the destination. The FHS Bridge selects an appropriate Identification and *NET headers as above then forwards the carrier packets into the secured spanning tree.

When the FHS and LHS Bridges are different, the LHS Bridge will receive carrier packets over the secured spanning tree from the FHS Bridge. The LHS Bridge reassembles/decapsulates to obtain the solicitation then creates an MFV (i.e., the same as the FHS Bridge had done) while assigning the current B entry in the MFVI List as the MFV "B2" index. The LHS Bridge also caches the ADM-ULA of the FHS Bridge as the spanning tree address for "B2", caches the solicitation Multilink Forwarding Parameters LHS information then generates a "B1" MFVI for the MFV while also writing it as the next B entry in the MFVI List. The LHS Bridge also writes its own ADM-ULA suffix and LHS INADDR into the OMNI Multilink Forwarding Parameters. The LHS Bridge then re-encapsulates with its own ADM-ULA as the source and the ADM-ULA of the LHS Proxy/Server as the destination, then selects an appropriate Identification and *NET headers and forwards the carrier packets into the secured spanning tree.

When the LHS Proxy/Server receives the carrier packets from the secured spanning tree, it reassembles/decapsulates to obtain the

solicitation then verifies that the LHS information supplied by the FHS source is consistent with its own cached information. If the information is consistent, the LHS Proxy/Server then creates an MFV and assigns the current B entry in the MFVI List as the "B2" MFVI the same as for the prior hop. If the solicitation destination is the MNP-LLA of the target Client, the LHS Proxy/Server also generates a "B1" MFVI and assigns it both to the MFVI and as the next B entry in the MFVI List. The LHS Proxy/Server then examines FHS FMT; if FMT-Forward is clear and FMT-Mode is set, the LHS Proxy/Server creates a NCE for the ADM-LLA of the FHS Proxy/Server (if necessary) and sets the state to STALE, then caches any Tunnel Window Synchronization parameters.

If the solicitation destination is its own ADM-LLA, the LHS Proxy/Server next prepares to return a solicited advertisement with Job code "01". If the solicitation source was the MNP-LLA of the FHS Client, the LHS Proxy/Server first creates or updates an NCE for the MNP-LLA with state set to STALE. The LHS Proxy/Server next caches the solicitation OMNI header window synchronization parameters and Multilink Forwarding Parameters information (including the MFVI List) in the NCE corresponding to the LLA source. When the LHS Proxy/Server forwards future carrier packets based on the NCE, it can populate reverse-path forwarding information in a CRH-32 routing header to enable forwarding based on the cached MFVI List B entries instead of ULA addresses.

The LHS Proxy/Server then creates an advertisement with Job code "01" while copying the solicitation OMNI Multilink Forwarding Parameters FHS/LHS information into the corresponding fields in the advertisement. The LHS Proxy/Server then generates an "A1" MFVI and both assigns it to the MFV and includes it as the first A entry in advertisement's MFVI List (see: [I-D.templin-6man-omni] for details on MFVI List A/B processing). The LHS Proxy/Server then includes end-to-end window synchronization parameters in the OMIN header (if necessary) and also tunnel window synchronization parameters in the Multilink Forwarding Parameters Tunnel block (if necessary). The LHS Proxy/Server then encapsulates the advertisement, sets the source to its own ADM-ULA, sets the destination to the ADM-ULA of the LHS Bridge, selects an appropriate Identification value and *NET headers then forwards the carrier packets into the secured spanning tree.

If the solicitation destination was the MNP-LLA of the LHS Client, the LHS Proxy/Server instead includes an authentication signature in the solicitation, then re-encapsulates/re-fragments with its own ADM-ULA as the source and the MNP-ULA of the LHS Client as the destination. The LHS Proxy/Server then selects an appropriate Identification value and *NET headers and forwards the carrier packets to the LHS Client. When the LHS Client receives the carrier

packets, it verifies the Identification and reassembles/decapsulates to obtain the solicitation. The LHS Client then creates a NCE for the solicitation LLA source address in the STALE state. If LHS FMT-Forward is set, FHS FMT-Forward is clear and the solicitation source was an MNP-LLA, the Client also creates a NCE for the ADM-LLA of the FHS Proxy/Server in the STALE state and caches any Tunnel Window Synchronization parameters. The Client then caches the solicitation OMNI header window synchronization parameters and Multilink Forwarding Parameters in the NCE corresponding to the solicitation LLA source, then creates an MFV and assigns both the current MFVI List B entry as "B2" and a locally generated "A1" MFVI the same as for previous hops (the LHS Client also includes the "A1" value in the solicited advertisement - see above and below). The LHS Client also caches the previous MFVI List B entry as "LHS-Bridge" since it can include this value when it sends future carrier packets directly to the Bridge (following appropriate neighbor coordination).

The LHS Client then prepares an advertisement using exactly the same procedures as for the LHS Proxy/Server above, except that it uses its MNP-LLA as the source. The LHS Client also includes an authentication signature, then encapsulates the advertisement with source set to its own ADM-ULA and destination set to the ADM-ULA of the LHS Proxy/Server. The LHS Client then includes an appropriate Identification and *NET headers and forwards the carrier packets to the LHS Proxy/Server. When the LHS Proxy/Server receives the carrier packets, it verifies the Identifications, reassembles/decapsulates to obtain the advertisement, verifies the authentication signature, then uses the current MVFI List B entry to locate the MFV. The LHS Proxy/Server then writes the current MFVI List A entry as the "A2" value for the MVF, generates an "A1" MFVI and both assigns it to the MFV and writes it as the next MFVI List A entry. The LHS Proxy/Server then examines the FHS/LHS FMT codes to determine if it needs to include Tunnel window synchronization parameters. The LHS Proxy/Server then re-encapsulates/re-fragments the advertisement, sets the OAL source to its own ADM-ULA and destination to the ADM-ULA of the LHS Bridge, includes an appropriate Identification and *NET headers and forwards the carrier packets into the secured spanning tree.

When the LHS Bridge receives the carrier packets, it reassembles/decapsulates to obtain the advertisement then uses the current MFVI List B entry to locate the MFV. The LHS Bridge then writes the current MFVI List A entry as the MFV "A2" index and generates a new "A1" value which it both assigns the MFV and writes as the next MFVI List A entry. (The LHS Bridge also caches the first A entry in the MFVI List as "LHS-Client" when LHS FMT-Forward/Mode are both set to enable future direct forwarding to this LHS Client.) If the LHS Bridge is connected directly to both the FHS and LHS segments (whether the segments are the same or different), the FHS/LHS Bridge

will have already cached the FHS/LHS information based on the original solicitation. The FHS/LHS Bridge then re-encapsulates the solicitation with its own ADM-ULA as the source and with the ADM-ULA of the FHS Proxy/Server as the destination. If the FHS and LHS prefixes are different, the FHS Bridge instead re-encapsulates/re-fragments with its own ADM-ULA as the source and with the ADM-ULA of the FHS Bridge as the destination. The LHS Bridge selects an appropriate Identification and *NET headers then forwards the carrier packets into the secured spanning tree.

When the FHS and LHS Bridges are different, the FHS Bridge will receive the carrier packets from the LHS Bridge over the secured spanning tree. The FHS Bridge reassembles/decapsulates to obtain the advertisement, then locates the MFV based on the current MFVI List B entry. The FHS Bridge then assigns the current MFVI List A entry as the MFV "A2" index and caches the ADM-ULA of the LHS Bridge as the spanning tree address for "A2". The FHS Bridge then generates an "A1" MVFI and both assigns it to the MVF and writes it as the next MFVI List A entry while also writing its ADM-ULA and INADDR in the advertisement FHS Bridge fields. The FHS Bridge then re-encapsulates/re-fragments with its own ADM-ULA as the source, with the ADM-ULA of the FHS Proxy/Server as the destination, then selects an appropriate Identification value and *NET headers and forwards the carrier packets into the secured spanning tree.

When the FHS Proxy/Server receives the carrier packets from the secured spanning tree, it reassembles/decapsulates to obtain the advertisement then locates the MFV based on the current MFVI List B entry. The FHS Proxy/Server then assigns the current MFVI List A entry as the "A2" MFVI the same as for the prior hop. If the advertisement destination is its own ADM-LLA, the FHS Proxy/Server then caches the advertisement Multilink Forwarding Parameters with the MFV and examines LHS FMT. If FMT-Forward is clear, the FHS Proxy/Server locates the NCE for the ADM-LLA of the LHS Proxy/Server and sets the state to REACHABLE then caches any Tunnel Window Synchronization parameters. If the advertisement source is the MNP-LLA of the LHS Client, the FHS Proxy/Server then locates the LHS Client NCE and sets the state to REACHABLE then caches the OMNI header window synchronization parameters and prepares to return an NA acknowledgement, if necessary.

If the advertisement destination is the MNP-LLA of the FHS Client, the FHS Proxy/Server also searches for and updates the NCE for the ADM-LLA of the LHS Proxy/Server if necessary the same as above. The FHS Proxy/Server then generates an "A1" MFVI and assigns it both to the MFVI and as the next MFVI List A entry, then includes an authentication signature in the advertisement message. The FHS Proxy/Server then re-encapsulates/re-fragments with its own ADM-ULA

as the source, with the MNP-ULA of the FHS Client as the destination, then selects an appropriate Identification value and *NET headers and forwards the carrier packets to the FHS Client.

When the FHS Client receives the carrier packets, it verifies the Identification, reassembles/decapsulates to obtain the advertisement then locates the MFV based on the current MFVI List B entry. The FHS Client then assigns the current MFVI List A entry as the "A2" MFVI the same as for the prior hop. The FHS Client then caches the advertisement Multilink Forwarding Parameters (including the MFVI List) with the MFV and examines LHS FMT. If FMT-Forward is clear, the FHS Client locates the NCE for the ADM-LLA of the LHS Proxy/Server and sets the state to REACHABLE then caches any Tunnel Window Synchronization parameters. If the advertisement source is the MNP-LLA of the LHS Client, the FHS Proxy/Server then locates the LHS Client NCE and sets the state to REACHABLE then caches the OMNI header window synchronization parameters and prepares to return an NA acknowledgement, if necessary. The FHS Client also caches the previous MFVI List A entry as "FHS-Bridge" since it can include this value when it sends future carrier packets directly to the Bridge (following appropriate neighbor coordination).

When either the FHS Client or FHS Proxy/Server needs to return an NA acknowledgement to complete window synchronization, it prepares an acknowledgement message with an OMNI Multilink Forwarding Parameters sub-option with Job code set to "10" (Follow A; Record B). The FHS node then includes Tunnel Window Synchronization parameters if necessary and sets the MFVI List to the cached list of A entries received in the LHS advertisement, but need not set any other FHS/LHS information. If the FHS node is the Client, it next includes an authentication signature then encapsulates/fragments with its own MNP-ULA as the source and the ADM-ULA of the FHS Proxy/Server as the destination, then selects an appropriate Identification value and *NET headers and forwards the carrier packets to the FHS Proxy/Server. The FHS Proxy/Server then verifies the Identification, reassembles/decapsulates, verifies the authentication signature and uses the current MFVI List A entry to locate the MFV. The FHS Proxy/Server then writes its "B1" MFVI as the next MFVI List B entry and determines whether it needs to include Tunnel Window Synchronization parameters the same as it had done when it forwarded the original solicitation.

The FHS Proxy/Server then re-encapsulates/re-fragments with its own ADM-ULA as the source and the ADM-ULA of the FHS Bridge as the destination, then selects an appropriate Identification and *NET headers and forwards the carrier packets into the secured spanning tree. When the FHS Bridge receives the carrier packets, it reassembles/decapsulates then uses the current MFVI List A entry to

locate the MFV. The FHS Bridge then writes its "B1" MFVI as the next MFVI List B entry. The FHS Bridge then re-encapsulates/re-fragments with its own ADM-ULA as the source and the ADM-ULA of the LHS Proxy/Server as the destination. If the FHS Bridge is also the LHS Bridge, it sets the ADM-ULA of the LHS Proxy/Server as the destination; otherwise it sets the ADM-ULA of the LHS Bridge. The FHS Bridge then selects an appropriate Identification and *NET headers and forwards the carrier packets into the secured spanning tree. If an LHS Bridge receives the carrier packets, it processes them exactly the same as the FHS Bridge had done while setting the carrier packet source to its own ADM-ULA and destination to the ADM-ULA of the LHS Proxy/Server.

When the LHS Proxy/Server receives the carrier packets, it reassembles/decapsulates to obtain the NA acknowledgement message. The LHS Proxy/Server then locates the MFV based on the current MFVI List A entry then determines whether it is a tunnel ingress the same as for the original solicitation. If it is a tunnel ingress, the LHS Proxy/Server updates the NCE for the tunnel far-end based on the Tunnel Window Synchronization parameters in the NA. If the NA destination is its own ADM-LLA, the LHS Proxy/Server next updates the NCE for the NA source LLA based on the OMNI header Window Synchronization parameters and MAY compare the MVFI List to the version it had cached in the MFV based on the original solicitation.

If the NA destination is the MNP-LLA of the LHS Client, the LHS Proxy/Server instead writes its "B1" MFV as the next MFVI List B entry, includes an authentication signature, re-encapsulates/re-fragments with its own ADM-ULA as the source and the MNP-ULA of the Client as the destination then selects an appropriate Identification and *NET headers and forwards the resulting carrier packets to the LHS Client. When the LHS Client receives the carrier packets, it verifies the Identification, reassembles/decapsulates to obtain the NA acknowledgement, verifies the authentication signature then processes the message exactly the same as for the LHS Proxy/Server case above.

Following the solicitation/advertisement/acknowledgement exchange, OAL end systems and tunnel endpoints can begin exchanging ordinary carrier packets with Identification values within their respective send/receive windows without requiring security signatures and/or secured spanning tree traversal. Either peer can refresh window synchronization parameters and/or send other carrier packets requiring security at any time using the same secured procedures described above. OAL end systems and intermediate nodes can also use their own A1/B1 MFVIs when they receive carrier packets to unambiguously locate the correct MFV and determine directionality and can use any discovered A2/B2 MFVIs to forward carrier packets to

other OAL nodes that configure the corresponding A1/B1 MFVIs. When an OAL node uses an MFVI included in a carrier packet to locate an MFV, it need not also examine the carrier packet addresses.

OAL sources can also begin including CRH-32s in carrier packets with a list of A/B MFVIs that OAL intermediate nodes can use for shortest-path carrier packet forwarding based on MFVIs instead of spanning tree addresses. OAL sources and intermediate nodes can also begin forwarding carrier packets with compressed headers (see: [I-D.templin-6man-omni]) that include only a single A/B MFVI meaningful to the next hop, since all nodes in the path up to (and sometimes including) the OAL destination have already established MFV forwarding information. Note that when an FHS OAL source receives a solicited advertisement with Job code "01', the message will contain an MFVI List with A entries populated in the reverse order needed for populating a CRH-32 routing header. The FHS OAL source must therefore write the MFVI List A entries last-to-first when it populates a CRH-32, or must select the correct A entry to include in a compressed OAL header based on the intended OAL intermediate node or destination.

When a Bridge receives unsecured carrier packets destined to a local segment Client that has asserted direct reachability, the Bridge employs NAT traversal procedures to enable direct carrier packet forwarding while bypassing the local Proxy/Server based on the Client's advertised MFVIs and discovered NATed L2ADDR information. If the Client cannot be reached directly (or if NAT traversal has not yet converged), the Bridge instead forwards carrier packets directly to the local Proxy/Server.

When a Proxy/Server receives carrier packets destined to a local Client or forwards carrier packets received from a local Client, it first locates the correct MFV. If the carrier packets include a secured IPv6 ND message, the Proxy/Server uses the Client's MVF established through RS/RA exchanges to re-encapsulate/re-fragment while forwarding outbound secured carrier packets via the secured spanning tree and forwarding inbound secured carrier packets while including an authentication signature. For ordinary carrier packets, the Proxy/Server uses the same MFV if directed by MFVI and/or OAL addressing. Otherwise it locates an MFV established through an NS/NA exchange between the Client and the remote peer, and forwards the carrier packets without first reassembling/decapsulating.

When a Proxy/Server or Client configured as a tunnel ingress receives a carrier packet with a full header with an MNP-ULA source or a compressed header with an MFVI that matches an MFV, the ingress encapsulates the carrier packet in a new OAL full or compressed header with the inner header containing Identification values

appropriate for the end-to-end window and the outer header containing an Identification value appropriate for the tunnel endpoints. When a Proxy/Server or Client configured as a tunnel egress receives an encapsulated carrier packet, it verifies the Identification in the outer header, then discards the outer header and forwards the inner carrier packet to the final destination.

When a source Client forwards carrier packets it can employ header compression according to the MFVIs established through an NS/NA exchange with a remote or local peer. When the source Client forwards to a remote peer, it can forward carrier packets to a local SRT Bridge (following the establishment of L2ADDR information) while bypassing the Proxy/Server. When a target Client receives carrier packets that match a local MFV, the Client first verifies the Identification then decompresses the headers if necessary, reassembles if necessary to obtain the OAL packet then decapsulates and delivers the IP packet to upper layers.

When synchronized peer Clients in the same SRT segment with FMT-Forward and FMT-Mode set discover each other's NATed L2ADDR addresses through NAT traversal, they can exchange carrier packets directly with header compression using MFVIs discovered as above. The FHS Client will have cached the A MFVI for the LHS Client, which will have cached the B MVFI for the FHS Client.

3.3. OMNI Interface Characteristics

OMNI interfaces are virtual interfaces configured over one or more underlying interfaces classified as follows:

- INET interfaces connect to an INET either natively or through one or more NATs. Native INET interfaces have global IP addresses that are reachable from any INET correspondent. The INET-facing interfaces of Proxy/Servers are native interfaces, as are Relay and Bridge interfaces. NATed INET interfaces connect to a private network behind one or more NATs that provide INET access. Clients that are behind a NAT are required to send periodic keepalive messages to keep NAT state alive when there are no carrier packets flowing.

- ANET interfaces connect to an ANET that is separated from the open INET by an FHS Proxy/Server. Clients can issue control messages over the ANET without including an authentication signature since the ANET is secured at the network layer or below. Proxy/Servers can actively issue control messages over the INET on behalf of ANET Clients to reduce ANET congestion.

- o VPNed interfaces use security encapsulation over the INET to a Virtual Private Network (VPN) server that also acts as an FHS Proxy/Server. Other than the link-layer encapsulation format, VPNed interfaces behave the same as Direct interfaces.

- o Direct (i.e., single-hop point-to-point) interfaces connect a Client directly to an FHS Proxy/Server without crossing any ANET/INET paths. An example is a line-of-sight link between a remote pilot and an unmanned aircraft. The same Client considerations apply as for VPNed interfaces.

OMNI interfaces use OAL encapsulation and fragmentation as discussed in Section 3.2.4. OMNI interfaces use *NET encapsulation (see: Section 3.6) to exchange carrier packets with OMNI link neighbors over INET or VPNed interfaces as well as over ANET interfaces for which the Client and FHS Proxy/Server may be multiple IP hops away. OMNI interfaces do not use link-layer encapsulation over Direct underlying interfaces or ANET interfaces when the Client and FHS Proxy/Server are known to be on the same underlying link.

OMNI interfaces maintain a neighbor cache for tracking per-neighbor state the same as for any interface. OMNI interfaces use IPv6 ND messages including Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS) and Neighbor Advertisement (NA) for neighbor cache management. In environments where spoofing may be a threat, OMNI neighbors should employ OAL Identification window synchronization in their IPv6 ND message exchanges.

OMNI interfaces send IPv6 ND messages with an OMNI option formatted as specified in [I-D.templin-6man-omni]. The OMNI option includes prefix registration information, Multilink Forwarding Parameters containing link information parameters for the OMNI interface's underlying interfaces and any other per-neighbor information.

A Client's OMNI interface may be configured over multiple underlying interfaces. For example, common mobile handheld devices have both wireless local area network ("WLAN") and cellular wireless links. These links are often used "one at a time" with low-cost WLAN preferred and highly-available cellular wireless as a standby, but a simultaneous-use capability could provide benefits. In a more complex example, aircraft frequently have many wireless data link types (e.g. satellite-based, cellular, terrestrial, air-to-air directional, etc.) with diverse performance and cost properties.

If a Client's multiple underlying interfaces are used "one at a time" (i.e., all other interfaces are in standby mode while one interface is active), then successive IPv6 ND messages all include OMNI option Multilink Forwarding Parameters sub-options with the same underlying

interface index. In that case, the Client would appear to have a single underlying interface but with a dynamically changing link-layer address.

If the Client has multiple active underlying interfaces, then from the perspective of IPv6 ND it would appear to have multiple link-layer addresses. In that case, IPv6 ND message OMNI options MAY include Multilink Forwarding Parameters sub-options with different underlying interface indexes.

Bridge and Proxy/Server OMNI interfaces are configured over underlying interfaces that provide both secured tunnels for carrying IPv6 ND and BGP protocol control plane messages and open INET access for carrying unsecured messages. The OMNI interface configures both an ADM-LLA and its corresponding ADM-ULA, and acts as an OAL source to encapsulate and fragment original IP packets while presenting the resulting carrier packets over the secured or unsecured underlying paths. Note that Bridge and Proxy/Server BGP protocol TCP sessions are run directly over the OMNI interface and use ADM-ULA source and destination addresses. The OMNI interface employs the OAL to encapsulate the original IP packets for these sessions as carrier packets (i.e., even though the OAL header may use the same ADM-ULAs as the original IP header) and forwards them over the secured underlying path.

3.4. OMNI Interface Initialization

AERO Proxy/Servers and Clients configure OMNI interfaces as their point of attachment to the OMNI link. AERO nodes assign the MSPs for the link to their OMNI interfaces (i.e., as a "route-to-interface") to ensure that original IP packets with destination addresses covered by an MNP not explicitly associated with another interface are directed to an OMNI interface.

OMNI interface initialization procedures for Proxy/Servers, Clients and Bridges are discussed in the following sections.

3.4.1. AERO Proxy/Server and Relay Behavior

When a Proxy/Server enables an OMNI interface, it assigns an ADM-{LLA,ULA} appropriate for the given OMNI link SRT segment. The Proxy/Server also configures secured tunnels with one or more neighboring Bridges and engages in BGP routing protocol sessions with one or more Bridges.

The OMNI interface provides a single interface abstraction to the IP layer, but internally includes an NBMA nexus for sending carrier packets to OMNI interface neighbors over underlying INET interfaces

and secured tunnels. The Proxy/Server further configures a service to facilitate IPv6 ND exchanges with AERO Clients and manages per-Client neighbor cache entries and IP forwarding table entries based on control message exchanges.

Relays are simply Proxy/Servers that run a dynamic routing protocol to redistribute routes between the OMNI interface and INET/EUN interfaces (see: Section 3.2.3). The Relay provisions MNPs to networks on the INET/EUN interfaces (i.e., the same as a Client would do) and advertises the MSP(s) for the OMNI link over the INET/EUN interfaces. The Relay further provides an attachment point of the OMNI link to a non-MNP-based global topology.

3.4.2. AERO Client Behavior

When a Client enables an OMNI interface, it assigns either an MNP-{LLA, ULA} or a Temporary ULA and sends RS messages over its underlying interfaces to an FHS Proxy/Server, which returns an RA message with corresponding parameters. The RS/RA messages may pass through one or more NATs in the case of a Client's INET interface. (Note: if the Client used a Temporary ULA in its initial RS message, it will discover an MNP-{LLA, ULA} in the corresponding RA that it receives from the FHS Proxy/Server and begin using these new addresses. If the Client is operating outside the context of AERO infrastructure such as in a Mobile Ad-hoc Network (MANET), however, it may continue using Temporary ULAs for Client-to-Client communications until it encounters an infrastructure element that can provide an MNP.)

3.4.3. AERO Bridge Behavior

AERO Bridges configure an OMNI interface and assign an ADM-ULA and corresponding Subnet Router Anycast address for each OMNI link SRT segment they connect to. Bridges configure secured tunnels with Proxy/Servers in the same SRT segment and other Bridges in the same (or an adjacent) SRT segment. Bridges then engage in a BGP routing protocol session with neighbors over the secured spanning tree (see: Section 3.2.3).

3.5. OMNI Interface Neighbor Cache Maintenance

Each OMNI interface maintains a conceptual neighbor cache that includes a Neighbor Cache Entry (NCE) for each of its active neighbors on the OMNI link per [RFC4861]. Each NCE is indexed by the LLA of the neighbor, while the OAL encapsulation ULA determines the context for Identification verification. Clients and Proxy/Servers maintain NCEs through RS/RA exchanges, and also maintain NCEs for any active correspondent peers through NS/NA exchanges.

Bridges also maintain NCEs for Clients within their local segments based on NS/NA(WIN) route optimization. When a Bridge creates/ updates a NCE for a local segment Client based on NS/NA(WIN) route optimization, it also maintains MVFI and L2ADDR state for messages destined to this local segment Client.

Hub Proxy/Servers add an additional state DEPARTED to the list of NCE states found in Section 7.3.2 of [RFC4861]. When a Client terminates its association, the Proxy/Server OMNI interface sets a "DepartTime" variable for the NCE to "DEPART_TIME" seconds. DepartTime is decremented unless a new IPv6 ND message causes the state to return to REACHABLE. While a NCE is in the DEPARTED state, the Proxy/Server forwards carrier packets destined to the target Client to the Client's new location instead. When DepartTime decrements to 0, the NCE is deleted. It is RECOMMENDED that DEPART_TIME be set to the default constant value REACHABLE_TIME plus 10 seconds (40 seconds by default) to allow a window for carrier packets in flight to be delivered while stale route optimization state may be present.

Hub Proxy/Servers act as RORs on behalf of their associated Clients according to the Proxy Neighbor Advertisement specification in Section 7.2.8 of [RFC4861]. When a Hub Proxy/Server ROR receives an authentic NS(AR) message, it first searches for a NCE for the target Client and accepts the message only if there is an entry. The Hub Proxy/Server then returns a solicited NA(AR) message while creating or updating a "Report List" entry in the target Client's NCE that caches both the LLA and ULA of ROS with a "ReportTime" variable set to REPORT_TIME seconds. The ROR resets ReportTime when it receives a new authentic NS(AR) message, and otherwise decrements ReportTime while no authentic NS(AR) messages have been received. It is RECOMMENDED that REPORT_TIME be set to the default constant value REACHABLE_TIME plus 10 seconds (40 seconds by default) to allow a window for route optimization to converge before ReportTime decrements below REACHABLE_TIME.

When the ROS receives a solicited NA(AR) message response to its NS(AR), it creates or updates a NCE for the target network-layer and link-layer addresses. The ROS then (re)sets ReachableTime for the NCE to REACHABLE_TIME seconds and performs reachability tests over specific underlying interface pairs to determine paths for forwarding carrier packets directly to the target. The ROS otherwise decrements ReachableTime while no further solicited NA messages arrive. It is RECOMMENDED that REACHABLE_TIME be set to the default constant value 30 seconds as specified in [RFC4861].

AERO nodes also use the value MAX_UNICAST_SOLICIT to limit the number of NS messages sent when a correspondent may have gone unreachable, the value MAX_RTR_SOLICITATIONS to limit the number of RS messages

sent without receiving an RA and the value MAX_NEIGHBOR_ADVERTISEMENT to limit the number of unsolicited NAs that can be sent based on a single event. It is RECOMMENDED that MAX_UNICAST_SOLICIT, MAX_RTR_SOLICITATIONS and MAX_NEIGHBOR_ADVERTISEMENT be set to 3 the same as specified in [RFC4861].

Different values for the above constants MAY be administratively set; however, if different values are chosen, all nodes on the link MUST consistently configure the same values. Most importantly, DEPART_TIME and REPORT_TIME SHOULD be set to a value that is sufficiently longer than REACHABLE_TIME to avoid packet loss due to stale route optimization state.

3.5.1. OMNI ND Messages

OMNI interfaces prepare IPv6 ND messages the same as for standard IPv6 ND, but also include a new option type termed the OMNI option [I-D.templin-6man-omni]. OMNI interfaces prepare IPv6 ND messages the same as for standard IPv6 ND, and include one or more OMNI options and any other options then completely populate all option information. If the OMNI interface includes an authentication signature, it sets the IPv6 ND message Checksum field to 0 and calculates the authentication signature over the entire length of the message (beginning with a pseudo-header of the IPv6 header) but does not then proceed to calculate the IPv6 ND message checksum itself. If the OMNI interface forwards the message to a next hop over the secured spanning tree path, it omits both the authentication signature and checksum since lower layers already ensure authentication and integrity. In all other cases, the OMNI interface calculates the standard IPv6 ND message checksum and writes the value in the Checksum field. OMNI interfaces verify authentication and integrity of each IPv6 ND message received according to the specific check(s) included, and process the message further only following verification.

OMNI options include per-neighbor information that provides multilink forwarding, link-layer address and traffic selector information for the neighbor's underlying interfaces. This information is stored in the neighbor cache and provides the basis for the forwarding algorithm specified in Section 3.10. The information is cumulative and reflects the union of the OMNI information from the most recent IPv6 ND messages received from the neighbor; it is therefore not required that each IPv6 ND message contain all neighbor information.

The OMNI option is distinct from any Source/Target Link-Layer Address Options (S/TLLAOs) that may appear in an IPv6 ND message according to the appropriate IPv6 over specific link layer specification (e.g., [RFC2464]). If both an OMNI option and S/TLLAO appear, the former

pertains to encapsulation addresses while the latter pertains to the native L2 address format of the underlying media

OMNI interface IPv6 ND messages may also include other IPv6 ND options. In particular, solicitation messages may include Nonce and/or Timestamp options if required for verification of advertisement replies. If an OMNI IPv6 ND solicitation message includes a Nonce option, the advertisement reply must echo the same Nonce. If an OMNI IPv6 ND solicitation message includes a Timestamp option, the advertisement reply should also include a Timestamp option.

AERO Clients send RS messages to the All-Routers multicast address or an ADM-LLA while using unicast link-layer addresses. AERO Proxy/Servers respond by returning unicast RA messages. During the RS/RA exchange, AERO Clients and Proxy/Servers include state synchronization parameters to establish Identification windows and other state.

AERO nodes use NS/NA messages for the following purposes:

o NS/NA(AR) messages are used for address resolution only. The ROS sends an NS(AR) to the solicited-node multicast address of the target, and the current ROR with addressing information for the target returns a unicast NA(AR). The NA(AR) contains current, consistent and authentic target address resolution information, but only an implicit third-party assertion of target reachability. NS/NA(AR) messages must be secured.

o NS/NA(WIN) messages are used for establishing and maintaining window synchronization state (and/or any other state such as Interface Attributes). The source sends an NS(WIN) to the unicast address of the target, and the target returns a unicast NA(WIN). The NS/NA(WIN) exchange synchronizes the sequence number windows for Identification values the neighbors will include in subsequent carrier packets, and asserts reachability for the target without necessarily testing a specific underlying interface pair. NS/NA(WIN) messages must be secured.

o NS/NA(NUD) messages are used for determining target reachability. The source sends an NS(NUD) to the unicast address of the target while naming a specific underlying interface pair, and the target returns a unicast NA(NUD). NS/NA(NUD) messages that use an in-window sequence number and do not update any other state need not be secured but should include an IPv6 ND message checksum. NS/NA(NUD) messages may also be used in combination with window synchronization (i.e., NUD+WIN), in which case the messages must be secured.

- Unsolicited NA (uNA) messages are used to signal addressing and/or other neighbor state changes (e.g., address changes due to mobility, signal degradation, traffic selector updates, etc.). uNA messages that include state update information must be secured.

- NS/NA(DAD) messages are not used in AERO, since Duplicate Address Detection is not required.

Additionally, nodes may send NA/RA messages with the OMNI option PNG flag set to receive a solicited NA response from the neighbor. The solicited NA response MUST set the ACK flag (without also setting the SYN or PNG flags) and include the Identification used in the PNG message in the Acknowledgement.

3.5.2. OMNI Neighbor Advertisement Message Flags

As discussed in Section 4.4 of [RFC4861] NA messages include three flag bits R, S and O. OMNI interface NA messages treat the flags as follows:

- R: The R ("Router") flag is set to 1 in the NA messages sent by all AERO/OMNI node types. Simple hosts that would set R to 0 do not occur on the OMNI link itself, but may occur on the downstream links of Clients and Relays.

- S: The S ("Solicited") flag is set exactly as specified in Section 4.4. of [RFC4861], i.e., it is set to 1 for Solicited NAs and set to 0 for uNAs (both unicast and multicast).

- O: The O ("Override") flag is set to 0 for solicited NAs returned by a Proxy/Server ROR and set to 1 for all other solicited and unsolicited NAs. For further study is whether solicited NAs for anycast targets apply for OMNI links. Since MNP-LLAs must be uniquely assigned to Clients to support correct IPv6 ND protocol operation, however, no role is currently seen for assigning the same MNP-LLA to multiple Clients.

3.5.3. OMNI Neighbor Window Synchronization

In secured environments (e.g., between secured spanning tree neighbors, between neighbors on the same secured ANET, etc.), OMNI interface neighbors can exchange OAL packets using randomly-initialized and monotonically-increasing Identification values (modulo 2*32) without window synchronization. In environments where spoofing is considered a threat, OMNI interface neighbors instead invoke window synchronization in NS/NA(WIN) message exchanges to maintain send/receive window state in their respective neighbor cache entries as specified in [I-D.templin-6man-omni].

3.6. OMNI Interface Encapsulation and Re-encapsulation

The OMNI interface admits original IP packets then acts as an OAL source to perform OAL encapsulation and fragmentation as specified in [I-D.templin-6man-omni] while including a CRH-32 if necessary as specified in Section 3.2.4. The OAL encapsulates original IP packets to form OAL packets subject to fragmentation, then encapsulates the resulting OAL fragments in *NET headers as carrier packets.

For carrier packets undergoing re-encapsulation at an OAL intermediate node, the OMNI interface decrements the OAL IPv6 header Hop Limit and discards the carrier packet if the Hop Limit reaches 0. The intermediate node next removes the *NET encapsulation headers from the first segment and re-encapsulates the packet in new *NET encapsulation headers for the next segment.

When an FHS Bridge receives a carrier packet with a compressed header that must be forwarded to an LHS Bridge over the unsecured spanning tree, it reconstructs the headers based on MFV state, inserts a CRH-32 immediately following the OAL header and adjusts the OAL payload length and destination address field. The FHS Bridge includes a single MFVI in the CRH-32 that will be meaningful to the LHS Bridge. When the LHS Bridge receives the carrier packet, it locates the MFV for the next hop based on the CRH-32 MFVI then re-applies header compression (resulting in the removal of the CRH-32) and forwards the carrier packet to the next hop.

3.7. OMNI Interface Decapsulation

OMNI interfaces (acting as OAL destinations) decapsulate and reassemble OAL packets into original IP packets destined either to the AERO node itself or to a destination reached via an interface other than the OMNI interface the original IP packet was received on. When carrier packets containing OAL fragments addressed to itself arrive, the OMNI interface discards the NET encapsulation headers and reassembles as discussed in Section 3.9.

3.8. OMNI Interface Data Origin Authentication

AERO nodes employ simple data origin authentication procedures. In particular:

- o AERO Bridges and Proxy/Servers accept carrier packets received from the secured spanning tree.

- o AERO Proxy/Servers and Clients accept carrier packets and original IP packets that originate from within the same secured ANET.

- AERO Clients and Relays accept original IP packets from downstream network correspondents based on ingress filtering.

- AERO Clients, Relays, Proxy/Servers and Bridges verify carrier packet UDP/IP encapsulation addresses according to [I-D.templin-6man-omni].

- AERO nodes accept carrier packets addressed to themselves with Identification values within the current window for the OAL source neighbor (when window synchronization is used) and drop any carrier packets with out-of-window Identification values. (AERO nodes may forward carrier packets not addressed to themselves without verifying the Identification value.)

AERO nodes silently drop any packets that do not satisfy the above data origin authentication procedures. Further security considerations are discussed in Section 6.

3.9. OMNI Interface MTU

The OMNI interface observes the link nature of tunnels, including the Maximum Transmission Unit (MTU), Maximum Reassembly Unit (MRU) and the role of fragmentation and reassembly [I-D.ietf-intarea-tunnels]. The OMNI interface employs an OMNI Adaptation Layer (OAL) that accommodates multiple underlying links with diverse MTUs while observing both a minimum and per-path Maximum Payload Size (MPS). The functions of the OAL and the OMNI interface MTU/MRU/MPS are specified in [I-D.templin-6man-omni] with MTU/MRU both set to the constant value 9180 bytes, with minimum MPS set to 400 bytes, and with potentially larger per-path MPS values depending on the underlying path.

When the network layer presents an original IP packet to the OMNI interface, the OAL source encapsulates and fragments the original IP packet if necessary. When the network layer presents the OMNI interface with multiple original IP packets bound to the same OAL destination, the OAL source can concatenate them together into a single OAL super-packet as discussed in [I-D.templin-6man-omni]. The OAL source then fragments the OAL packet if necessary according to the minimum/path MPS such that the OAL headers appear in each fragment while the original IP packet header appears only in the first fragment. The OAL source then encapsulates each OAL fragment in *NET headers for transmission as carrier packets over an underlying interface connected to either a physical link (such as Ethernet, WiFi and the like) or a virtual link such as an Internet or higher-layer tunnel (see the definition of link in [RFC8200]).

Note: Although a CRH-32 may be inserted or removed by a Bridge in the path (see: Section 3.10.3), this does not interfere with the destination's ability to reassemble since the CRH-32 is not included in the fragmentable part and its removal/transformation does not invalidate fragment header information.

3.10. OMNI Interface Forwarding Algorithm

Original IP packets enter a node's OMNI interface either from the network layer (i.e., from a local application or the IP forwarding system) while carrier packets enter from the link layer (i.e., from an OMNI interface neighbor). All original IP packets and carrier packets entering a node's OMNI interface first undergo data origin authentication as discussed in Section 3.8. Those that satisfy data origin authentication are processed further, while all others are dropped silently.

Original IP packets that enter the OMNI interface from the network layer are forwarded to an OMNI interface neighbor using OAL encapsulation and fragmentation to produce carrier packets for transmission over underlying interfaces. (If routing indicates that the original IP packet should instead be forwarded back to the network layer, the packet is dropped to avoid looping). Carrier packets that enter the OMNI interface from the link layer are either re-encapsulated and re-admitted into the OMNI link, or reassembled and forwarded to the network layer where they are subject to either local delivery or IP forwarding. In all cases, the OAL MUST NOT decrement the original IP packet TTL/Hop-count since its forwarding actions occur below the network layer.

OMNI interfaces may have multiple underlying interfaces and/or neighbor cache entries for neighbors with multiple underlying interfaces (see Section 3.3). The OAL uses Interface Attributes and/or Traffic Selectors (e.g., port number, flow specification, etc.) to select an outbound underlying interface for each OAL packet and also to select segment routing and/or link-layer destination addresses based on the neighbor's underlying interfaces. AERO implementations SHOULD permit network management to dynamically adjust Traffic Selector values at runtime.

If an OAL packet matches the Traffic Selectors of multiple outgoing interfaces and/or neighbor interfaces, the OMNI interface replicates the packet and sends one copy via each of the (outgoing / neighbor) interface pairs; otherwise, it sends a single copy of the OAL packet via an interface with the best matching Traffic Selector. (While not strictly required, the likelihood of successful reassembly may improve when the OMNI interface sends all fragments of the same fragmented OAL packet consecutively over the same underlying

interface pair to avoid complicating factors such as delay variance and reordering.) AERO nodes keep track of which underlying interfaces are currently "reachable" or "unreachable", and only use "reachable" interfaces for forwarding purposes.

The following sections discuss the OMNI interface forwarding algorithms for Clients, Proxy/Servers and Bridges. In the following discussion, an original IP packet's destination address is said to "match" if it is the same as a cached address, or if it is covered by a cached prefix (which may be encoded in an MNP-LLA).

3.10.1. Client Forwarding Algorithm

When an original IP packet enters a Client's OMNI interface from the network layer the Client searches for a NCE that matches the destination. If there is a match, the Client selects one or more "reachable" neighbor interfaces in the entry for forwarding purposes. If there is no NCE, the Client instead either enqueues the original IP packet and invokes route optimization while forwarding the original IP packet toward an FHS Proxy/Server. The Client (acting as an OAL source) performs OAL encapsulation and sets the OAL destination address to the MNP-ULA of the target if there is a matching NCE; otherwise, it sets the OAL destination to the ADM-ULA of the FHS Proxy/Server. If the Client has multiple original IP packets to send to the same neighbor, it can concatenate them in a single super-packet [I-D.templin-6man-omni]. The OAL source then performs fragmentation to create OAL fragments (see: Section 3.9), appends any *NET encapsulation, and sends the resulting carrier packets over underlying interfaces to the neighbor acting as an OAL destination.

If the neighbor interface selected for forwarding is located on the same OMNI link segment and not behind a NAT, the Client forwards the carrier packets directly according to the L2ADDR information for the neighbor. If the neighbor interface is behind a NAT on the same OMNI link segment, the Client instead forwards the initial carrier packets to the LHS Proxy/Server (while inserting an CRH-32 if necessary) and initiates NAT traversal procedures. If the Client's intended source underlying interface is also behind a NAT and located on the same OMNI link segment, it sends a "direct bubble" over the interface per [RFC6081][RFC4380] to the L2ADDR found in the neighbor cache in order to establish state in its own NAT by generating traffic toward the neighbor (note that no response to the bubble is expected).

The Client next sends an NS(NUD) message toward the MNP-ULA of the neighbor via the LHS Proxy/Server as discussed in Section 3.15. If the Client receives an NA(NUD) from the neighbor over the underlying interface, it marks the neighbor interface as "trusted" and sends

future carrier packets directly to the L2ADDR information for the neighbor instead of indirectly via the LHS Proxy/Server. The Client must honor the neighbor cache maintenance procedure by sending additional direct bubbles and/or NS/NA(NUD) messages as discussed in [RFC6081][RFC4380] in order to keep NAT state alive as long as carrier packets are still flowing.

When a carrier packet enters a Client's OMNI interface from the link-layer, if the OAL destination matches one of the Client's ULAs the Client (acting as an OAL destination) verifies that the Identification is in-window for this OAL source, then reassembles and decapsulates as necessary and delivers the original IP packet to the network layer. If the OAL destination does not match, the Client drops the original IP packet and MAY return a network-layer ICMP Destination Unreachable message subject to rate limiting (see: Section 3.11).

Note: When a Client performs an NS/NA(WIN) exchange with a peer over the spanning tree, MFIB information will be established in all FHS and LHS intermediate nodes in the path. The Client can then opportunistically "skip ahead" in the chain of hops to bypass intermediate nodes that are not required to forward packets. The Client will have the map of MFVIs held at each hop, and can include the MFVI for the first hop it visits in a compressed OMNI header or in a CRH-32 routing header.

Note: Clients and their FHS Proxy/Server (and other Client) peers can exchange original IP packets over ANET underlying interfaces without invoking the OAL, since the ANET is secured at the link and physical layers. By forwarding original IP packets without invoking the OAL, however, the ANET peers can engage only in classical path MTU discovery since the packets are subject to loss and/or corruption due to the various per-link MTU limitations that may occur within the ANET. Moreover, the original IP packets do not include either the OAL integrity check or per-packet Identification values that can be used for data origin authentication and link-layer retransmissions. The tradeoff therefore involves an assessment of the per-packet encapsulation overhead saved by bypassing the OAL vs. inheritance of classical network "brittleness". (Note however that ANET peers can send small original IP packets without invoking the OAL, while invoking the OAL for larger packets. This presents the beneficial aspects of both small packet efficiency and large packet robustness, with delay variance and reordering as possible side effects.)

3.10.2. Proxy/Server and Relay Forwarding Algorithm

When the Proxy/Server receives an original IP packet from the network layer, it drops the packet if routing indicates that it should be forwarded back to the network layer to avoid looping. Otherwise, the Proxy/Server regards the original IP packet the same as if it had arrived as carrier packets with OAL destination set to its own ADM-ULA. When the Proxy/Server receives carrier packets on underlying interfaces with OAL destination set to its own ADM-ULA, it performs OAL reassembly if necessary to obtain the original IP packet.

The Proxy/Server next searches for a NCE that matches the original IP destination and proceeds as follows:

- if the packet is an NA(WIN) message for a local Client NCE, the Proxy/Server examines the Multilink Forwarding Parameters information and rewrites the fields if the NA(WIN) was not already processed by a (local segment) Bridge as discussed in Section 3.2.7.

- else, if the original IP packet destination matches a NCE, the Proxy/Sever uses one or more "reachable" neighbor interfaces in the entry for packet forwarding using OAL encapsulation and fragmentation according to the cached link-layer address information. If the neighbor interface is in a different OMNI link segment, the Proxy/Server performs OAL encapsulation and fragmentation, inserts an CRH-32 if necessary and forwards the resulting carrier packets via the spanning tree to a Bridge; otherwise, it forwards the carrier packets directly to the neighbor via INET encapsulation. If the neighbor is behind a NAT, this FHS Proxy/Server instead forwards initial carrier packets via a Bridge (or more directly via an LHS Proxy/Server) while sending an NS(NUD) to the neighbor. When the Proxy/Server receives the NA(NUD), it can begin forwarding carrier packets directly to the neighbor the same as discussed in Section 3.10.1 while sending additional NS(NUD) messages as necessary to maintain NAT state. Note that no direct bubbles are necessary since the Proxy/Server is by definition not located behind a NAT.

- else, if the original IP destination matches a non-MNP route in the IP forwarding table or an ADM-LLA assigned to the Proxy/Server's OMNI interface, the Proxy/Server acting as a Relay presents the original IP packet to the network layer for local delivery or IP forwarding.

- else, the Proxy/Server initiates address resolution as discussed in Section 3.14, while submitting the original IP packets for OAL

encapsulation and forwarding the resulting carrier packets into the secured spanning tree subject to rate limiting.

When the Proxy/Server receives a carrier packet with OAL destination set to an MNP-ULA that does not match the MSP, it accepts the carrier packet only if data origin authentication succeeds and if there is a network layer routing table entry for a GUA route that matches the MNP-ULA. If there is no route, the Proxy/Server drops the carrier packet; otherwise, it reassembles and decapsulates to obtain the original IP packet then acts as a Relay to present it to the network layer where it will be delivered according to standard IP forwarding.

When a Proxy/Server receives a carrier packet from one of its Client neighbors with OAL destination set to another node, it forwards the packets via a matching NCE or via the spanning tree if there is no matching entry. When the Proxy/Server receives a carrier packet with OAL destination set to the MNP-ULA of one of its Client neighbors established through RS/RA exchanges, it accepts the carrier packet only if data origin authentication succeeds. If the NCE state is DEPARTED, the Proxy/Server changes the OAL destination address to the ADM-ULA of the new Proxy/Server, then re-encapsulates the carrier packet and forwards it to a Bridge which will eventually deliver it to the new Proxy/Server.

If the neighbor cache state for the MNP-ULA is REACHABLE, the Proxy/Server forwards the carrier packets to the Client which then must reassemble. (Note that the Proxy/Server does not reassemble carrier packets not explicitly addressed to its own ADM-ULA, since some of the carrier packets of the same original IP packet could be forwarded through a different Proxy/Server.) In that case, the Client may receive fragments that are smaller than its link MTU but that can still be reassembled.

FHS Clients maintain a single Hub Proxy/Server and one or more FHS Proxy/Servers. FHS Clients can forward carrier packets to the FHS Proxy/Server for a specific outbound underlying interface while also initiating route optimization, and the FHS Proxy/Server will reassemble, re-encapsulate and re-fragment then send the resulting carrier packets into the secured spanning tree subject to rate limiting while route optimization is in progress. The secured spanning tree carrier packets will arrive at the Hub LHS Proxy/Server for an LHS Client, which reassembles, re-encapsulates, re-fragments and forwards to the LHS Client. Once route optimization has converged, the FHS and LHS Clients can enact "shortcuts" to avoid slow-path forwarding of carrier packets over the secured spanning tree.

Proxy/Servers process carrier packets with OAL destinations that do not match their ADM-ULA in the same manner as for traditional IP forwarding within the OAL, i.e., nodes use IP forwarding to forward packets not explicitly addressed to themselves. Proxy/Servers process carrier packets with their ADM-ULA as the destination by first examining the packet for a CRH-32 header or a compressed OAL header. In that case, the Proxy/Server examines the next MFVI in the carrier packet to locate the MFV entry in the MFIB for next hop forwarding (i.e., without examining IP addresses). When the Proxy/Server forwards the carrier packet, it changes the destination address according to the MFVI value for the next hop found either in the CRH-32 header or in the node's own MFIB.

Note: Proxy/Servers may receive carrier packets with CRH-32s that include additional forwarding information. Proxy/Servers use the forwarding information to determine the correct NCE and underlying interface for forwarding to the target Client, then remove the CRH-32 and forward the carrier packet. If necessary, the Proxy/Server reassembles first before re-encapsulating (and possibly also re-fragmenting) then forwards to the target Client. For a full discussion see: Section 3.14.6.

Note: Clients and their FHS Proxy/Server peers can exchange original IP packets over ANET underlying interfaces without invoking the OAL, since the ANET is secured at the link and physical layers. By forwarding original IP packets without invoking the OAL, however, the Client and Proxy/Server can engage only in classical path MTU discovery since the packets are subject to loss and/or corruption due to the various per-link MTU limitations that may occur within the ANET. Moreover, the original IP packets do not include either the OAL integrity check or per-packet Identification values that can be used for data origin authentication and link-layer retransmissions. The tradeoff therefore involves an assessment of the per-packet encapsulation overhead saved by bypassing the OAL vs. inheritance of classical network "brittleness". (Note however that ANET peers can send small original IP packets without invoking the OAL, while invoking the OAL for larger packets. This presents the beneficial aspects of both small packet efficiency and large packet robustness.)

Note: When a Proxy/Server receives a (non-OAL) original IP packet from an ANET Client, or a carrier packet with OAL destination set to its own ADM-ULA from any Client, the Proxy/Server reassembles if necessary then performs ROS functions on behalf of the Client. The Client may at some later time begin sending carrier packets to the OAL address of the actual target instead of the Proxy/Server, at which point it may begin functioning as an ROS on its own behalf and thereby "override" the Proxy/Server's ROS role.

Note; Proxy/Servers drop any original IP packets (received either directly from an ANET Client or following reassembly of carrier packets received from an ANET/INET Client) with a destination that corresponds to the Client's delegated MNP. Similarly, Proxy/Servers drop any carrier packet received with both a source and destination that correspond to the Client's delegated MNP regardless of their OMNI link point of origin. These checks are necessary to prevent Clients from either accidentally or intentionally establishing endless loops that could congest Proxy/Servers and/or ANET/INET links.

Note: Proxy/Servers forward secure control plane carrier packets via the SRT secured spanning tree and forward other carrier packets via the unsecured spanning tree. When a Proxy/Server receives a carrier packet from the secured spanning tree, it considers the message as authentic without having to verify upper layer authentication signatures. When a Proxy/Server receives a carrier packet from the unsecured spanning tree, it verifies any upper layer authentication signatures and/or forwards the unsecured message toward the destination which must apply data origin authentication.

Note: If the Proxy/Server has multiple original IP packets to send to the same neighbor, it can concatenate them in a single OAL super-packet [I-D.templin-6man-omni].

3.10.3. Bridge Forwarding Algorithm

Bridges forward spanning tree carrier packets while decrementing the OAL header Hop Count but not the original IP header Hop Count/TTL. Bridges convey carrier packets that encapsulate critical IPv6 ND control messages or routing protocol control messages via the secured spanning tree, and may convey other carrier packets via the unsecured spanning tree or via more direct paths according to MFIB information. When the Bridge receives a carrier packet, it removes the outer *NET header and searches for an MFIB entry that matches an MFVI or an IP forwarding table entry that matches the OAL destination address.

Bridges process carrier packets with OAL destinations that do not match their ADM-ULA or the SRT Subnet Router Anycast address in the same manner as for traditional IP forwarding within the OAL, i.e., nodes use IP forwarding to forward packets not explicitly addressed to themselves. Bridges process carrier packets with their ADM-ULA or the SRT Subnet Router Anycast address as the destination by first examining the packet for a CRH-32 header or a compressed OAL header. In that case, the Bridge examines the next MFVI in the carrier packet to locate the MFV entry in the MFIB for next hop forwarding (i.e., without examining IP addresses). When the Bridge forwards the carrier packet, it changes the destination address according to the

MFVI value for the next hop found either in the CRH-32 header or in the node's own MFIB.

Bridges forward carrier packets received from a first segment via the SRT secured spanning tree to the next segment also via the secured spanning tree. Bridges forward carrier packets received from a first segment via the unsecured spanning tree to the next segment also via the unsecured spanning tree. Bridges use a single IPv6 routing table that always determines the same next hop for a given OAL destination, where the secured/unsecured spanning tree is determined through the selection of the underlying interface to be used for transmission (i.e., a secured tunnel or an open INET interface).

3.11. OMNI Interface Error Handling

When an AERO node admits an original IP packet into the OMNI interface, it may receive link-layer or network-layer error indications. The AERO node may also receive OMNI link error indications in OAL-encapsulated uNA messages that include authentication signatures.

A link-layer error indication is an ICMP error message generated by a router in the INET on the path to the neighbor or by the neighbor itself. The message includes an IP header with the address of the node that generated the error as the source address and with the link-layer address of the AERO node as the destination address.

The IP header is followed by an ICMP header that includes an error Type, Code and Checksum. Valid type values include "Destination Unreachable", "Time Exceeded" and "Parameter Problem" [RFC0792][RFC4443]. (OMNI interfaces ignore link-layer IPv4 "Fragmentation Needed" and IPv6 "Packet Too Big" messages for carrier packets that are no larger than the minimum/path MPS as discussed in Section 3.9, however these messages may provide useful hints of probe failures during path MPS probing.)

The ICMP header is followed by the leading portion of the carrier packet that generated the error, also known as the "packet-in-error". For ICMPv6, [RFC4443] specifies that the packet-in-error includes: "As much of invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU" (i.e., no more than 1280 bytes). For ICMPv4, [RFC0792] specifies that the packet-in-error includes: "Internet Header + 64 bits of Original Data Datagram", however [RFC1812] Section 4.3.2.3 updates this specification by stating: "the ICMP datagram SHOULD contain as much of the original datagram as possible without the length of the ICMP datagram exceeding 576 bytes".

The link-layer error message format is shown in Figure 4:

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     ~                               ~
     |    IP Header of link layer    |
     |         error message         |
     ~                               ~
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |          ICMP Header          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
     ~                               ~   P
     |  carrier packet *NET and OAL  |   a
     |     encapsulation headers     |   c
     ~                               ~   k
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   e
     ~                               ~   t
     |  original IP packet headers   |
     |    (first-fragment only)      |   i
     ~                               ~   n
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     ~                               ~   e
     |    Portion of the body of     |   r
     |    the original IP packet     |   r
     |       (all fragments)         |   o
     ~                               ~   r
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
```

Figure 4: OMNI Interface Link-Layer Error Message Format

The AERO node rules for processing these link-layer error messages are as follows:

- o When an AERO node receives a link-layer Parameter Problem message, it processes the message the same as described as for ordinary ICMP errors in the normative references [RFC0792][RFC4443].

- o When an AERO node receives persistent link-layer Time Exceeded messages, the IP ID field may be wrapping before earlier fragments awaiting reassembly have been processed. In that case, the node should begin including integrity checks and/or institute rate limits for subsequent packets.

- o When an AERO node receives persistent link-layer Destination Unreachable messages in response to carrier packets that it sends to one of its neighbor correspondents, the node should process the message as an indication that a path may be failing, and optionally initiate NUD over that path. If it receives Destination Unreachable messages over multiple paths, the node

should allow future carrier packets destined to the correspondent to flow through a default route and re-initiate route optimization.

- When an AERO Client receives persistent link-layer Destination Unreachable messages in response to carrier packets that it sends to one of its neighbor Proxy/Servers, the Client should mark the path as unusable and use another path. If it receives Destination Unreachable messages on many or all paths, the Client should associate with a new Proxy/Server and release its association with the old Proxy/Server as specified in Section 3.16.5.

- When an AERO Proxy/Server receives persistent link-layer Destination Unreachable messages in response to carrier packets that it sends to one of its neighbor Clients, the Proxy/Server should mark the underlying path as unusable and use another underlying path.

- When an AERO Proxy/Server receives link-layer Destination Unreachable messages in response to a carrier packet that it sends to one of its permanent neighbors, it treats the messages as an indication that the path to the neighbor may be failing. However, the dynamic routing protocol should soon reconverge and correct the temporary outage.

When an AERO Bridge receives a carrier packet for which the network-layer destination address is covered by an MSP, the Bridge drops the packet if there is no more-specific routing information for the destination and returns an OMNI interface Destination Unreachable message subject to rate limiting.

When an AERO node receives a carrier packet for which reassembly is currently congested, it returns an OMNI interface Packet Too Big (PTB) message as discussed in [I-D.templin-6man-omni] (note that the PTB messages could indicate either "hard" or "soft" errors).

AERO nodes include ICMPv6 error messages intended for the OAL source as sub-options in the OMNI option of secured uNA messages. When the OAL source receives the uNA message, it can extract the ICMPv6 error message enclosed in the OMNI option and either process it locally or translate it into a network-layer error to return to the original source.

3.12. AERO Router Discovery, Prefix Delegation and Autoconfiguration

AERO Router Discovery, Prefix Delegation and Autoconfiguration are coordinated as discussed in the following Sections.

3.12.1. AERO Service Model

Each AERO Proxy/Server on the OMNI link is configured to facilitate Client prefix delegation/registration requests. Each Proxy/Server is provisioned with a database of MNP-to-Client ID mappings for all Clients enrolled in the AERO service, as well as any information necessary to authenticate each Client. The Client database is maintained by a central administrative authority for the OMNI link and securely distributed to all Proxy/Servers, e.g., via the Lightweight Directory Access Protocol (LDAP) [RFC4511], via static configuration, etc. Clients receive the same service regardless of the Proxy/Servers they select.

Clients associate each of their underlying interfaces with a FHS Proxy/Server. Each FHS Proxy/Server may locally service one or more of the Client's underlying interfaces, and the Client selects one among them to serve as the Hub Proxy/Server. The Hub Proxy/Server is responsible for short-term packet forwarding, for acting as a mobility anchor point and for acting as an ROR for NS(AR) messages directed to the Client. All of the Client's other FHS Proxy/Servers forward proxyed copies of RS/RA messages between the Hub Proxy/Server and Client without assuming the Hub role functions themselves.

Each Client associates with a single Hub Proxy/Server at a time, while all FHS Proxy/Servers are candidates for providing the Hub role for other Clients. An FHS Proxy/Server assumes the Hub role when it receives an RS message with its own ADM-LLA or All-Routers multicast as the destination. An FHS Proxy/Server assumes the proxy role when it receives an RS message with the ADM-LLA of another Proxy/Server as the destination.

AERO Clients and Proxy/Servers use IPv6 ND messages to maintain neighbor cache entries. AERO Proxy/Servers configure their OMNI interfaces as advertising NBMA interfaces, and therefore send unicast RA messages with a short Router Lifetime value (e.g., ReachableTime seconds) in response to a Client's RS message. Thereafter, Clients send additional RS messages to keep Proxy/Server state alive.

AERO Clients and Hub Proxy/Servers include prefix delegation and/or registration parameters in RS/RA messages (see [I-D.templin-6man-omni]). The IPv6 ND messages are exchanged between the Client and Hub Proxy/Server (via any FHS Proxy/Servers acting as proxies) according to the prefix management schedule required by the service. If the Client knows its MNP in advance, it can employ prefix registration by including its MNP-LLA as the source address of an RS message and with an OMNI option with valid prefix registration information for the MNP. If the Hub Proxy/Server accepts the Client's MNP assertion, it injects the MNP into the routing system

and establishes the necessary neighbor cache state. If the Client does not have a pre-assigned MNP, it can instead employ prefix delegation by including the unspecified address (::) as the source address of an RS message and with an OMNI option with prefix delegation parameters to request an MNP.

The following sections specify the Client and Proxy/Server behavior.

3.12.2. AERO Client Behavior

AERO Clients discover the addresses of candidate FHS Proxy/Servers by resolving the Potential Router List (PRL) in a similar manner as described in [RFC5214]. Discovery methods include static configuration (e.g., a flat-file map of Proxy/Server addresses and locations), or through an automated means such as Domain Name System (DNS) name resolution [RFC1035]. Alternatively, the Client can discover Proxy/Server addresses through a layer 2 data link login exchange, or through a unicast RA response to a multicast/anycast RS as described below. In the absence of other information, the Client can resolve the DNS Fully-Qualified Domain Name (FQDN) "linkupnetworks.[domainname]" where "linkupnetworks" is a constant text string and "[domainname]" is a DNS suffix for the OMNI link (e.g., "example.com").

To associate with a Hub Proxy/Server over a first underlying interface, the Client acts as a requesting router to request MNPs by preparing an RS message with prefix management parameters. If the Client already knows the Proxy/Server's ADM-LLA, it includes the LLA as the network-layer destination address; otherwise, the Client includes the (link-local) All-Routers multicast as the network-layer destination. The Client can use its MNP-LLA as the network-layer source address and include an OMNI option with prefix registration information. If the Client does not yet have an MNP-LLA, it instead sets the network-layer source address to unspecified (::) and includes prefix delegation parameters in the OMNI option (see: [I-D.templin-6man-omni]).

The Client next includes an authentication sub-option if necessary and Multilink Forwarding Parameters corresponding to the underlying interface over which it will send the RS message. Next, the Client submits the RS for OAL encapsulation and fragmentation if necessary with its own MNP-ULA and the Proxy/Server's ADM-ULA or an OMNI IPv6 anycast address as the OAL addresses while selecting an Identification value and invoking window synchronization as specified in [I-D.templin-6man-omni].

The Client then sends the RS (either directly via Direct interfaces, via a VPN for VPNed interfaces, via an access router for ANET

interfaces or via INET encapsulation for INET interfaces) then waits up to RetransTimer milliseconds for an RA message reply (see Section 3.12.3) (retrying up to MAX_RTR_SOLICITATIONS). If the Client receives no RAs, or if it receives an RA with Router Lifetime set to 0, the Client SHOULD abandon attempts through the first candidate Hub Proxy/Server and try another FHS Proxy/Server. Otherwise, the Client processes the prefix information found in the RA message.

When the Client processes an RA, it first performs OAL reassembly and decapsulation if necessary then creates a NCE with the Hub Proxy/Server's ADM-LLA as the network-layer address and the Hub Proxy/Server's encapsulation and/or link-layer addresses as the link-layer address. The Client then caches the Multilink Forwarding Parameters information. The Client next records the RA Router Lifetime field value in the NCE as the time for which the Hub Proxy/Server has committed to maintaining the MNP in the routing system via this underlying interface, and caches the other RA configuration information including Cur Hop Limit, M and O flags, Reachable Time and Retrans Timer. The Client then autoconfigures MNP-LLAs for any delegated MNPs and assigns them to the OMNI interface. The Client also caches any MSPs included in Route Information Options (RIOs) [RFC4191] as MSPs to associate with the OMNI link, and assigns the MTU value in the MTU option to the underlying interface.

The Client then registers its additional underlying interfaces with FHS Proxy/Servers for those interfaces discovered by sending RS messages via each additional interface but with the ADM-LLA of the Hub Proxy/Server as the destination. The additional FHS Proxy/Servers will assume the proxy role and forward proxyed copies of the RS/RA exchanges between the Client and Hub Proxy/Server. The Client finally sub-delegates the MNPs to its attached EUNs and/or the Client's own internal virtual interfaces as described in [I-D.templin-v6ops-pdhost] to support the Client's downstream attached "Internet of Things (IoT)". The Client then sends additional RS messages over each underlying interface before the Router Lifetime received for that interface expires.

After the Client registers its underlying interfaces, it may wish to change one or more registrations, e.g., if an interface changes address or becomes unavailable, if traffic selectors change, etc. To do so, the Client prepares an RS message to send over any available underlying interface as above. The RS includes an OMNI option with prefix registration/delegation information and with Multilink Forwarding Parameters specific to the selected underlying interface. When the Client receives the Hub Proxy/Server's RA response, it has assurance that the Proxy/Server has been updated with the new information.

If the Client wishes to discontinue use of a Hub Proxy/Server it issues an RS message over any underlying interface with an OMNI option with a prefix release indication. When the Hub Proxy/Server processes the message, it releases the MNP, sets the NCE state for the Client to DEPARTED and returns an RA reply with Router Lifetime set to 0. After a short delay (e.g., 2 seconds), the Hub Proxy/ Server withdraws the MNP from the routing system.

3.12.3. AERO Proxy/Server Behavior

AERO Proxy/Servers act as both IP routers and IPv6 ND proxies, and support a prefix delegation/registration service for Clients. Proxy/ Servers arrange to add their ADM-LLAs to the PRL maintained in a static map of Proxy/Server addresses for the link, the DNS resource records for the FQDN "linkupnetworks.[domainname]", etc. before entering service. The PRL should be arranged such that Clients can discover the addresses of Proxy/Servers that are geographically and/ or topologically "close" to their underlying network connections.

When an FHS Proxy/Server receives a prospective Client's RS message, it SHOULD return an immediate RA reply with Router Lifetime set to 0 if it is currently too busy or otherwise unable to service the Client. Otherwise, the Proxy/Server performs OAL reassembly if necessary, then decapsulates and authenticates the RS message. If the RS message destination is All-Routers multicast or the Proxy/ Server's own ADM-LLA, the Proxy/Server assumes the Hub role. If the RS message destination is the ADM-LLA of another node, the Proxy/ Server assumes the proxy role and forwards the RS to the Hub Proxy/ server via the secured spanning tree.

The Hub Proxy/Server then determines the correct MNPs to provide to the Client by processing the MNP-LLA prefix parameters and/or the DHCPv6 OMNI sub-option. When the Hub Proxy/Server returns the MNPs, it also creates a forwarding table entry for the MNP-ULA corresponding to each MNP resulting in a BGP update (see: Section 3.2.3). For IPv6, the Hub Proxy/Server creates an IPv6 forwarding table entry for each MNP. For IPv4, the Hub Proxy/Server creates an IPv6 forwarding table entry with the IPv4-compatibility MNP-ULA prefix corresponding to the IPv4 address.

The Hub Proxy/Server next creates a NCE for the Client using the base MNP-LLA as the network-layer address. Next, the Hub Proxy/Server updates the NCE by recording the information in the Multilink Forwarding Parameters sub-option in the RS OMNI option. The Hub Proxy/Server also records the actual OAL/*NET addresses and RS message window synchronization parameters (if any) in the NCE.

Next, the Hub Proxy/Server prepares an RA message using its ADM-LLA as the network-layer source address and the network-layer source address of the RS message as the network-layer destination address. The Hub Proxy/Server sets the Router Lifetime to the time for which it will maintain both this underlying interface individually and the NCE as a whole. The Hub Proxy/Server also sets Cur Hop Limit, M and O flags, Reachable Time and Retrans Timer to values appropriate for the OMNI link. The Hub Proxy/Server includes the MNPs, any other prefix management parameters and an OMNI option with a Multilink Forwarding Parameters sub-option with FHS addressing information filled out. The Hub Proxy/Server then includes one or more RIOs that encode the MSPs for the OMNI link, plus an MTU option (see Section 3.9). The Hub Proxy/Server finally forwards the message to the Client using OAL encapsulation/fragmentation if necessary while including an acknowledgement if the RS invoked window synchronization.

After the initial RS/RA exchange, the Hub Proxy/Server maintains a ReachableTime timer for each of the Client's underlying interfaces individually (and for the Client's NCE collectively) set to expire after ReachableTime seconds. If the Client (or an FHS Proxy/Server) issues additional RS messages, the Hub Proxy/Server sends an RA response and resets ReachableTime. If the Hub Proxy/Server receives an IPv6 ND message with a prefix release indication it sets the Client's NCE to the DEPARTED state and withdraws the MNP from the routing system after a short delay (e.g., 2 seconds). If ReachableTime expires before a new RS is received on an individual underlying interface, the Hub Proxy/Server marks the interface as DOWN. If ReachableTime expires before any new RS is received on any individual underlying interface, the Hub Proxy/Server sets the NCE state to STALE and sets a 10 second timer. If the Hub Proxy/Server has not received a new RS or uNA message with a prefix release indication before the 10 second timer expires, it deletes the NCE and withdraws the MNP from the routing system.

The Hub Proxy/Server processes any IPv6 ND messages pertaining to the Client and returns an NA/RA reply in response to solicitations. The Hub Proxy/Server may also issue unsolicited RA messages, e.g., with reconfigure parameters to cause the Client to renegotiate its prefix delegation/registrations, with Router Lifetime set to 0 if it can no longer service this Client, etc. Finally, If the NCE is in the DEPARTED state, the Hub Proxy/Server deletes the entry after DepartTime expires.

The Hub Proxy/Server may also receive carrier packets via the secured spanning tree that contain initial data packets sent while route optimization is in progress. The Hub Proxy/Server reassembles, then re-encapsulates/re-fragments and forwards the packets to the target

Client. Although these fragments will have traversed the secured spanning tree, the security only assures correct reassembly and does not assure message content security.

Note: Clients SHOULD notify former Hub Proxy/Servers of their departures, but Hub Proxy/Servers are responsible for expiring neighbor cache entries and withdrawing routes even if no departure notification is received (e.g., if the Client leaves the network unexpectedly). Hub Proxy/Servers SHOULD therefore set Router Lifetime to ReachableTime seconds in solicited RA messages to minimize persistent stale cache information in the absence of Client departure notifications. A short Router Lifetime also ensures that proactive RS/RA messaging between Clients and FHS Proxy/Servers will keep any NAT state alive (see above).

Note: All Proxy/Servers on an OMNI link MUST advertise consistent values in the RA Cur Hop Limit, M and O flags, Reachable Time and Retrans Timer fields the same as for any link, since unpredictable behavior could result if different Proxy/Servers on the same link advertised different values.

3.12.3.1. DHCPv6-Based Prefix Registration

When a Client is not pre-provisioned with an MNP-LLA, it will need for the Hub Proxy/Server to select one or more MNPs on its behalf and set up the correct state in the AERO routing service. (A Client with a pre-provisioned MNP may also request the Hub Proxy/Server to select additional MNPs.) The DHCPv6 service [RFC8415] is used to support this requirement.

When a Client needs to have the Hub Proxy/Server select MNPs, it sends an RS message with source address set to the unspecified address (::) and with an OMNI option that includes a DHCPv6 message sub-option with DHCPv6 Prefix Delegation (DHCPv6-PD) parameters. When the Hub Proxy/Server receives the RS message, it extracts the DHCPv6-PD message from the OMNI option.

The Hub Proxy/Server then acts as a "Proxy DHCPv6 Client" in a message exchange with the locally-resident DHCPv6 server, which delegates MNPs and returns a DHCPv6-PD Reply message. (If the Hub Proxy/Server wishes to defer creation of MN state until the DHCPv6-PD Reply is received, it can instead act as a Lightweight DHCPv6 Relay Agent per [RFC6221] by encapsulating the DHCPv6-PD message in a Relay-forward/reply exchange with Relay Message and Interface ID options.)

When the Hub Proxy/Server receives the DHCPv6-PD Reply, it adds a route to the routing system and creates an MNP-LLA based on the

delegated MNP. The Hub Proxy/Server then sends an RA back to the Client with the (newly-created) MNP-LLA as the destination address and with the DHCPv6-PD Reply message coded in the OMNI option. When the Client receives the RA, it creates a default route, assigns the Subnet Router Anycast address and sets its MNP-LLA based on the delegated MNP.

Note: See [I-D.templin-6man-omni] for an MNP delegation alternative that avoids including a DHCPv6 message sub-option in the RS. Namely, when the Client requests a single MNP it can set the RS source to the unspecified address (::) and include a Node Identification sub-option and Preflen in the OMNI option (but with no DHCPv6 message sub-option). When the Hub Proxy/Server receives the RS message, it forwards a self-generated DHCPv6 Solicit message to the DHCPv6 server on behalf of the Client. When the Hub Proxy/Server receives the DHCPv6 Reply, it prepares an RA message with an OMNI option with Preflen information (but with no DHCPv6 message sub-option), then places the (newly-created) MNP-LLA in the RA destination address and returns the message to the Client.

3.13. AERO Proxy/Server Coordination

OMNI link Clients register with FHS Proxy/Servers for each underlying interface. Each of the Client's FHS Proxy/Servers must inform a single Hub Proxy/Server of all of the Client's additional underlying interfaces. For Clients on Direct and VPNed underlying interfaces, the FHS Proxy/Server for each interface is directly connected, for Clients on ANET underlying interfaces the FHS Proxy/Server is located on the ANET/INET boundary, and for Clients on INET underlying interfaces the FHS Proxy/Server is located somewhere in the connected Internetwork. When FHS Proxy/Server "A" processes a Client registration, it must either assume the Hub role or forward a proxyed registration to another FHS Proxy/Server acting as the Hub. Proxy/Servers satisfy these requirements as follows:

o when Proxy/Server "A" receives a Client RS message, it first verifies that the OAL Identification is within the window for the NCE that matches the MNP-ULA for this Client neighbor and authenticates the message. (If no NCE was found, Proxy/Server "A instead creates one in the STALE state and returns an RA message with an authentication signature if necessary and any window synchronization parameters.) Proxy/Server "A" then examines the network-layer destination address. If the destination address is the ADM-LLA of a different Proxy/Server "B", Proxy/Server "A" prepares a separate proxyed version of the RS message with an OAL header with source set to its own ADM-ULA and destination set to Proxy/Server B's ADM-ULA. Proxy/Server "A" also writes its own information over the Multilink Forwarding Parameters sub-option

supplied by the Client then sets the S/T-omIndex to the value for this Client underlying interface, then forwards the message into the OMNI link secured spanning tree.

- when Proxy/Server "B" receives the RS, it assume the Hub role and creates or updates a NCE for the Client with FHS Proxy/Server "A"'s Multilink Forwarding Parameters as the link-layer address information for this S/T-omIndex and caches any window synchronization parameters supplied by the Client. Hub Proxy/Server "B" then prepares an RA message with source set to its own LLA and destination set to the Client's MNP-LLA, and with any window synchronization acknowledgements. Hub Proxy/Server "B" then encapsulates the RA in an OAL header with source set to its own ADM-ULA and destination set to the ADM-ULA of FHS Proxy/Server "A, performs fragmentation if necessary, then sends the resulting carrier packets into the secured spanning tree.

- when Proxy/Server "A" reassembles the RA, it locates the Client NCE based on the RA destination LLA. Proxy/Server "A" then re-encapsulates the RA message with OAL source set to its own ADM-ULA and OAL destination set to the MNP-ULA of the Client, includes an authentication signature if necessary, and echoes the Multilink Forwarding Parameters sub-option. Proxy/Server "A" then fragments if necessary and returns the fragments to the Client.

- The Client repeats this process over each of its additional underlying interfaces while treating each FHS Proxy/Server "C", "D", "E", etc. as a proxy to facilitate RS/RA exchanges between the Hub and the Client.

After the initial RS/RA exchanges each FHS Proxy/Server forwards any of the Client's carrier packets with OAL destinations for which there is no matching NCE to a Bridge using OAL encapsulation with its own ADM-ULA as the source and with destination determined by the Client. The Proxy/Server instead forwards any carrier packets destined to a neighbor cache target directly to the target according to the OAL/link-layer information - the process of establishing neighbor cache entries is specified in Section 3.14.

While the Client is still associated with each FHS Proxy/Server "A", "A" can send NS, RS and/or unsolicited NA messages to update the neighbor cache entries of other AERO nodes on behalf of the Client and/or to convey Multilink Forwarding Parameter updates. This allows for higher-frequency Proxy-initiated RS/RA messaging over well-connected INET infrastructure supplemented by lower-frequency Client-initiated RS/RA messaging over constrained ANET data links.

If the Hub Proxy/Server "A" ceases to send solicited RAs, Proxy/ Servers "B", "C", "D" send unsolicited RAs over the Client's underlying interface with destination set to (link-local) All-Nodes multicast and with Router Lifetime set to zero to inform Clients that the Hub Proxy/Server has failed. Although Proxy/Servers "B", "C" and "D" can engage in IPv6 ND exchanges on behalf of the Client, the Client can also send IPv6 ND messages on its own behalf, e.g., if it is in a better position to convey state changes. The IPv6 ND messages sent by the Client include the Client's MNP-LLA as the source in order to differentiate them from the IPv6 ND messages sent by Proxy/Server "A".

If the Client becomes unreachable over all underlying interface it serves, the Hub Proxy/Server sets the NCE state to DEPARTED and retains the entry for DepartTime seconds. While the state is DEPARTED, the Hub Proxy/Server forwards any carrier packets destined to the Client to a Bridge via OAL encapsulation. When DepartTime expires, the Hub Proxy/Server deletes the NCE and discards any further carrier packets destined to the former Client.

In some ANETs that employ a Proxy/Server, the Client's MNP can be injected into the ANET routing system. In that case, the Client can send original IP packets without invoking the OAL so that the ANET routing system transports the original IP packets to the Proxy. This can be very beneficial, e.g., if the Client connects to the ANET via low-end data links such as some aviation wireless links.

If the ANET first-hop access router is on the same underlying link as the Client and recognizes the AERO/OMNI protocol, the Client can avoid OAL encapsulation for both its control and data messages. When the Client connects to the link, it can send an unencapsulated RS message with source address set to its own MNP-LLA (or to a Temporary LLA), and with destination address set to the ADM-LLA of the Client's selected Proxy/Server or to (link-local) All-Routers multicast. The Client includes an OMNI option formatted as specified in [I-D.templin-6man-omni]. The Client then sends the unencapsulated RS message, which will be intercepted by the AERO-Aware access router.

The ANET access router then performs OAL encapsulation on the RS message and forwards it to a Proxy/Server at the ANET/INET boundary. When the access router and Proxy/Server are one and the same node, the Proxy/Server would share and underlying link with the Client but its message exchanges with outside correspondents would need to pass through a security gateway at the ANET/INET border. The method for deploying access routers and Proxys (i.e. as a single node or multiple nodes) is an ANET-local administrative consideration.

Note: When a Proxy/Server alters the IPv6 ND message contents before forwarding (e.g., such as altering the OMNI option contents), the IPv6 ND message checksum and/or authentication signature are invalidated. If the Proxy/Server forwards the message over the secured spanning tree, however, it need not re-calculate the checksum/signature since they will not be examined by the next hop.

Note: When a Proxy/Server receives a secured Client NS message, it performs the same proxying procedures as for described for RS messages above. The proxying procedures for NS/NA message exchanges is specified in Section 3.14.

3.13.1. Detecting and Responding to Proxy/Server Failures

In environments where fast recovery from Proxy/Server failure is required, FHS Proxy/Servers SHOULD use proactive Neighbor Unreachability Detection (NUD) to track Hub Proxy/Server reachability in a similar fashion as for Bidirectional Forwarding Detection (BFD) [RFC5880]. Each FHS Proxy/Server can then quickly detect and react to failures so that cached information is re-established through alternate paths. The NS/NA(NUD) control messaging is carried only over well-connected ground domain networks (i.e., and not low-end aeronautical radio links) and can therefore be tuned for rapid response.

FHS Proxy/Servers perform continuous NS/NA(NUD) exchanges with the Hub Proxy/Server in rapid succession, e.g., one exchange per second. The FHS Proxy/Server sends the NS(NUD) message via the spanning tree with its own ADM-LLA as the source and the ADM-LLA of the Hub Proxy/Server as the destination, and the Hub Proxy/Server responds with an NA(NUD). When the FHS Proxy/Server is also sending RS messages to a Hub Proxy/Server on behalf of Clients, the resulting RA responses can be considered as equivalent hints of forward progress. This means that the FHS Proxy/Server need not also send a periodic NS(NUD) if it has already sent an RS within the same period. If the Hub Proxy/Server fails (i.e., if the FHS Proxy/Server ceases to receive advertisements), the FHS Proxy/Server can quickly inform Clients by sending unsolicited RA messages

The FHS Proxy/Server sends unsolicited RA messages with source address set to the Hub Proxy/Server's address, destination address set to (link-local) All-Nodes multicast, and Router Lifetime set to 0. The FHS Proxy/Server SHOULD send MAX_FINAL_RTR_ADVERTISEMENTS RA messages separated by small delays [RFC4861]. Any Clients that had been using the failed Hub Proxy/Server will receive the RA messages and select one of its other FHS Proxy/Servers to assume the Hub role (i.e., by sending an RS with destination set to the ADM-LLA of the new Hub).

3.14. AERO Route Optimization

AERO nodes invoke route optimization when they need to forward packets to new target destinations. Route optimization is based on IPv6 ND Address Resolution messaging between a Route Optimization Source (ROS) and the target Client's current Hub Proxy/Server acting as a Route Optimization Responder (ROR). Route optimization is initiated by the first eligible ROS closest to the source as follows:

- For Clients on VPNed and Direct interfaces, the Client's FHS Proxy/Server is the ROS.

- For Clients on ANET interfaces, either the Client or the FHS Proxy/Server may be the ROS.

- For Clients on INET interfaces, the Client itself is the ROS.

- For correspondent nodes on INET/EUN interfaces serviced by a Relay, the Relay is the ROS.

The route optimization procedure is conducted between the ROS and the LHS Hub Proxy/Server/Relay for the target selected by routing as the ROR. In this arrangement, the ROS is always the Client or Proxy/Server (or Relay) nearest the source over the selected source underlying interface, while the ROR is always the target's current Hub Proxy/Server.

The AERO routing system directs a route optimization request sent by the ROS to the ROR, which returns a route optimization reply which must include information that is current, consistent and authentic. The ROS is responsible for periodically refreshing the route optimization, and the ROR is responsible for quickly informing the ROS of any changes.

The procedures are specified in the following sections.

3.14.1. Route Optimization Initiation

When an original IP packet from a source node destined to a target node arrives, the ROS checks for a NCE with an MNP-LLA that matches the target destination. If there is a NCE in the REACHABLE state, the ROS invokes the OAL and forwards the resulting carrier packets according to the cached state then returns from processing. Otherwise, if there is no NCE the ROS creates one in the INCOMPLETE state.

The ROS next invokes the OAL and forwards the resulting carrier packets into the secured spanning tree, then sends an NS message for

Address Resolution (NS(AR)) to receive a solicited NA(AR) message from the ROR. While route optimization is in progress, the ROS may forward additional original IP packets into the secured spanning tree but if so must impose rate limiting to minimize secured spanning tree traffic as well as ROR reassembly.

The NS(AR) message must be sent securely, and includes:

- the LLA of the ROS as the source address.
- the MNP-LLA corresponding to the original IP packet's destination as the Target Address, e.g., for 2001:db8:1:2::10:2000 the Target Address is fe80::2001:db8:1:2.
- the Solicited-Node multicast address [RFC4291] formed from the lower 24 bits of the original IP packet's destination as the destination address, e.g., for 2001:db8:1:2::10:2000 the NS(AR) destination address is ff02:0:0:0:0:1:ff10:2000.

The NS(AR) message also includes an OMNI option with an authentication sub-option if necessary and with Preflen set to the prefix length associated with the NS(AR) source. The ROS then selects an Identification value and submits the NS(AR) message for OAL encapsulation with OAL source set to its own ULA and OAL destination set to the ULA corresponding to the target. (The ROS does not include any window synchronization parameters, since it will not exchange other packet types with the ROR.) The ROS then sends the resulting carrier packet into the SRT secured spanning tree without decrementing the network-layer TTL/Hop Limit field.

When the ROS is an INET Client, it must instead forward the resulting carrier packet to the ADM-ULA of one of its current Proxy/Servers. The Proxy/Server then verifies the NS(AR) authentication signature, then re-encapsulates with the OAL source set to its own ADM-ULA and OAL destination set to the ULA corresponding to the target and forwards the resulting carrier packets into the secured spanning tree on behalf of the Client.

3.14.2. Relaying the NS(AR) *NET Packet(s)

When the Bridge receives the carrier packet containing the RS from the ROS, it discards the *NET headers and determines the next hop by consulting its standard IPv6 forwarding table for the OAL header destination address. The Bridge then decrements the OAL header Hop-Limit, then re-encapsulates and forwards the carrier packet(s) via the secured spanning tree the same as for any IPv6 router, where it may traverse multiple OMNI link segments. The final-hop Bridge will

deliver the carrier packet via the secured spanning tree to the ROR for the target.

3.14.3. Processing the NS(AR) and Sending the NA(AR)

When the ROR for the target receives the secured carrier packet, it examines the NS(AR) target to determine whether it has a matching NCE and/or non-MNP route. If there is no match, the ROR drops the message. Otherwise, the ROR continues processing as follows:

- if the NS(AR) target matches a Client NCE in the DEPARTED state, the ROR re-encapsulates while setting the OAL source to the ULA of the ROS and OAL destination address to the ADM-ULA of the Client's new Proxy/Server. The ROR then forwards the resulting carrier packet over the secured spanning tree then returns from processing.

- If the NS(AR) target matches the MNP-LLA of a Client NCE in the REACHABLE state, the ROR notes whether the NS (AR) arrived from the secured spanning tree then provides route optimization information on behalf of the Client. If the message arrived via the secured spanning tree the ROR need not perform further authentication; otherwise, it must verify the message authentication signature before accepting.

- If the NS(AR) target matches one of its non-MNP routes, the ROR serves as both a Relay and a route optimization target, since the Relay forwards IP packets toward the (fixed network) target at the network layer.

The ROR next checks the target NCE for a Report List entry that matches the NS(AR) source LLA/ULA of the ROS. If there is a Report List entry, the ROR refreshes ReportTime for this ROR; otherwise, the ROR creates a new entry for the ROS and records both the LLA and ULA.

The ROR then prepares a (solicited) NA(AR) message to return to the ROS with the source address set to its own ADM-LLA, the destination address set to the NS(AR) LLA source address and the Target Address set to the target Client's MNP-LLA. The ROR includes an OMNI option with Preflen set to the prefix length associated with the NA(AR) source address, with S/T-omIndex set to the value that appeared in the NS(AR) and with Interface Attributes sub-options for all of the target's underlying interfaces with current information for each interface.

For each Interface Attributes sub-option, the ROR sets the L2ADDR according to its own INET address for VPNed, Direct, ANET and NATed Client interfaces, or to the Client's INET address for native Client

interfaces. The ROR then includes the lower 32 bits of its ADM-ULA as the LHS, encodes the ADM-ULA SRT prefix length in the SRT field and sets FMT as specified in Section 3.3.

The ROR then sets the NA(AR) message R flag to 1 (as a router) and S flag to 1 (as a response to a solicitation) and sets the O flag to 0 (as a proxy). The ROR finally submits the NA(AR) for OAL encapsulation with source set to its own ULA and destination set to the same ULA that appeared in the NS(AR) OAL source, then performs OAL encapsulation using the same Identification value that appeared in the NS(AR) and finally forwards the resulting (*NET-encapsulated) carrier packet via the secured spanning tree without decrementing the network-layer TTL/Hop Limit field.

3.14.4. Relaying the NA(AR)

When the Bridge receives NA(AR) carrier packet from the ROR, it discards the *NET header and determines the next hop by consulting its standard IPv6 forwarding table for the OAL header destination address. The Bridge then decrements the OAL header Hop-Limit, re-encapsulates the carrier packet and forwards it via the SRT secured spanning tree, where it may traverse multiple OMNI link segments. The final-hop Bridge will deliver the carrier packet via the secured spanning tree to a Proxy/Server for the ROS.

3.14.5. Processing the NA(AR)

When the ROS receives the NA(AR) message, it first searches for a NCE that matches the NA(AR) target address. The ROS then processes the message the same as for standard IPv6 Address Resolution [RFC4861]. In the process, it caches all OMNI option information in the target NCE (including all Interface Attributes), and caches the NA(AR) ADM-{LLA,ULA} source addresses as the addresses of the ROR. If the ROS receives additional NA(AR) or uNA messages for this target Client with the same ADM-LLA source address but a different ADM-ULA source address, it configures the ADM-LLA corresponding to the new ADM-ULA, then caches the new ADM-{LLA,ULA} and deprecates the former ADM-{LLA,ULA}.

When the ROS is a Client, the SRT secured spanning tree will first deliver the solicited NA(AR) message to the local Proxy/Server, which re-encapsulates and forwards the message to the Client. If the Client is on a well-managed ANET, physical security and protected spectrum ensures security for the unmodified NA(AR); if the Client is on the open INET the Proxy/Server must instead include an authentication signature (while adjusting the OMNI option size, if necessary). The Proxy/Server uses its own ADM-ULA as the OAL source and the MNP-ULA of the Client as the OAL destination.

3.14.6. Forwarding Packets to Route Optimized Targets

After the ROS receives the route optimization NA(AR) and updates the target NCE, it sends additional NS(AR) messages to the ADM-ULA of the ROR to refresh the NCE ReachableTime before expiration while it still has sustained interest in this target. While the NCE remains REACHABLE, the ROS can forward packets along paths that use best underlying interface pairs based on local preferences and target Interface Attributes. The ROS selects target underlying interfaces according to traffic selectors and/or any other traffic discriminators, but must first establish window synchronization state for each target if necessary.

The ROS initiates window synchronization through a secured uncast NS/NA(WIN) exchange as specified in Section 3.2.7. The NS/NA(WIN) exchange is conducted over a first underlying interface pair and registers only those interfaces. If the ROS and target have additional underlying interface pairs serviced by the same source/destination LLAs, they may register new interfaces by sending additional NS/NA(WIN) messages but need not include window synchronization parameters. If the ROS and target have additional underlying interface pairs services by different source/destination LLAs, they must include window synchronization parameters when they send NS/NA(WIN) messages to establish NCE state for the new source/destination LLAs.

After window synchronization state has been established, the ROS and target Client can begin forwarding carrier packets while performing additional NS/NA(WIN) exchanges as above to update window state, register new interfaces and/or test reachability. The ROS sends carrier packets to the FHS Bridge discovered through the NS/NA(WIN) exchange which verifies the Identification is in window for the target Client. The FHS Bridge then forwards the carrier packets over the unsecured spanning tree to the LHS Bridge, which forwards them via LHS encapsulation to the LHS Proxy/Server or directly to the target Client itself. The target Client in turn sends packets to the ROS in the reverse direction while forwarding through the Bridges to minimize Proxy/Server load whenever possible.

While the ROS continues to actively forward packets to the target Client, it is responsible for updating window synchronization state and per-interface reachability before expiration. Window synchronization state is shared by all underlying interfaces in the ROS' NCE that use the same destination LLA so that a single NS/NA(WIN) exchange applies for all interfaces regardless of the (single) interface used to conduct the exchange. However, the window synchronization exchange only confirms target Client reachability over the specific interface used to conduct the exchange.

Reachability for other underlying interfaces that share the same window synchronization state must be determined individually using NS/NA(NUD) messages which need not be secured as long as they use in-window Identifications and do not update other state information.

3.15. Neighbor Unreachability Detection (NUD)

AERO nodes perform Neighbor Unreachability Detection (NUD) per [RFC4861] either reactively in response to persistent link-layer errors (see Section 3.11) or proactively to confirm reachability. The NUD algorithm is based on periodic control message exchanges and may further be seeded by IPv6 ND hints of forward progress, but care must be taken to avoid inferring reachability based on spoofed information. For example, IPv6 ND message exchanges that include authentication codes and/or in-window Identifications may be considered as acceptable hints of forward progress, while spurious random carrier packets should be ignored.

AERO nodes can perform NS/NA(NUD) exchanges over the OMNI link secured spanning tree (i.e. the same as described above for NS/NA(WIN)) to test reachability without risk of DoS attacks from nodes pretending to be a neighbor. These NS/NA(NUD) messages use the unicast LLAs and ULAs of the parties involved in the NUD test. When only reachability information is required without updating any other NCE state, AERO nodes can instead perform NS/NA(NUD) exchanges directly between neighbors without employing the secured spanning tree as long as they include in-window Identifications and either an authentication signature or checksum.

When an ROR directs an ROS to a target neighbor with one or more link-layer addresses, the ROS probes each unsecured target underlying interface either proactively or on-demand of carrier packets directed to the path by multilink forwarding to maintain the interface's state as reachable. Probing is performed through NS(NUD) messages over the SRT secured or unsecured spanning tree, or through NS(NUD) messages sent directly to an underlying interface of the target itself. While testing a target underlying interface, the ROS can optionally continue to forward carrier packets via alternate interfaces and/or maintain a small queue of carrier packets until target reachability is confirmed.

NS(NUD) messages are encapsulated, fragmented and transmitted as carrier packets the same as for ordinary original IP data packets, however the encapsulated destinations are the LLA of the ROS and either the ADM-LLA of the LHS Proxy/Server or the MNP-LLA of the target itself. The ROS encapsulates the NS(NUD) message the same as described in Section 3.2.7 and sets the NS(NUD) OMNI header S/T-omIndex to identify the underlying interface used for forwarding

(or to 0 if any underlying interface can be used). The ROS then fragments the OAL packet and forwards the resulting carrier packets into the unsecured spanning tree or via direct encapsulation for local segment targets.

When the target receives the NS(NUD) carrier packets, it verifies that it has a NCE for this ROS and that the Identification is in-window, then submits the carrier packets for reassembly. The node then verifies the authentication signature or checksum, then searches for Interface Attributes in its NCE for the ROS that match the NS(NUD) S/T-omIndex for the NA(NUD) reply. The node then prepares the NA(NUD) with the source and destination LLAs reversed, encapsulates and sets the OAL source and destination, sets the NA(NUD) S/T-omIndex to the index of the underlying interface the NS(NUD) arrived on and sets the Target Address to the same value included in the NS(NUD). The target next sets the R flag to 1, the S flag to 1 and the O flag to 1, then selects an in-window Identification for the ROS and performs fragmentation. The node then forwards the carrier packets into the unsecured spanning tree, directly to the ROS if it is in the local segment or directly to a Bridge in the local segment.

When the ROS receives the NA(NUD), it marks the target underlying interface tested as "reachable". Note that underlying interface states are maintained independently of the overall NCE REACHABLE state, and that a single NCE may have multiple target underlying interfaces in various states "reachable" and otherwise while the NCE state as a whole remains REACHABLE.

Note also that the exchange of NS/NA(NUD) messages has the useful side-benefit of opening holes in NATs that may be useful for NAT traversal. For example, a Client that discovers the address of a Bridge on the local SRT segment during an NS/NA(WIN) exchange with a peer that established MFIB state can send an NS(NUD) message directly to the INET address of the Bridge while including an authentication signature. The NS(NUD) will open a hole in any NATs on the path from the Client to the Bridge, and the Bridge can verify the authentication signature before returning a direct NA(NUD) to the Client's NATed L2ADDR while also including an authentication signature. Future carrier packets exchanged between the Client and peer can then be forwarded directly via the Bridge while bypassing the Client's FHS Proxy/Server.

3.16. Mobility Management and Quality of Service (QoS)

AERO is a Distributed Mobility Management (DMM) service. Each Proxy/Server is responsible for only a subset of the Clients on the OMNI link, as opposed to a Centralized Mobility Management (CMM) service

where there is a single network mobility collective entity for all Clients. Clients coordinate with their associated FHS and Hub Proxy/Servers via RS/RA exchanges to maintain the DMM profile, and the AERO routing system tracks all current Client/Proxy/Server peering relationships.

Hub Proxy/Servers provide ROR, default routing and mobility anchor point services for their dependent Clients, while FHS Proxy/Servers provide a proxy conduit between the Client and the Hub. Clients are responsible for maintaining neighbor relationships with their Proxy/Servers through periodic RS/RA exchanges, which also serves to confirm neighbor reachability. When a Client's underlying interface attributes change, the Client is responsible for updating the Hub Proxy/Server with this new information while using the FHS Proxy/Server as a first-hop conduit. The FHS Proxy/Server can also act as a proxy to perform some IPv6 ND exchanges on the Client's behalf without consuming bandwidth on the Client underlying interface.

Mobility management considerations are specified in the following sections.

3.16.1. Mobility Update Messaging

RORs accommodate Client mobility and/or multilink change events by sending secured uNA messages to each ROS in the target Client's Report List. When an ROR sends a uNA message, it sets the IPv6 source address to the its own ADM-LLA, sets the destination address to the ROS LLA (i.e., an MNP-LLA if the ROS is a Client and an ADM-LLA if the ROS is a Proxy/Server) and sets the Target Address to the Client's MNP-LLA. The ROR also includes an OMNI option with Preflen set to the prefix length associated with the Client's MNP-LLA, with Interface Attributes for the target Client's underlying interfaces and with the OMNI header S/T-omIndex set to 0. The ROR then sets the uNA R flag to 1, S flag to 0 and O flag to 1, then encapsulates the message in an OAL header with source set to its own ADM-ULA and destination set to the ROS ULA (i.e., the ADM-ULA of the ROS Proxy/Server) and sends the message into the secured spanning tree.

As discussed in Section 7.2.6 of [RFC4861], the transmission and reception of uNA messages is unreliable but provides a useful optimization. In well-connected Internetworks with robust data links uNA messages will be delivered with high probability, but in any case the ROR can optionally send up to MAX_NEIGHBOR_ADVERTISEMENT uNAs to each ROS to increase the likelihood that at least one will be received. Alternatively, the ROR can set the PNG flag in the uNA OMNI option header to request a solicited NA acknowledgement as specified in [I-D.templin-6man-omni].

When the ROS Proxy/Server receives a uNA message prepared as above, it ignores the message if the OAL destination is not its own ADM-ULA. If the uNA destination was its own ADM-LLA, the ROS Proxy/Server uses the included OMNI option information to update its NCE for the target but does not reset ReachableTime since the receipt of an unsolicited NA message from the ROR does not provide confirmation that any forward paths to the target Client are working. If the destination was the MNP-LLA of the ROS Client, the Proxy/Server instead re-encapsulates with the OAL source set to its own ADM-ULA, OAL destination set to the MNP-ULA of the ROS Client with an authentication signature if necessary, and with an in-window Identification for this Client. Finally, if the uNA message PNG flag was set, the ROS returns a solicited NA acknowledgement as specified in [I-D.templin-6man-omni].

In addition to sending uNA messages to the current set of ROSs for the target Client, the ROR also sends uNAs to the former Proxy/Server associated with the underlying interface for which the link-layer address has changed. These uNA messages update former Proxy/Servers that cannot easily detect (e.g., without active probing) when a formerly-active Client has departed. When the ROR sends the uNA, it sets the source address to its ADM-LLA, sets the destination address to the former Proxy/Server's ADM-LLA, and sets the Target Address to the Client's MNP-LLA. The ROR also includes an OMNI option with Preflen set to the prefix length associated with the Client's MNP-LLA, with Interface Attributes for the changed underlying interface, and with the OMNI header S/T-omIndex set to 0. The ROR then sets the uNA R flag to 1, S flag to 0 and O flag to 1, then encapsulates the message in an OAL header with source set to its own ADM-ULA and destination set to the ADM-ULA of the former Proxy/Server and sends the message into the secured spanning tree.

3.16.2. Announcing Link-Layer Address and/or QoS Preference Changes

When a Client needs to change its underlying Interface Attributes (e.g., due to a mobility event), the Client sends an RS message to its Hub Proxy/Server (i.e., the ROR) via a first-hop FHS Proxy/Server, if necessary. The RS includes an OMNI option with a Multilink Forwarding Parameters sub-option with the new link quality and address information. Note that the first FHS Proxy/Server may change due to the underlying interface change; any stale state in former FHS Proxy/Servers will simply expire after ReachableTime expires with no effect on the Hub Proxy/Server.

Up to MAX_RTR_SOLICITATIONS RS messages MAY be sent in parallel with sending carrier packets containing user data in case one or more RAs are lost. If all RAs are lost, the Client SHOULD re-associate with a new Proxy/Server.

When the Proxy/Server receives the Client's changes, it sends uNA messages to all nodes in the Report List the same as described in the previous section.

3.16.3. Bringing New Links Into Service

When a Client needs to bring new underlying interfaces into service (e.g., when it activates a new data link), it sends an RS message to the Hub Proxy/Server via a FHS Proxy/Server for the underlying interface (if necessary) with an OMNI option that includes Multilink Forwarding Parameters with appropriate link quality values and with link-layer address information for the new link.

3.16.4. Deactivating Existing Links

When a Client needs to deactivate an existing underlying interface, it sends an RS message to an FHS Proxy/Server with an OMNI option with appropriate Multilink Forwarding Parameter values for the deactivated link - in particular, the link quality value 0 assures that neighbors will cease to use the link.

If the Client needs to send RS messages over an underlying interface other than the one being deactivated, it MUST include Interface Attributes with appropriate link quality values for any underlying interfaces being deactivated.

Note that when a Client deactivates an underlying interface, neighbors that have received the RS/uNA messages need not purge all references for the underlying interface from their neighbor cache entries. The Client may reactivate or reuse the underlying interface and/or its omIndex at a later point in time, when it will send new RS messages to an FHS Proxy/Server with fresh interface parameters to update any neighbors.

3.16.5. Moving Between Proxy/Servers

The Client performs the procedures specified in Section 3.12.2 when it first associates with a new Hub Proxy/Server or renews its association with an existing Hub Proxy/Server.

When an FHS Proxy/Server receives the Client's RS message destined to a new Hub Proxy/Server, it forwards the RS and also sends uNA messages to inform the old Hub Proxy/Server that the Client has DEPARTED. The FHS Proxy/Server sets the uNA source to the ADM-LLA of the new Hub Proxy/Server, sets the destination to the ADM-LLA of the old Hub Proxy/Server, sets the OAL source to its own ADM-ULA and sets the OAL destination to the ADM-ULA of the old Hub Proxy/Server. The FHS Proxy/Server then submits the uNA for OAL encapsulation and

fragmentation, then forwards the resulting carrier packets into the secured spanning tree.

When the old Hub Proxy/Server receives the uNA, it changes the Client's NCE state to DEPARTED, sets the interface attributes information for the Client to point to the new Hub Proxy/Server, and resets DepartTime. After a short delay (e.g., 2 seconds) the old Hub Proxy/Server withdraws the Client's MNP from the routing system. After DepartTime expires, the old Hub Proxy/Server deletes the Client's NCE.

The old Hub Proxy/Server also iteratively sends uNA messages to each ROS in the Client's Report List with its own ADM-LLA as the source and the LLA of the ROS as the destination. The old Proxy/Server then encapsulates the uNA with OAL source address set to the ADM-ULA of the new Hub Proxy/Server and OAL destination address set to the ADM-ULA of the ROS Proxy/Server and sends the carrier packets over the secured spanning tree. When the ROS Proxy/Server receives the uNA, it forwards the message to the ROS Client if the destination is an MNP-LLA. The ROS then examines the uNA Target Address to locate the target Client's NCE and the ADM-LLA source address to identify the old Hub Proxy/Server. The ROS then caches the ULA source address as the ADM-{LLA/ULA} for the new Hub Proxy/Server for this target NCE and marks the entry as STALE. While in the STALE state, the ROS sends new NS(AR) messages using its own ULA as the OAL source and the ADM-ULA of the new Hub Proxy/Server as the OAL destination address. The new Hub Proxy/Server will then process the NS(AR) and return an NA(AR) response.

Clients SHOULD NOT move rapidly between Hub Proxy/Servers in order to avoid causing excessive oscillations in the AERO routing system. Examples of when a Client might wish to change to a different Hub Proxy/Server include a Hub Proxy/Server that has gone unreachable, topological movements of significant distance, movement to a new geographic region, movement to a new OMNI link segment, etc.

3.17. Multicast

Clients provide an IGMP (IPv4) [RFC2236] or MLD (IPv6) [RFC3810] proxy service for its EUNs and/or hosted applications [RFC4605] and act as a Protocol Independent Multicast - Sparse-Mode (PIM-SM, or simply "PIM") Designated Router (DR) [RFC7761] on the OMNI link. Proxy/Servers act as OMNI link PIM routers for Clients on ANET, VPNed or Direct interfaces, and Relays also act as OMNI link PIM routers on behalf of nodes on other links/networks.

Clients on VPNed, Direct or ANET underlying interfaces for which the ANET has deployed native multicast services forward IGMP/MLD messages

into the ANET. The IGMP/MLD messages may be further forwarded by a first-hop ANET access router acting as an IGMP/MLD-snooping switch [RFC4541], then ultimately delivered to an ANET Proxy/Server. The Proxy/Server then acts as an ROS to send NS(AR) messages to an ROR. Clients on INET and ANET underlying interfaces without native multicast services instead send NS(AR) messages as an ROS to cause their Proxy/Server forward the message to an ROR. When the ROR receives an NA(AR) response, it initiates PIM protocol messaging according to the Source-Specific Multicast (SSM) and Any-Source Multicast (ASM) operational modes as discussed in the following sections.

3.17.1. Source-Specific Multicast (SSM)

When an ROS "X" (i.e., either a Client or Proxy Server) acting as PIM router receives a Join/Prune message from a node on its downstream interfaces containing one or more ((S)ource, (G)roup) pairs, it updates its Multicast Routing Information Base (MRIB) accordingly. For each S belonging to a prefix reachable via X's non-OMNI interfaces, X then forwards the (S, G) Join/Prune to any PIM routers on those interfaces per [RFC7761].

For each S belonging to a prefix reachable via X's OMNI interface, X sends an NS(AR) message (see: Section 3.14) using its own LLA as the source address, the solicited node multicast address corresponding to S as the destination and the LLA of S as the target address. X then encapsulates the NS(AR) in an OAL header with source address set to its own ULA and destination address set to the ULA for S, then forwards the message into the secured spanning tree which delivers it to ROR "Y" that services S. The resulting NA(AR) will return an OMNI option with Interface Attributes for any underlying interfaces that are currently servicing S.

When X processes the NA(AR) it selects one or more underlying interfaces for S and performs an NS/NA(WIN) exchange over the secured spanning tree while including a PIM Join/Prune message for each multicast group of interest in the OMNI option. If S is located behind any Proxys "Z"*, each Z* then updates its MRIB accordingly and maintains the LLA of X as the next hop in the reverse path. Since Bridges forward messages not addressed to themselves without examining them, this means that the (reverse) multicast tree path is simply from each Z* (and/or S) to X with no other multicast-aware routers in the path.

Following the initial combined Join/Prune and NS/NA(WIN) messaging, X maintains a NCE for each S the same as if X was sending unicast data traffic to S. In particular, X performs additional NS/NA(WIN) exchanges to keep the NCE alive for up to t_periodic seconds

[RFC7761]. If no new Joins are received within t_periodic seconds, X allows the NCE to expire. Finally, if X receives any additional Join/Prune messages for (S,G) it forwards the messages over the secured spanning tree.

Client C that holds an MNP for source S may later depart from a first Proxy/Server Z1 and/or connect via a new Proxy/Server Z2. In that case, Y sends a uNA message to X the same as specified for unicast mobility in Section 3.16. When X receives the uNA message, it updates its NCE for the LLA for source S and sends new Join messages in NS/NA(WIN) exchanges addressed to the new target Client underlying interface connection for S. There is no requirement to send any Prune messages to old Proxy/Server Z1 since source S will no longer source any multicast data traffic via Z1. Instead, the multicast state for (S,G) in Proxy/Server Z1 will soon expire since no new Joins will arrive.

3.17.2. Any-Source Multicast (ASM)

When an ROS X acting as a PIM router receives Join/Prune messages from a node on its downstream interfaces containing one or more (*,G) pairs, it updates its Multicast Routing Information Base (MRIB) accordingly. X first performs an NS/NA(AR) exchange to receive route optimization information for Rendezvous Point (RP) R for each G. X then includes a copy of each Join/Prune message in the OMNI option of an NS(WIN) message with its own LLA as the source address and the LLA for R as the destination address, then encapsulates the NS(WIN) message in an OAL header with its own ULA as the source and the ADM-ULA of R's Proxy/Server as the destination then sends the message into the secured spanning tree.

For each source S that sends multicast traffic to group G via R, Client S* that aggregates S (or its Proxy/Server) encapsulates the original IP packets in PIM Register messages, includes the PIM Register messages in the OMNI options of uNA messages, performs OAL encapsulation and fragmentation then forwards the resulting carrier packets with Identification values within the receive window for Client R* that aggregates R. Client R* may then elect to send a PIM Join to S* in the OMNI option of a uNA over the secured spanning tree. This will result in an (S,G) tree rooted at S* with R as the next hop so that R will begin to receive two copies of the original IP packet; one native copy from the (S, G) tree and a second copy from the pre-existing (*, G) tree that still uses uNA PIM Register encapsulation. R can then issue a uNA PIM Register-stop message over the secured spanning tree to suppress the Register-encapsulated stream. At some later time, if Client S* moves to a new Proxy/Server, it resumes sending original IP packets via uNA PIM Register encapsulation via the new Proxy/Server.

At the same time, as multicast listeners discover individual S's for a given G, they can initiate an (S,G) Join for each S under the same procedures discussed in Section 3.17.1. Once the (S,G) tree is established, the listeners can send (S, G) Prune messages to R so that multicast original IP packets for group G sourced by S will only be delivered via the (S, G) tree and not from the (*, G) tree rooted at R. All mobility considerations discussed for SSM apply.

3.17.3. Bi-Directional PIM (BIDIR-PIM)

Bi-Directional PIM (BIDIR-PIM) [RFC5015] provides an alternate approach to ASM that treats the Rendezvous Point (RP) as a Designated Forwarder (DF). Further considerations for BIDIR-PIM are out of scope.

3.18. Operation over Multiple OMNI Links

An AERO Client can connect to multiple OMNI links the same as for any data link service. In that case, the Client maintains a distinct OMNI interface for each link, e.g., 'omni0' for the first link, 'omni1' for the second, 'omni2' for the third, etc. Each OMNI link would include its own distinct set of Bridges and Proxy/Servers, thereby providing redundancy in case of failures.

Each OMNI link could utilize the same or different ANET connections. The links can be distinguished at the link-layer via the SRT prefix in a similar fashion as for Virtual Local Area Network (VLAN) tagging (e.g., IEEE 802.1Q) and/or through assignment of distinct sets of MSPs on each link. This gives rise to the opportunity for supporting multiple redundant networked paths (see: Section 3.2.5).

The Client's IP layer can select the outgoing OMNI interface appropriate for a given traffic profile while (in the reverse direction) correspondent nodes must have some way of steering their original IP packets destined to a target via the correct OMNI link.

In a first alternative, if each OMNI link services different MSPs the Client can receive a distinct MNP from each of the links. IP routing will therefore assure that the correct OMNI link is used for both outbound and inbound traffic. This can be accomplished using existing technologies and approaches, and without requiring any special supporting code in correspondent nodes or Bridges.

In a second alternative, if each OMNI link services the same MSP(s) then each link could assign a distinct "OMNI link Anycast" address that is configured by all Bridges on the link. Correspondent nodes can then perform Segment Routing to select the correct SRT, which

will then direct the original IP packet over multiple hops to the target.

3.19. DNS Considerations

AERO Client MNs and INET correspondent nodes consult the Domain Name System (DNS) the same as for any Internetworking node. When correspondent nodes and Client MNs use different IP protocol versions (e.g., IPv4 correspondents and IPv6 MNs), the INET DNS must maintain A records for IPv4 address mappings to MNs which must then be populated in Relay NAT64 mapping caches. In that way, an IPv4 correspondent node can send original IPv4 packets to the IPv4 address mapping of the target MN, and the Relay will translate the IPv4 header and destination address into an IPv6 header and IPv6 destination address of the MN.

When an AERO Client registers with an AERO Proxy/Server, the Proxy/Server can return the address(es) of DNS servers in RDNSS options [RFC6106]. The DNS server provides the IP addresses of other MNs and correspondent nodes in AAAA records for IPv6 or A records for IPv4.

3.20. Transition/Coexistence Considerations

OAL encapsulation ensures that dissimilar INET partitions can be joined into a single unified OMNI link, even though the partitions themselves may have differing protocol versions and/or incompatible addressing plans. However, a commonality can be achieved by incrementally distributing globally routable (i.e., native) IP prefixes to eventually reach all nodes (both mobile and fixed) in all OMNI link segments. This can be accomplished by incrementally deploying AERO Bridges on each INET partition, with each Bridge distributing its MNPs and/or discovering non-MNP IP GUA prefixes on its INET links.

This gives rise to the opportunity to eventually distribute native IP addresses to all nodes, and to present a unified OMNI link view even if the INET partitions remain in their current protocol and addressing plans. In that way, the OMNI link can serve the dual purpose of providing a mobility/multilink service and a transition/coexistence service. Or, if an INET partition is transitioned to a native IP protocol version and addressing scheme that is compatible with the OMNI link MNP-based addressing scheme, the partition and OMNI link can be joined by Bridges.

Relays that connect INETs/EUNs with dissimilar IP protocol versions may need to employ a network address and protocol translation function such as NAT64 [RFC6146].

3.21. Detecting and Reacting to Proxy/Server and Bridge Failures

In environments where rapid failure recovery is required, Proxy/Servers and Bridges SHOULD use Bidirectional Forwarding Detection (BFD) [RFC5880]. Nodes that use BFD can quickly detect and react to failures so that cached information is re-established through alternate nodes. BFD control messaging is carried only over well-connected ground domain networks (i.e., and not low-end radio links) and can therefore be tuned for rapid response.

Proxy/Servers and Bridges maintain BFD sessions in parallel with their BGP peerings. If a Proxy/Server or Bridge fails, BGP peers will quickly re-establish routes through alternate paths the same as for common BGP deployments. Similarly, Proxys maintain BFD sessions with their associated Bridges even though they do not establish BGP peerings with them.

3.22. AERO Clients on the Open Internet

AERO Clients that connect to the open Internet via INET interfaces can establish a VPN or direct link to securely connect to a FHS/Hub Proxy/Server in a "tethered" arrangement with all of the Client's traffic transiting the Proxy/Server which acts as a router. Alternatively, the Client can associate with an INET FHS/Hub Proxy/Server using UDP/IP encapsulation and control message securing services as discussed in the following sections.

When a Client's OMNI interface enables an INET underlying interface, it first examines the INET address. For IPv4, the Client assumes it is on the open Internet if the INET address is not a special-use IPv4 address per [RFC3330]. Similarly for IPv6, the Client assumes it is on the open Internet if the INET address is a Global Unicast Address (GUA) [RFC4291]. Otherwise, the Client should assume it is behind one or several NATs.

The Client then prepares an RS message with IPv6 source address set to its MNP-LLA, with IPv6 destination set to (link-local) All-Routers multicast and with an OMNI option with underlying interface attributes. If the Client believes that it is on the open Internet, it SHOULD include its IP address and UDP port number in the Multilink Forwarding Parameters sub-option corresponding to the underlying interface. If the underlying address is IPv4, the Client includes the Port Number and IPv4 address written in obfuscated form [RFC4380] as discussed in Section 3.3. If the underlying interface address is IPv6, the Client instead includes the Port Number and IPv6 address in obfuscated form. The Client finally includes an authentication signature per [I-D.templin-6man-omni] to provide message authentication, selects an Identification value and window

synchronization parameters, and submits the RS for OAL encapsulation. The Client then encapsulates the OAL atomic fragment in UDP/IP headers to form a carrier packet, sets the UDP/IP source to its INET address and UDP port, sets the UDP/IP destination to the FHS Proxy/ Server's INET address and the AERO service port number (8060), then sends the carrier packet to the Proxy/Server.

When the FHS Proxy/Server receives the RS, it discards the OAL encapsulation, authenticates the RS message, and examines the destination address. If the destination is the ADM-LLA of another Proxy/Server, the FHS Proxy/Server assumes the proxy role and forwards the message into the secured spanning tree. If the destination is All-Routers multicast or its own ADM-LLA, the FHS Proxy/Server instead assumes the Hub role, creates a NCE and registers the Client's MNP, window synchronization state and INET interface information according to the OMNI option parameters. If the Multilink Forwarding Paramters sub-option includes a non-zero L2ADDR, the Hub Proxy/Server compares the encapsulation IP address and UDP port number with the (unobfuscated) values. If the values are the same, the Hub Proxy/Server caches the Client's information as an "INET" address meaning that the Client is likely to accept direct messages without requiring NAT traversal exchanges. If the values are different (or, if the OMNI option did not include an L2ADDR) the Hub Proxy/Server instead caches the Client's information as a "mapped" address meaning that NAT traversal exchanges may be necessary.

The Hub Proxy/Server then prepares an RA message with IPv6 source and destination set corresponding to the addresses in the RS, and with an OMNI option with an Origin Indication sub-option per [I-D.templin-6man-omni] with the mapped and obfuscated Port Number and IP address observed in the encapsulation headers. The Proxy/ Server also includes a Multilink Forwarding Parameters sub-option, an authentication signature sub-option per [I-D.templin-6man-omni] and/ or a symmetric window synchronization/acknowledgement if necessary. The Hub Proxy/Server then performs OAL encapsulation then encapsulates the carrier packet in UDP/IP headers with addresses set per the L2ADDR information in the NCE for the Client.

When the Client receives the RA, it authenticates the message then process the window synchronization/acknowledgement and compares the mapped Port Number and IP address from the Multilink Forwarding Parameters sub-option with its own address. If the addresses are the same, the Client assumes the open Internet / Cone NAT principle; if the addresses are different, the Client instead assumes that further qualification procedures are necessary to detect the type of NAT and performs NAT traversal on-demand according to standard procedures [RFC6081][RFC4380]. The Client also caches the RA rest of the

Multilink Forwarding Parameters information to discover the FHS Proxy/Server's local spanning tree segment. The Client finally arranges to return an explicit/implicit acknowledgement, and sends periodic RS messages to receive fresh RA messages before the Router Lifetime received on each INET interface expires.

When the Client sends messages to target IP addresses, it also invokes route optimization per Section 3.14. For route optimized targets in the same OMNI link segment, if the target's L2ADDR is on the open INET, the Client forwards carrier packets directly to the target INET address. If the target is behind a NAT, the Client first establishes NAT state for the L2ADDR using the "direct bubble" and NS/NA(NUD) mechanisms discussed in Section 3.10.1. The Client continues to send carrier packets via the local Bridge discovered during window synchronization until NAT state is populated, then begins forwarding carrier packets via the direct path through the NAT to the target. For targets in different OMNI link segments, the Client forwards carrier packets to the local Bridge.

The Client can send original IP packets to route-optimized neighbors in the same OMNI link segment no larger than the minimum/path MPS in one piece and with OAL encapsulation as atomic fragments. For larger original IP packets, the Client applies OAL encapsulation then fragments if necessary according to Section 3.9, with OAL header with source set to its own MNP-ULA and destination set to the MNP-ULA of the target, and with an in-window Identification value. The Client then encapsulates each resulting carrier packet in UDP/IP *NET headers and sends them to the neighbor.

INET Clients exchange NS/NA(WIN) messages to associate with a new peer as discussed in Section 3.2.7. The exchange establishes MFIB state in the Client, peer and all OMNI intermediate nodes in the path. After MFIB state is established, INET Clients and peers can exchange carrier packets with compressed headers that include an MFVI which is updated on a hop-by-hop basis, while employing "shortcuts" to skip any unnecessary hops.

Note: The NAT traversal procedures specified in this document are applicable for Cone, Address-Restricted and Port-Restricted NATs only. While future updates to this document may specify procedures for other NAT variations (e.g., hairpinning and various forms of Symmetric NATs), it should be noted that continuous communications are always possible through Proxy/Server forwarding even for these other NAT variations.

3.23. Time-Varying MNPs

In some use cases, it is desirable, beneficial and efficient for the Client to receive a constant MNP that travels with the Client wherever it moves. For example, this would allow air traffic controllers to easily track aircraft, etc. In other cases, however (e.g., intelligent transportation systems), the MN may be willing to sacrifice a modicum of efficiency in order to have time-varying MNPs that can be changed every so often to defeat adversarial tracking.

The DHCPv6 service offers a way for Clients that desire time-varying MNPs to obtain short-lived prefixes (e.g., on the order of a small number of minutes). In that case, the identity of the Client would not be bound to the MNP but rather to a Node Identification value (see: [I-D.templin-6man-omni]) to be used as the Client ID seed for MNP prefix delegation. The Client would then be obligated to renumber its internal networks whenever its MNP (and therefore also its MNP-LLA) changes. This should not present a challenge for Clients with automated network renumbering services, however presents limits for the durations of ongoing sessions that would prefer to use a constant address.

4. Implementation Status

An early AERO implementation based on OpenVPN (https://openvpn.net/) was announced on the v6ops mailing list on January 10, 2018 and an initial public release of the AERO proof-of-concept source code was announced on the intarea mailing list on August 21, 2015.

AERO Release-3.2 was tagged on March 30, 2021, and is undergoing internal testing. Additional internal releases expected within the coming months, with first public release expected end of 1H2021.

Many AERO/OMNI functions are implemented and undergoing final integration. OAL fragmentation/reassembly buffer management code has been cleared for public release and will be presented at the June 2021 ICAO mobility subgroup meeting.

5. IANA Considerations

The IANA has assigned the UDP port number "8060" for an earlier experimental first version of AERO [RFC6706]. This document together with [I-D.templin-6man-omni] reclaims UDP port number "8060" as the service port for UDP/IP encapsulation. This document makes no request of IANA, since [I-D.templin-6man-omni] already provides instructions. (Note: although [RFC6706] was not widely implemented or deployed, it need not be obsoleted since its messages use the

invalid ICMPv6 message type number '0' which implementations of this specification can easily distinguish and ignore.)

No further IANA actions are required.

6. Security Considerations

AERO Bridges configure secured tunnels with AERO Proxy/Servers and Relays within their local OMNI link segments. Applicable secured tunnel alternatives include IPsec [RFC4301], TLS/SSL [RFC8446], DTLS [RFC6347], WireGuard [WG], etc. The AERO Bridges of all OMNI link segments in turn configure secured tunnels for their neighboring AERO Bridges in a secured spanning tree topology. Therefore, control messages exchanged between any pair of OMNI link neighbors over the secured spanning tree are already protected.

To prevent spoofing vectors, Proxy/Servers MUST discard without responding to any unsecured NS(AR) messages. Also, Proxy/Servers MUST discard without forwarding any original IP packets received from one of their own Clients (whether directly or following OAL reassembly) with a source address that does not match the Client's MNP and/or a destination address that does match the Client's MNP. Finally, Proxy/Servers MUST discard without forwarding any carrier packets with an OAL source and destination that both match the same MNP.

For INET partitions that require strong security in the data plane, two options for securing communications include 1) disable route optimization so that all traffic is conveyed over secured tunnels, or 2) enable on-demand secure tunnel creation between Client neighbors. Option 1) would result in longer routes than necessary and impose traffic concentration on critical infrastructure elements. Option 2) could be coordinated between Clients using NS/NA messages with OMNI Host Identity Protocol (HIP) "Initiator/Responder" message sub-options [RFC7401][I-D.templin-6man-omni] to create a secured tunnel on-demand.

AERO Clients that connect to secured ANETs need not apply security to their IPv6 ND messages, since the messages will be authenticated and forwarded by a perimeter Proxy/Server that applies security on its INET-facing interface as part of the spanning tree (see above). AERO Clients connected to the open INET can use network and/or transport layer security services such as VPNs or can by some other means establish a direct link to a Proxy/Server. When a VPN or direct link may be impractical, however, INET Clients and Proxy/Servers SHOULD include and verify authentication signatures for their IPv6 ND messages as specified in [I-D.templin-6man-omni].

Application endpoints SHOULD use transport-layer (or higher-layer) security services such as TLS/SSL, DTLS or SSH [RFC4251] to assure the same level of protection as for critical secured Internet services. AERO Clients that require host-based VPN services SHOULD use network and/or transport layer security services such as IPsec, TLS/SSL, DTLS, etc. AERO Proxys and Proxy/Servers can also provide a network-based VPN service on behalf of the Client, e.g., if the Client is located within a secured enclave and cannot establish a VPN on its own behalf.

AERO Proxy/Servers and Bridges present targets for traffic amplification Denial of Service (DoS) attacks. This concern is no different than for widely-deployed VPN security gateways in the Internet, where attackers could send spoofed packets to the gateways at high data rates. This can be mitigated through the AERO/OMNI data origin authentication procedures, as well as connecting Proxy/Servers and Bridges over dedicated links with no connections to the Internet and/or when connections to the Internet are only permitted through well-managed firewalls. Traffic amplification DoS attacks can also target an AERO Client's low data rate links. This is a concern not only for Clients located on the open Internet but also for Clients in secured enclaves. AERO Proxy/Servers and Proxys can institute rate limits that protect Clients from receiving packet floods that could DoS low data rate links.

AERO Relays must implement ingress filtering to avoid a spoofing attack in which spurious messages with ULA addresses are injected into an OMNI link from an outside attacker. AERO Clients MUST ensure that their connectivity is not used by unauthorized nodes on their EUNs to gain access to a protected network, i.e., AERO Clients that act as routers MUST NOT provide routing services for unauthorized nodes. (This concern is no different than for ordinary hosts that receive an IP address delegation but then "share" the address with other nodes via some form of Internet connection sharing such as tethering.)

The PRL MUST be well-managed and secured from unauthorized tampering, even though the list contains only public information. The PRL can be conveyed to the Client in a similar fashion as in [RFC5214] (e.g., through layer 2 data link login messaging, secure upload of a static file, DNS lookups, etc.).

The AERO service for open INET Clients depends on a public key distribution service in which Client public keys and identities are maintained in a shared database accessible to all open INET Proxy/Servers. Similarly, each Client must be able to determine the public key of each Proxy/Server, e.g. by consulting an online database. When AERO nodes register their public keys indexed by a unique Host

Identity Tag (HIT) [RFC7401] in a distributed database such as the DNS, and use the HIT as an identity for applying IPv6 ND message authentication signatures, a means for determining public key attestation is available.

Security considerations for IPv6 fragmentation and reassembly are discussed in [I-D.templin-6man-omni]. In environments where spoofing is considered a threat, OMNI nodes SHOULD employ Identification window synchronization and OAL destinations SHOULD configure an (end-system-based) firewall.

SRH authentication facilities are specified in [RFC8754]. Security considerations for accepting link-layer ICMP messages and reflected packets are discussed throughout the document.

7. Acknowledgements

Discussions in the IETF, aviation standards communities and private exchanges helped shape some of the concepts in this work. Individuals who contributed insights include Mikael Abrahamsson, Mark Andrews, Fred Baker, Bob Braden, Stewart Bryant, Scott Burleigh, Brian Carpenter, Wojciech Dec, Pavel Drasil, Ralph Droms, Adrian Farrel, Nick Green, Sri Gundavelli, Brian Haberman, Bernhard Haindl, Joel Halpern, Tom Herbert, Bob Hinden, Sascha Hlusiak, Lee Howard, Christian Huitema, Zdenek Jaron, Andre Kostur, Hubert Kuenig, Ted Lemon, Andy Malis, Satoru Matsushima, Tomek Mrugalski, Thomas Narten, Madhu Niraula, Alexandru Petrescu, Behcet Saikaya, Michal Skorepa, Dave Thaler, Joe Touch, Bernie Volz, Ryuji Wakikawa, Tony Whyman, Lloyd Wood and James Woodyatt. Members of the IESG also provided valuable input during their review process that greatly improved the document. Special thanks go to Stewart Bryant, Joel Halpern and Brian Haberman for their shepherding guidance during the publication of the AERO first edition.

This work has further been encouraged and supported by Boeing colleagues including Kyle Bae, M. Wayne Benson, Dave Bernhardt, Cam Brodie, John Bush, Balaguruna Chidambaram, Irene Chin, Bruce Cornish, Claudiu Danilov, Don Dillenburg, Joe Dudkowski, Wen Fang, Samad Farooqui, Anthony Gregory, Jeff Holland, Seth Jahne, Brian Jaury, Greg Kimberly, Ed King, Madhuri Madhava Badgandi, Laurel Matthew, Gene MacLean III, Kyle Mikos, Rob Muszkiewicz, Sean O'Sullivan, Vijay Rajagopalan, Greg Saccone, Rod Santiago, Kent Shuey, Brian Skeen, Mike Slane, Carrie Spiker, Katie Tran, Brendan Williams, Amelia Wilson, Julie Wulff, Yueli Yang, Eric Yeh and other members of the Boeing mobility, networking and autonomy teams. Kyle Bae, Wayne Benson, Madhuri Madhava Badgandi, Vijayasarathy Rajagopalan, Katie Tran and Eric Yeh are especially acknowledged for their work on the

AERO implementation. Chuck Klabunde is honored and remembered for his early leadership, and we mourn his untimely loss.

This work was inspired by the support and encouragement of countless outstanding colleagues, managers and program directors over the span of many decades. Beginning in the late 1980s,' the Digital Equipment Corporation (DEC) Ultrix Engineering and DECnet Architects groups identified early issues with fragmentation and bridging links with diverse MTUs. In the early 1990s, engagements at DEC Project Sequoia at UC Berkeley and the DEC Western Research Lab in Palo Alto included investigations into large-scale networked filesystems, ATM vs Internet and network security proxies. In the mid-1990s to early 2000s employment at the NASA Ames Research Center (Sterling Software) and SRI International supported early investigations of IPv6, ONR UAV Communications and the IETF. An employment at Nokia where important IETF documents were published gave way to a present-day engagement with The Boeing Company. The work matured at Boeing through major programs including Future Combat Systems, Advanced Airplane Program, DTN for the International Space Station, Mobility Vision Lab, CAST, Caravan, Airplane Internet of Things, the NASA UAS/CNS program, the FAA/ICAO ATN/IPS program and many others. An attempt to name all who gave support and encouragement would double the current document size and result in many unintentional omissions - but to all a humble thanks.

Earlier works on NBMA tunneling approaches are found in [RFC2529][RFC5214][RFC5569].

Many of the constructs presented in this second edition of AERO are based on the author's earlier works, including:

- The Internet Routing Overlay Network (IRON) [RFC6179][I-D.templin-ironbis]

- Virtual Enterprise Traversal (VET) [RFC5558][I-D.templin-intarea-vet]

- The Subnetwork Encapsulation and Adaptation Layer (SEAL) [RFC5320][I-D.templin-intarea-seal]

- AERO, First Edition [RFC6706]

Note that these works cite numerous earlier efforts that are not also cited here due to space limitations. The authors of those earlier works are acknowledged for their insights.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the Boeing Commercial Airplanes (BCA) Internet of Things (IoT) and autonomy programs.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program.

8. References

8.1. Normative References

[I-D.templin-6man-omni]
Templin, F. L. and T. Whyman, "Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces", draft-templin-6man-omni-03 (work in progress), April 2021.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <https://www.rfc-editor.org/info/rfc3971>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <https://www.rfc-editor.org/info/rfc3972>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/info/rfc4193>.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <https://www.rfc-editor.org/info/rfc4380>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC6081] Thaler, D., "Teredo Extensions", RFC 6081, DOI 10.17487/RFC6081, January 2011, <https://www.rfc-editor.org/info/rfc6081>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <https://www.rfc-editor.org/info/rfc7401>.

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <https://www.rfc-editor.org/info/rfc7739>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

8.2. Informative References

[BGP] Huston, G., "BGP in 2015, http://potaroo.net", January 2016.

[I-D.bonica-6man-comp-rtg-hdr] Bonica, R., Kamite, Y., Alston, A., Henriques, D., and L. Jalil, "The IPv6 Compact Routing Header (CRH)", draft-bonica-6man-comp-rtg-hdr-24 (work in progress), January 2021.

[I-D.bonica-6man-crh-helper-opt] Li, X., Bao, C., Ruan, E., and R. Bonica, "Compressed Routing Header (CRH) Helper Option", draft-bonica-6man-crh-helper-opt-03 (work in progress), April 2021.

[I-D.ietf-intarea-frag-fragile] Bonica, R., Baker, F., Huston, G., Hinden, R. M., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", draft-ietf-intarea-frag-fragile-17 (work in progress), September 2019.

[I-D.ietf-intarea-tunnels] Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", draft-ietf-intarea-tunnels-10 (work in progress), September 2019.

[I-D.ietf-ipwave-vehicular-networking] (editor), J. (. J., "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases", draft-ietf-ipwave-vehicular-networking-20 (work in progress), March 2021.

[I-D.ietf-rtgwg-atn-bgp] Templin, F. L., Saccone, G., Dawra, G., Lindem, A., and V. Moreno, "A Simple BGP-based Mobile Routing System for the Aeronautical Telecommunications Network", draft-ietf-rtgwg-atn-bgp-10 (work in progress), January 2021.

[I-D.templin-6man-dhcpv6-ndopt] Templin, F. L., "A Unified Stateful/Stateless Configuration Service for IPv6", draft-templin-6man-dhcpv6-ndopt-11 (work in progress), January 2021.

[I-D.templin-intarea-seal] Templin, F. L., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", draft-templin-intarea-seal-68 (work in progress), January 2014.

[I-D.templin-intarea-vet]
Templin, F. L., "Virtual Enterprise Traversal (VET)", draft-templin-intarea-vet-40 (work in progress), May 2013.

[I-D.templin-ipwave-uam-its]
Templin, F. L., "Urban Air Mobility Implications for Intelligent Transportation Systems", draft-templin-ipwave-uam-its-04 (work in progress), January 2021.

[I-D.templin-ironbis]
Templin, F. L., "The Interior Routing Overlay Network (IRON)", draft-templin-ironbis-16 (work in progress), March 2014.

[I-D.templin-v6ops-pdhost]
Templin, F. L., "IPv6 Prefix Delegation and Multi-Addressing Models", draft-templin-v6ops-pdhost-27 (work in progress), January 2021.

[OVPN] OpenVPN, O., "http://openvpn.net", October 2016.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <https://www.rfc-editor.org/info/rfc1812>.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <https://www.rfc-editor.org/info/rfc2003>.

[RFC2004] Perkins, C., "Minimal Encapsulation within IP", RFC 2004, DOI 10.17487/RFC2004, October 1996, <https://www.rfc-editor.org/info/rfc2004>.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <https://www.rfc-editor.org/info/rfc2236>.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <https://www.rfc-editor.org/info/rfc2464>.

[RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, DOI 10.17487/RFC2529, March 1999, <https://www.rfc-editor.org/info/rfc2529>.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, DOI 10.17487/RFC2983, October 2000, <https://www.rfc-editor.org/info/rfc2983>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3330] IANA, "Special-Use IPv4 Addresses", RFC 3330, DOI 10.17487/RFC3330, September 2002, <https://www.rfc-editor.org/info/rfc3330>.

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/info/rfc3810>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/info/rfc4122>.

[RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/info/rfc4251>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, DOI 10.17487/RFC4389, April 2006, <https://www.rfc-editor.org/info/rfc4389>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4511] Sermersheim, J., Ed., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, DOI 10.17487/RFC4511, June 2006, <https://www.rfc-editor.org/info/rfc4511>.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, DOI 10.17487/RFC4541, May 2006, <https://www.rfc-editor.org/info/rfc4541>.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, DOI 10.17487/RFC4605, August 2006, <https://www.rfc-editor.org/info/rfc4605>.

[RFC4982] Bagnulo, M. and J. Arkko, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)", RFC 4982, DOI 10.17487/RFC4982, July 2007, <https://www.rfc-editor.org/info/rfc4982>.

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <https://www.rfc-editor.org/info/rfc5015>.

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, DOI 10.17487/RFC5214, March 2008, <https://www.rfc-editor.org/info/rfc5214>.

[RFC5320] Templin, F., Ed., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", RFC 5320, DOI 10.17487/RFC5320, February 2010, <https://www.rfc-editor.org/info/rfc5320>.

[RFC5522] Eddy, W., Ivancic, W., and T. Davis, "Network Mobility Route Optimization Requirements for Operational Use in Aeronautics and Space Exploration Mobile Networks", RFC 5522, DOI 10.17487/RFC5522, October 2009, <https://www.rfc-editor.org/info/rfc5522>.

[RFC5558] Templin, F., Ed., "Virtual Enterprise Traversal (VET)", RFC 5558, DOI 10.17487/RFC5558, February 2010, <https://www.rfc-editor.org/info/rfc5558>.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, DOI 10.17487/RFC5569, January 2010, <https://www.rfc-editor.org/info/rfc5569>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, DOI 10.17487/RFC6106, November 2010, <https://www.rfc-editor.org/info/rfc6106>.

[RFC6139] Russert, S., Ed., Fleischman, E., Ed., and F. Templin, Ed., "Routing and Addressing in Networks with Global Enterprise Recursion (RANGER) Scenarios", RFC 6139, DOI 10.17487/RFC6139, February 2011, <https://www.rfc-editor.org/info/rfc6139>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC6179] Templin, F., Ed., "The Internet Routing Overlay Network (IRON)", RFC 6179, DOI 10.17487/RFC6179, March 2011, <https://www.rfc-editor.org/info/rfc6179>.

[RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, DOI 10.17487/RFC6221, May 2011, <https://www.rfc-editor.org/info/rfc6221>.

[RFC6273] Kukec, A., Krishnan, S., and S. Jiang, "The Secure Neighbor Discovery (SEND) Hash Threat Analysis", RFC 6273, DOI 10.17487/RFC6273, June 2011, <https://www.rfc-editor.org/info/rfc6273>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6355] Narten, T. and J. Johnson, "Definition of the UUID-Based DHCPv6 Unique Identifier (DUID-UUID)", RFC 6355, DOI 10.17487/RFC6355, August 2011, <https://www.rfc-editor.org/info/rfc6355>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

[RFC6706] Templin, F., Ed., "Asymmetric Extended Route Optimization (AERO)", RFC 6706, DOI 10.17487/RFC6706, August 2012, <https://www.rfc-editor.org/info/rfc6706>.

[RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, DOI 10.17487/RFC6935, April 2013, <https://www.rfc-editor.org/info/rfc6935>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", RFC 7333, DOI 10.17487/RFC7333, August 2014, <https://www.rfc-editor.org/info/rfc7333>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[WG] Wireguard, "WireGuard, https://www.wireguard.com", August 2020.

Appendix A. Non-Normative Considerations

AERO can be applied to a multitude of Internetworking scenarios, with each having its own adaptations. The following considerations are provided as non-normative guidance:

A.1. Implementation Strategies for Route Optimization

Route optimization as discussed in Section 3.14 results in the route optimization source (ROS) creating a NCE for the target neighbor. The NCE state is set to REACHABLE for at most ReachableTime seconds. In order to refresh the NCE lifetime before the ReachableTime timer expires, the specification requires implementations to issue a new NS/NA(AR) exchange to reset ReachableTime while data packets are still flowing. However, the decision of when to initiate a new NS/NA(AR) exchange and to perpetuate the process is left as an implementation detail.

One possible strategy may be to monitor the NCE watching for data packets for (ReachableTime - 5) seconds. If any data packets have been sent to the neighbor within this timeframe, then send an NS(AR) to receive a new NA(AR). If no data packets have been sent, wait for 5 additional seconds and send an immediate NS(AR) if any data packets are sent within this "expiration pending" 5 second window. If no additional data packets are sent within the 5 second window, reset the NCE state to STALE.

The monitoring of the neighbor data packet traffic therefore becomes an ongoing process during the NCE lifetime. If the NCE expires, future data packets will trigger a new NS/NA(AR) exchange while the packets themselves are delivered over a longer path until route optimization state is re-established.

A.2. Implicit Mobility Management

OMNI interface neighbors MAY provide a configuration option that allows them to perform implicit mobility management in which no IPv6 ND messaging is used. In that case, the Client only transmits packets over a single interface at a time, and the neighbor always observes packets arriving from the Client from the same link-layer source address.

If the Client's underlying interface address changes (either due to a readdressing of the original interface or switching to a new interface) the neighbor immediately updates the NCE for the Client

and begins accepting and sending packets according to the Client's new address. This implicit mobility method applies to use cases such as cellphones with both WiFi and Cellular interfaces where only one of the interfaces is active at a given time, and the Client automatically switches over to the backup interface if the primary interface fails.

A.3. Direct Underlying Interfaces

When a Client's OMNI interface is configured over a Direct interface, the neighbor at the other end of the Direct link can receive packets without any encapsulation. In that case, the Client sends packets over the Direct link according to traffic selectors. If the Direct interface is selected, then the Client's IP packets are transmitted directly to the peer without going through an ANET/INET. If other interfaces are selected, then the Client's IP packets are transmitted via a different interface, which may result in the inclusion of Proxy/Servers and Bridges in the communications path. Direct interfaces must be tested periodically for reachability, e.g., via NUD.

A.4. AERO Critical Infrastructure Considerations

AERO Bridges can be either Commercial off-the Shelf (COTS) standard IP routers or virtual machines in the cloud. Bridges must be provisioned, supported and managed by the INET administrative authority, and connected to the Bridges of other INETs via inter-domain peerings. Cost for purchasing, configuring and managing Bridges is nominal even for very large OMNI links.

AERO INET Proxy/Servers can be standard dedicated server platforms, but most often will be deployed as virtual machines in the cloud. The only requirements for INET Proxy/Servers are that they can run the AERO/OMNI code and have at least one network interface connection to the INET. INET Proxy/Servers must be provisioned, supported and managed by the INET administrative authority. Cost for purchasing, configuring and managing cloud Proxy/Servers is nominal especially for virtual machines.

AERO ANET Proxy/Servers are most often standard dedicated server platforms with one underlying interface connected to the ANET and a second interface connected to an INET. As with INET Proxy/Servers, the only requirements are that they can run the AERO/OMNI code and have at least one interface connection to the INET. ANET Proxy/Servers must be provisioned, supported and managed by the ANET administrative authority. Cost for purchasing, configuring and managing Proxys is nominal, and borne by the ANET administrative authority.

AERO Relays are simply Proxy/Servers connected to INETs and/or EUNs that provide forwarding services for non-MNP destinations. The Relay connects to the OMNI link and engages in eBGP peering with one or more Bridges as a stub AS. The Relay then injects its MNPs and/or non-MNP prefixes into the BGP routing system, and provisions the prefixes to its downstream-attached networks. The Relay can perform ROS/ROR services the same as for any Proxy/Server, and can route between the MNP and non-MNP address spaces.

A.5. AERO Server Failure Implications

AERO Proxy/Servers may appear as a single point of failure in the architecture, but such is not the case since all Proxy/Servers on the link provide identical services and loss of a Proxy/Server does not imply immediate and/or comprehensive communication failures. Proxy/Server failure is quickly detected and conveyed by Bidirectional Forward Detection (BFD) and/or proactive NUD allowing Clients to migrate to new Proxy/Servers.

If a Proxy/Server fails, ongoing packet forwarding to Clients will continue by virtue of the neighbor cache entries that have already been established in route optimization sources (ROSs). If a Client also experiences mobility events at roughly the same time the Proxy/Server fails, uNA messages may be lost but neighbor cache entries in the DEPARTED state will ensure that packet forwarding to the Client's new locations will continue for up to DepartTime seconds.

If a Client is left without a Proxy/Server for a considerable length of time (e.g., greater than ReachableTime seconds) then existing neighbor cache entries will eventually expire and both ongoing and new communications will fail. The original source will continue to retransmit until the Client has established a new Proxy/Server relationship, after which time continuous communications will resume.

Therefore, providing many Proxy/Servers on the link with high availability profiles provides resilience against loss of individual Proxy/Servers and assurance that Clients can establish new Proxy/Server relationships quickly in event of a Proxy/Server failure.

A.6. AERO Client / Server Architecture

The AERO architectural model is client / server in the control plane, with route optimization in the data plane. The same as for common Internet services, the AERO Client discovers the addresses of AERO Proxy/Servers and connects to one or more of them. The AERO service is analogous to common Internet services such as google.com, yahoo.com, cnn.com, etc. However, there is only one AERO service for the link and all Proxy/Servers provide identical services.

Common Internet services provide differing strategies for advertising server addresses to clients. The strategy is conveyed through the DNS resource records returned in response to name resolution queries. As of January 2020 Internet-based 'nslookup' services were used to determine the following:

- When a client resolves the domainname "google.com", the DNS always returns one A record (i.e., an IPv4 address) and one AAAA record (i.e., an IPv6 address). The client receives the same addresses each time it resolves the domainname via the same DNS resolver, but may receive different addresses when it resolves the domainname via different DNS resolvers. But, in each case, exactly one A and one AAAA record are returned.

- When a client resolves the domainname "ietf.org", the DNS always returns one A record and one AAAA record with the same addresses regardless of which DNS resolver is used.

- When a client resolves the domainname "yahoo.com", the DNS always returns a list of 4 A records and 4 AAAA records. Each time the client resolves the domainname via the same DNS resolver, the same list of addresses are returned but in randomized order (i.e., consistent with a DNS round-robin strategy). But, interestingly, the same addresses are returned (albeit in randomized order) when the domainname is resolved via different DNS resolvers.

- When a client resolves the domainname "amazon.com", the DNS always returns a list of 3 A records and no AAAA records. As with "yahoo.com", the same three A records are returned from any worldwide Internet connection point in randomized order.

The above example strategies show differing approaches to Internet resilience and service distribution offered by major Internet services. The Google approach exposes only a single IPv4 and a single IPv6 address to clients. Clients can then select whichever IP protocol version offers the best response, but will always use the same IP address according to the current Internet connection point. This means that the IP address offered by the network must lead to a highly-available server and/or service distribution point. In other words, resilience is predicated on high availability within the network and with no client-initiated failovers expected (i.e., it is all-or-nothing from the client's perspective). However, Google does provide for worldwide distributed service distribution by virtue of the fact that each Internet connection point responds with a different IPv6 and IPv4 address. The IETF approach is like google (all-or-nothing from the client's perspective), but provides only a single IPv4 or IPv6 address on a worldwide basis. This means that the addresses must be made highly-available at the network level with

no client failover possibility, and if there is any worldwide service distribution it would need to be conducted by a network element that is reached via the IP address acting as a service distribution point.

In contrast to the Google and IETF philosophies, Yahoo and Amazon both provide clients with a (short) list of IP addresses with Yahoo providing both IP protocol versions and Amazon as IPv4-only. The order of the list is randomized with each name service query response, with the effect of round-robin load balancing for service distribution. With a short list of addresses, there is still expectation that the network will implement high availability for each address but in case any single address fails the client can switch over to using a different address. The balance then becomes one of function in the network vs function in the end system.

The same implications observed for common highly-available services in the Internet apply also to the AERO client/server architecture. When an AERO Client connects to one or more ANETs, it discovers one or more AERO Proxy/Server addresses through the mechanisms discussed in earlier sections. Each Proxy/Server address presumably leads to a fault-tolerant clustering arrangement such as supported by Linux-HA, Extended Virtual Synchrony or Paxos. Such an arrangement has precedence in common Internet service deployments in lightweight virtual machines without requiring expensive hardware deployment. Similarly, common Internet service deployments set service IP addresses on service distribution points that may relay requests to many different servers.

For AERO, the expectation is that a combination of the Google/IETF and Yahoo/Amazon philosophies would be employed. The AERO Client connects to different ANET access points and can receive 1-2 Proxy/Server ADM-LLAs at each point. It then selects one AERO Proxy/Server address, and engages in RS/RA exchanges with the same Proxy/Server from all ANET connections. The Client remains with this Proxy/Server unless or until the Proxy/Server fails, in which case it can switch over to an alternate Proxy/Server. The Client can likewise switch over to a different Proxy/Server at any time if there is some reason for it to do so. So, the AERO expectation is for a balance of function in the network and end system, with fault tolerance and resilience at both levels.

Appendix B. Change Log

<< RFC Editor - remove prior to publication >>

Changes from draft-templin-6man-aero-21 to draft-templin-6man-aero-22:

- That's it, folks.

Changes from draft-templin-6man-aero-20 to draft-templin-6man-aero-21:

- Major updates to Hub-and-Spokes Proxy/Server coordination.

Changes from draft-templin-6man-aero-19 to draft-templin-6man-aero-20:

- Major updates especially in Section 3.2.7.

Changes from draft-templin-6man-aero-18 to draft-templin-6man-aero-19:

- Major revision update for review.

Changes from draft-templin-6man-aero-17 to draft-templin-6man-aero-18:

- Interim version with extensive new text - cleanup planned for next release.

Changes from draft-templin-6man-aero-16 to draft-templin-6man-aero-17:

- Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-15 to draft-templin-6man-aero-16:

- Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-14 to draft-templin-6man-aero-15:

- Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-13 to draft-templin-6man-aero-14:

- o Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-12 to draft-templin-6man-aero-13:

- o Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-11 to draft-templin-6man-aero-12:

- o Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-10 to draft-templin-6man-aero-11:

- o Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-09 to draft-templin-6man-aero-10:

- o Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-08 to draft-templin-6man-aero-09:

- o Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-07 to draft-templin-6man-aero-08:

- o Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-06 to draft-templin-6man-aero-07:

- Final editorial review pass resulting in multiple changes. Document now submit for final approval (with reference to rfcdiff from previous version).

Changes from draft-templin-6man-aero-05 to draft-templin-6man-aero-06:

- Final editorial review pass resulting in multiple changes. Document now submit for final approval.

Changes from draft-templin-6man-aero-04 to draft-templin-6man-aero-05:

- Changed to use traffic selectors instead of the former multilink selection strategy.

Changes from draft-templin-6man-aero-03 to draft-templin-6man-aero-04:

- Removed documents from "Obsoletes" list.

- Introduced the concept of "secured" and "unsecured" spanning tree.

- Additional security considerations.

- Additional route optimization considerations.

Changes from draft-templin-6man-aero-02 to draft-templin-6man-aero-03:

- Support for extended route optimization from ROR to target over target's underlying interfaces.

Changes from draft-templin-6man-aero-01 to draft-templin-6man-aero-02:

- Changed reference citations to "draft-templin-6man-omni".

- Several important updates to IPv6 ND cache states and route optimization message addressing.

- Included introductory description of the "6M's".

- Updated Multicast specification.

Changes from draft-templin-6man-aero-00 to draft-templin-6man-aero-01:

- Changed category to "Informational".

- Updated implementation status.

Changes from earlier versions to draft-templin-6man-aero-00:

- Established working baseline reference.

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org