Network Working Group
Internet-Draft
Intended status: Informational
Expires: 15 April 2023

F. L. Templin, Ed.
Boeing Research & Technology

12 October 2022

Automatic Extended Route Optimization (AERO)
draft-templin-6man-aero-63

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 unique-local address format for IPv6 Neighbor Discovery (IPv6 ND) messaging over the OMNI virtual link. Router discovery and neighbor coordination are employed for network admission and to manage the OMNI link forwarding and routing systems. Secure multilink operation, mobility management, multicast, traffic path selection and route optimization are naturally supported through dynamic neighbor cache updates. AERO is a widely-applicable mobile internetworking service especially well-suited to aviation, 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 manifested by IPv6 encapsulation and configured over a network-of-networks concatenation of underlay Internetworks. Nodes on the link can exchange original IP packets or parcels [I-D.templin-intarea-parcels] as single-hop neighbors - both IP protocol versions (IPv4 and IPv6) are supported. The OMNI Adaptation Layer (OAL) supports multilink operation for increased reliability and path optimization while providing fragmentation and reassembly services to support improved performance and Maximum Transmission Unit (MTU) diversity. This specification provides a mobility service architecture companion to the OMNI specification.

The AERO service connects Hosts and Clients as OMNI link neighbors via Proxy/Servers and Relays as intermediate nodes as necessary; AERO further employs Gateways that interconnect diverse Internetworks as OMNI link segments through OAL forwarding at a layer below IP. Each node's OMNI interface uses an IPv6 unique-local address format that supports operation of the IPv6 Neighbor Discovery (IPv6 ND) protocol [RFC4861]. A Client's OMNI interface can be configured over multiple underlay 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 proxys and/or designated routers while fixed nodes may use any Relay on the link for efficient communications. Fixed nodes forward original IP packets/parcels 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 using stable IP Addresses not subject to dynamic fluctuations.

AERO Gateways peer with Proxy/Servers in a secured private BGP overlay routing instance to establish a Segment Routing Topology (SRT) spanning tree over the underlay Internetworks of one or more disjoint administrative domains concatenated 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 underlay Internetworks to (mobile or fixed) nodes on the OMNI link. From the perspective of underlay Internetworks, each Relay appears as the source of a route to the MSP; hence uplink traffic to mobile nodes 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 IPv4 and IPv6 Internets. In both cases, Clients may be located behind Network Address Translators (NATs) on the path to their associated Proxy/ Servers. A means for robust traversal of NATs while avoiding "triangle routing" and critical infrastructure traffic concentration through a service known as "route optimization" 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 "6 M's" of modern Internetworking, including:

1. Multilink - a mobile node's ability to coordinate multiple diverse underlay 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 administrative domain network segments while maintaining seamless end-to-end communications between mobile Clients and correspondents such as air traffic controllers, fleet administrators, other mobile Clients, 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 underlay 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/parcels of various robust sizes between peers without loss due to a link size restriction, and to dynamically adjust packet/parcel 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] in conjunction with the OMNI extensions specified in [I-D.templin-6man-omni].

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.

L3
The Network layer in the OSI network model. Also known as "layer-3", "IP layer", etc.

L2
The Data Link layer in the OSI network model. Also known as "layer-2", "link-layer", "sub-IP layer", etc.

Adaptation layer
A mid-layer that adapts L3 to a diverse collection of L2 underlay interfaces and their encapsulations. (No layer number is assigned, since numbering was an artifact of the legacy reference model that need not carry forward in the modern architecture.) The adaptation layer sees the upper layer as "L3" and sees all lower layer encapsulations as "L2 encapsulations", which may include UDP, IP and true link-layer (e.g., Ethernet, etc.) headers.

Access Network (ANET)
a connected network region (e.g., an aviation radio access network, satellite service provider network, cellular operator network, WiFi network, etc.) that joins Clients to the Mobility Service. Physical and/or data link level security is assumed, and sometimes referred to as "protected spectrum". Private enterprise networks and ground domain aviation service networks may provide multiple secured IP hops between the Client's point of connection and the nearest Proxy/Server.

Internetwork (INET)
a connected network region with a coherent IP addressing plan that provides transit forwarding services between ANETs and AERO/OMNI nodes that coordinate with the Mobility Service over unprotected media. No physical and/or data link level security is assumed, therefore security must be applied by upper layers. The global public Internet itself is an example.

End-user Network (ENET)
a simple or complex "downstream" network that travels with the Client as a single logical unit. The ENET could be as simple as a single link connecting a single Host, or as complex as a large network with many links, routers, bridges and Hosts. The ENET could also provide an "upstream" link in a recursively-descending chain of additional Clients and ENETs. In this way, an ENET of an upstream Client is seen as the ANET of a downstream Client.

{A,I,E}NET interface
a node's attachment to a link in an {A,I,E}NET.

underlay network/interface

an ANET/INET/ENET network/interface over which an OMNI interface is configured. The OMNI interface is seen as a network layer (L3) interface by the IP layer, and the OMNI adaptation layer sees the underlay interface as a data link layer (L2) interface. The underlay interface either connects directly to the physical communications media or coordinates with another node where the physical media is hosted.

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 underlay network 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 underlay network hops; AERO nodes can employ Segment Routing [RFC8402] to navigate between different OMNI links, and/or to cause packets/parcels to visit selected waypoints within the same OMNI link.

OMNI Adaptation Layer (OAL)

an OMNI interface sublayer service that encapsulates original IP packets/parcels admitted into the interface in an IPv6 header and/or subjects them to 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 an extended OMNI link.

OMNI Interface

a node’s attachment to an OMNI link (i.e., the same as defined in [I-D.templin-6man-omni]). 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].

(network) partition

frequently, underlay networks 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 partition is seen as a separate OMNI link segment as discussed throughout this document.)

(OMNI) L2 encapsulation

the OMNI protocol encapsulation of OAL packets/fragments in an outer header or headers to form carrier packets that can be routed within the scope of the local {A,I,E}NET underlay network partition. Common L2 encapsulation combinations include UDP/IP/ Ethernet, etc. using a port/protocol/type number for OMNI.

L2 address (L2ADDR)
an address that appears in the L2 encapsulation for an underlay interface and also in IPv6 ND message OMNI options. L2ADDR can be either an IP address for IP encapsulations or an IEEE EUI address [EUI] for direct data link encapsulation. (When UDP/IP encapsulation is used, the UDP port number is considered an ancillary extension of the IP L2ADDR.)

original IP packet/parcel
a whole IP packet/parcel 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/parcel encapsulated in an OAL IPv6 header before OAL fragmentation, or following OAL reassembly.

OAL fragment
a portion of an OAL packet following fragmentation but prior to L2 encapsulation, or following L2 decapsulation 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 packet/fragment following L2 encapsulation or prior to L2 decapsulation. OAL sources and destinations exchange carrier packets over underlay interfaces, and may be separated by one or more OAL intermediate nodes. OAL intermediate nodes re-encapsulate OAL packets/fragments during forwarding by removing the L2 headers of the previous hop underlay network and replacing them with new L2 headers for the next hop underlay network.

OAL source
an OMNI interface acts as an OAL source when it encapsulates original IP packets/parcels to form OAL packets, then performs OAL fragmentation and L2 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/parcel.

OAL intermediate node
an OMNI interface acts as an OAL intermediate node when it removes the L2 headers of carrier packets received from a previous hop, then re-encapsulates the enclosed OAL packets/fragments in new L2 headers and sends these new carrier packets to the next hop. OAL intermediate nodes decrement the OAL Hop Limit during forwarding, and discard the OAL packet/fragment if the Hop Limit reaches 0. OAL intermediate nodes do not decrement the TTL/Hop Limit of the original IP packet/parcel.

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.

Interface Identifier (IID)
the least significant 64 bits of an IPv6 address, as specified in the IPv6 addressing architecture [RFC4291].

Link Local Address (LLA)
an IPv6 address beginning with fe80::/64 per the IPv6 addressing architecture [RFC4291] and with either a 64-bit MNP (LLA-MNP) or a 56-bit random value (LLA-RND) encoded in the IID as specified in [I-D.templin-6man-omni].

Unique Local Address (ULA)
an IPv6 address beginning with fd00::/8 followed by a 40-bit Global ID followed by a 16-bit Subnet ID per [RFC4193] and with either a 64-bit MNP (ULA-MNP) or a 56-bit random value (ULA-RND) encoded in the IID as specified in [I-D.templin-6man-omni]. (Note that [RFC4193] specifies a second form of ULAs based on the prefix fc00::/8, which are referred to as "ULA-C" throughout this document to distinguish them from the ULAs defined here.)

Temporary Local Address (TLA)

a ULA beginning with fd00::/16 followed by a 48-bit randomly-initialized value followed by an MNP-based (TLA-MNP) or random (TLA-RND) IID as specified in [I-D.templin-6man-omni]. Clients use TLAs to bootstrap autoconfiguration in the presence of OMNI link infrastructure or for sustained communications in the absence of infrastructure. (Note that in some environments Clients can instead use a (Hierarchical) Host Identity Tag ((H)HIT) instead of a TLA - see: [I-D.templin-6man-omni].)

eXtended Local Address (XLA)

a ULA beginning with fd00::/64 followed by an MNP-based (XLA-MNP) or random (XLA-RND) IID as specified in [I-D.templin-6man-omni]. An XLA can be used to supply a stable address for IPv6 ND messaging, a routing table entry for the OMNI link routing system, etc. (Note that XLAs can also be statelessly formed from LLAs (and vice-versa) simply by inverting prefix bits 7 and 8.)

AERO node

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

AERO Host ("Host")

an AERO node that configures an OMNI interface over an ENET underlying interface serviced by an upstream Client. The Host does not assign an LLA or ULA to the OMNI interface, but instead assigns the address taken from the ENET underlying interface. When an AERO host forwards an original IP packet/parcel to another AERO node on the same ENET, it uses simple IP-in-L2 OMNI encapsulation without including an OAL encapsulation header. The Host is therefore an OMNI link termination endpoint. (Note: as an implementation matter, the Host may instead configure the "OMNI interface" as a virtual sublayer of the underlay interface itself.)

AERO Client ("Client")

an AERO node that configures an OMNI interface over one or more underlay interfaces and requests MNP delegation/registration service from AERO Proxy/Servers. The Client assigns an XLA-MNP (as well as Proxy/Server-specific ULA-MNPs) to the OMNI interface for use in IPv6 ND exchanges with other AERO nodes and forwards original IP packets/parcels to correspondents according to OMNI interface neighbor cache state. The Client coordinates with Proxy/Servers and/or other Clients over upstream ANET/INET interfaces and may also provide Proxy/Server services for Hosts and/or other Clients over downstream ENET interfaces.

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 designated router services for coordination with correspondents on its INET-facing interfaces. (Proxy/Servers in the open INET instead configure only a single INET interface and no ANET interfaces.) The Proxy/Server configures an OMNI interface and assigns a ULA-RND to support the operation of IPv6 ND services, while advertising any associated MNPs for which it is acting as a hub via BGP peerings with AERO Gateways.

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 assign a ULA-RND to an OMNI interface and maintain BGP peerings with Gateways the same as Proxy/Servers. Relays also run a dynamic routing protocol to discover any non-MNP IP GUA routes in service on other links/networks, advertise OMNI link MSP(s) to other links/networks, and redistribute routes discovered on other links/networks into the OMNI link BGP routing system. (Relays that connect to major Internetworks such as the global IPv6 or IPv4 Internet can also be configured to advertise "default" routes into the OMNI link BGP routing system.)

AERO Gateway ("Gateway")

a BGP hub autonomous system node that also provides OAL forwarding services for nodes on an OMNI link. Gateways forward OAL packets/ fragments 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. Gateways peer with Proxy/Servers and other Gateways to form an IPv6-based OAL spanning tree over all OMNI link segments and to discover the set of all MNP and non-MNP prefixes in service. Gateways process OAL packets/fragments received over the secured spanning tree that are addressed to themselves, while forwarding all other OAL packets/ fragments to the next hop also via the secured spanning tree. Gateways forward OAL packets/fragments 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.

First-Hop Segment (FHS) Client

a Client that initiates communications with a target peer by sending an NS message to establish reverse-path multilink forwarding state in OMNI link intermediate nodes on the path to the target. Note that in some arrangements the Client’s (FHS) Proxy/Server (and not the Client itself) initiates the NS.

Last-Hop Segment (LHS) Client

a Client that responds to a communications request from a source peer's NS by returning an NA response to establish forward-path multilink forwarding state in OMNI link intermediate nodes on the path to the source. Note that in some arrangements the Client's (LHS) Proxy/Server (and not the Client itself) returns the NA.

First-Hop Segment (FHS) Proxy/Server

a Proxy/Server for an FHS Client's underlay interface that forwards the Client's OAL packets into the segment routing topology. FHS Proxy/Servers also 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 underlay interface of an LHS Client that forwards OAL packets received from the segment routing topology to the Client over that interface.

Hub Proxy/Server

a single Proxy/Server selected by a Client that injects the Client's XLA-MNP into the BGP routing system and provides a designated router service for all of the Client's underlay interfaces. Clients often 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 as a Hub), however the Client can also select any available Proxy/Server for the OMNI link (as there is no requirement that the Hub must also be one of the Client's FHS Proxy/Servers).

Segment Routing Topology (SRT)

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

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/parcels between any downstream-attached networks and the OMNI link. The MR is the MN entity that hosts the AERO Client.

Address Resolution Source (ARS)
the node nearest the original source that initiates OMNI link address resolution. The ARS may be a Proxy/Server or Relay for the source, or may be the source Client itself. The ARS is often (but not always) also the same node that becomes the FHS source during route optimization.

Address Resolution Target (ART)
the node toward which address resolution is directed. The ART may be a Relay or the target Client itself. The ART is often (but not always) also the same node that becomes the LHS target during route optimization.

Address Resolution Responder (ARR)
the node that responds to address resolution requests on behalf of the ART. The ARR may be a Relay, the ART itself, or the ART's current Hub Proxy/Server. Note that a Hub Proxy/Server can assume the ARR role even if it is located on a different SRT segment than the ART. The Hub Proxy/Server assumes the ARR role only when it receives an RS message from the ART with the 'A' flag set (see: [I-D.templin-6man-omni]).

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 Gateways that tracks all Proxy/Server-to-Client associations.

Mobility Service (MS)
the collective set of all Proxy/Servers, Gateways and Relays that provide the AERO Service to Clients.

AERO Forwarding Information Base (AFIB)
A forwarding table on each OAL source, destination and intermediate node that includes AERO Forwarding Vectors (AFV) with both multilink forwarding instructions and context for reconstructing compressed headers for specific communicating peer underlay interface pairs. The AFIB also supports route optimization where one or more OAL intermediate nodes in the path can be "skipped" to reduce path stretch and decrease load on critical infrastructure elements.

AERO Forwarding Vector (AFV)
An AFIB entry that includes soft state (including addressing and Identification information) for each underlay interface pairwise communication session between peer OAL nodes. AFVs are identified by both a forward and reverse path AFV Index (AFVI). OAL nodes establish reverse path AFVIs when they forward an IPv6 ND unicast Neighbor Solicitation (NS) message and establish forward path AFVIs when they forward the solicited IPv6 ND unicast Neighbor Advertisement (NA) response.

AERO Forwarding Vector Index (AFVI)
A locally-unique 4 octet value automatically generated by an OAL node when it creates an AFV. OAL intermediate nodes assign two distinct AFVIs (called "A" and "B") to each AFV, with "A" representing the forward path and "B" representing the reverse path. Meanwhile, the OAL source assigns a single "B" AFVI, and the OAL destination assigns a single "A" AFVI. Each OAL node advertises its "A" AFVI to previous hop nodes on the reverse path toward the source and advertises its "B" AFVI to next hop nodes on the forward path toward the destination.

AERO Forwarding Parameters (AFP)
An OMNI option sub-option that appears in IPv6 ND NS/NA messages and includes all parameters necessary for establishing AFV state in OAL nodes in the path (see: [I-D.templin-6man-omni]).

Throughout the document, the simple terms "Host", "Client", "Proxy/Server", "Gateway" and "Relay" refer to "AERO/OMNI Host", "AERO/OMNI Client", "AERO/OMNI Proxy/Server", "AERO/OMNI Gateway" and "AERO/OMNI Relay", respectively. Capitalization is used to distinguish these terms from other common Internetworking uses in which they appear without capitalization, and implies that the node in question both configures an OMNI interface and engages the OMNI Adaptation Layer.

The terminology of IPv6 ND [RFC4861], DHCPv6 [RFC8415] and OMNI [I-D.templin-6man-omni] (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 Hosts configure an OMNI interface over an underlay interface connected to a Client's ENET and coordinate with both other AERO Hosts and Clients over the ENET. As an implementation matter, the Host either assigns the same (MNP-based) IP address from the underlay interface to the OMNI interface, or configures the "OMNI interface" as a virtual sublayer of the underlay interface itself. AERO Hosts treat the ENET as an ANET, and treat the upstream Client for the ENET as a Proxy/Server. AERO Hosts are seen as OMNI link termination endpoints.

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 underlay 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 ENETs (which may connect AERO Hosts and other Clients). AERO Clients provide Proxy/Server-like services for Hosts and other Clients on downstream-attached ENETs.

AERO Gateways, 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 and Hosts. AERO Gateways (together with Proxy/Servers) provide the secured backbone supporting infrastructure for a Segment Routing Topology (SRT) spanning tree for the OMNI link.

AERO Gateways forward 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 an adaptation layer forwarding service that upper layers perceive as L2 bridging, since the inner IP TTL/Hop Limit is not decremented. Each Gateway also peers with Proxy/Servers and other Gateways 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). Gateways assign one or more Mobility Service Prefixes (MSPs) to the OMNI link and configure secured tunnels with Proxy/Servers, Relays and other Gateways; they further maintain forwarding table entries for each MNP or non-MNP prefix in service on the OMNI link.

AERO Proxy/Servers distributed across one or more SRT segments provide default forwarding and mobility/multilink services for AERO Client mobile nodes. Each Proxy/Server also peers with Gateways in an adaptation layer 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 primary forwarding service for ANET Clients/Host communications with peers in external INETs, while Proxy/Servers in open INETs provide an authentication service IPv6 ND messages but should be used only a last resort data plane forwarding service when a Client cannot forward directly to an INET peer or Gateway. 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/parcels between the OMNI link and nodes on other links/networks. Relays run a dynamic routing protocol to discover any non-MNP prefixes in service on other links/networks, and Relays that connect to larger Internetworks (such as the Internet) may originate default routes. The Relay redistributes OMNI link MSP(s) into other links/networks, and redistributes non-MNP prefixes via OMNI link Gateway 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 Gateway G1 |
                         | Nbr: S1, S2, P1 |
                         |(X1->S1; X2->S2) |
                         |      MSP M1     |
                         +--------+--------+
       +--------------+           |            +--------------+
       |  AERO P/S S1 |           |            |  AERO P/S S2 |
       |  Nbr: C1, G1 |           |            |  Nbr: C2, G1 |
       |  default->G1 |           |            |  default->G1 |
       |    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   )-.   |  AERO |     |  AERO |    .-(_  IP   )-.
   (__    ENET     )--|Host H1|     |Host H2|--(__    ENET     )
      `-(______)-'    +-------+     +-------+     `-(______)-'
```

Figure 1: AERO/OMNI Reference Model

In this model:

- the OMNI link is an overlay network service configured over one or more underlay SRT segments which may be managed by diverse administrative domains using incompatible protocols and/or addressing plans.

- AERO Gateway G1 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). Gateways provide the backbone for an SRT spanning tree for the OMNI link.

* AERO Proxy/Servers S1 and S2 configure secured tunnels with Gateway G1 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 ENETs.

* AERO Hosts H1 and H2 attach to the ENETs served by Clients C1 and C2, respectively.

An OMNI link configured over a single underlay network appears as a single unified link with a consistent addressing plan; all nodes on the link can exchange carrier packets via simple L2 encapsulation (i.e., following any necessary NAT traversal) since the underlay is connected. In common practice, however, OMNI links are often configured over an SRT spanning tree that bridges multiple distinct underlay network segments managed under different administrative authorities (e.g., as for worldwide aviation service providers such as ARINC, SITA, Inmarsat, etc.). Individual underlay networks 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 Gateways.

The OMNI link spans multiple SRT segments 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/parcel. An example OMNI link SRT is shown in Figure 2:

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

Figure 2: OMNI Link Segment Routing Topology (SRT)

Gateway, Proxy/Server and Relay OMNI interfaces are configured over both secured tunnels and open INET underlay interfaces within their respective SRT segments. Within each segment, Gateways configure "hub-and-spokes" BGP peerings with Proxy/Servers and Relays as "spokes". Adjacent SRT segments are joined by Gateway-to-Gateway peerings to collectively form a spanning tree over the entire SRT. The "secured" spanning tree supports authentication and integrity for critical control plane messages (and any trailing data plane message extensions). 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.

The AERO Multinet service concatenates SRT segments to form a larger network through Gateway-to-Gateway peerings as originally suggested in the "Catenet Model for Internetworking" [IEN48]; especially Figure 2 follows directly from the illustrations in [IEN48-2]. The

Catenet concept suggested a "network-of-networks" concatenation of independent and diverse Internetwork "segments" to form a much larger network supporting end-to-end services.

The Catenet concept first articulated in the 1970’s was distorted through the evolution of the Internet in the decades that followed, since a critical element was missing from the architecture. As a result, the Internet evolved as a single, large public routing and addressing domain interconnecting private domains (i.e., instead of a true network-of-networks) which has impeded flexibility and inhibited end-to-end services. With the advent of the AERO/OMNI adaptation layer, however, the original Catenet "network-of-networks" vision is restored.

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 to the OMNI interface to satisfy the requirements of [RFC4861]. AERO Clients configure LLAs constructed from MNPs (i.e., "LLA-MNPs") while AERO infrastructure nodes construct LLAs based on 56-bit random values ("LLA-RNDs") per [I-D.templin-6man-omni]. Non-MNP routes are also represented the same as for MNPs, 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 Global ID to form the prefix {ULA}::/48, then include a 16-bit Subnet ID ’*’ to form the prefix {ULA*}::/64 [RFC4291]. The AERO node then uses the prefix {ULA*}::/64 to form "ULA-MNPs" or "ULA-RNDs" as specified in [I-D.templin-6man-omni] to support OAL addressing. (The prefix {ULA*}::/64 appearing alone and with no suffix represents "default" for that prefix.)

AERO Clients also use Temporary Local Addresses (TLAs) and eXtended Local Addresses (XLAs) constructed per [I-D.templin-6man-omni], where TLAs are distinguished from ordinary ULAs based on the prefix fd00::/16 and XLAs are distinguished from ULAs/TLAs based on the prefix fd00::/64. Clients use TLA-RNDs only in initial control message exchanges until a stable MNP is assigned, but may sometimes also use them for sustained communications within a local routing region. AERO nodes use XLA-MNPs to provide forwarding information for the global routing table as well as IPv6 ND message addressing information.

AERO MSPs, MNPs and non-MNP routes are typically based on Global Unicast Addresses (GUAs), but in some cases may be based on IPv4 private addresses [RFC1918] or IPv6 ULA-C’s [RFC4193]. A GUA block is also reserved for OMNI link anycast purposes. See [I-D.templin-6man-omni] for a full specification of LLAs, ULAs, TLAs, XLAs, GUAs and anycast addresses 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 Gateways and Proxy/Servers (Relays also engage in the routing system as simplified Proxy/Servers). The service supports OAL packet/fragment 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 connected 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 Gateways but does not peer with other Proxy/Servers. Each SRT segment in the OMNI link must include one or more Gateways in a "hub" AS, which peer with the Proxy/Servers within that segment as "spoke" ASes. All Gateways 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 Gateways of different segments peer with one another using eBGP.

Gateways maintain forwarding table entries only for ULA prefixes for infrastructure elements and XLA-MNPs corresponding to MNP and non-MNP routes that are currently active; Gateways also maintain black-hole routes for the OMNI link MSPs so that OAL packets/fragments destined to non-existent more-specific routes are dropped with a Destination Unreachable message returned. 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 OAL packets/fragments to Gateways which have full topology knowledge.

Each OMNI link segment assigns a unique sub-prefix of {ULA}::/48 known as the "SRT prefix". For example, a first segment could assign {ULA}:1000::/56, a second could assign {ULA}:2000::/56, a third could assign {ULA}:3000::/56, etc. Within each segment, each Proxy/Server configures a ULA-RND within the segment's SRT prefix with a 56-bit random value in the interface identifier as specified in [I-D.templin-6man-omni].

The administrative authorities for each segment must therefore coordinate to assure mutually-exclusive ULA prefix assignments, but internal provisioning of ULAs is an independent local consideration for each administrative authority. For each ULA prefix, the Gateway(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::/64 is simply {ULA}:1023::/64.

ULA prefixes are statically represented in Gateway forwarding tables. Gateways 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 ULA prefixes either via standard BGP routing or static routes. For example, if three Gateways ('A', 'B' and 'C') from different segments serviced {ULA}:1000::/56, {ULA}:2000::/56 and {ULA}:3000::/56 respectively, then the forwarding tables in each Gateway appear as follows:

A: {ULA}:1000::/56->local, {ULA}:2000::/56->B, {ULA}:3000::/56->C

B: {ULA}:1000::/56->A, {ULA}:2000::/56->local, {ULA}:3000::/56->C

C: {ULA}:1000::/56->A, {ULA}:2000::/56->B, {ULA}:3000::/56->local

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

MNP (and non-MNP) routes are instead dynamically advertised in the AERO routing system by Proxy/Servers and Relays that provide service for their corresponding MNPs. The routes are advertised as XLA-MNP prefixes, i.e., as fd00::{MNP} (see: [I-D.templin-6man-omni]). 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: fd00::2001:db8:1000:2000/120

E: fd00::2001:db8:3000:4000/120

F: fd00::2001:db8:5000:6000/120

Note that the MNP length found in OMNI Prefix Length sub-option encodes a value between 1 and 64, but the corresponding XLA-MNP is entered into the routing system with length (64 + MNP length). A full discussion of the BGP-based routing system used by AERO is found in [I-D.ietf-rtgwg-atn-bgp].

3.2.4. Segment Routing Topologies (SRTs)

The distinct {ULA}::/48 prefixes in an OMNI link domain identify 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 the 40-bit ULA Global ID field 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 Gateways and Proxy/Servers of each independent SRT engage in BGP peerings to form a spanning tree with the Gateways in non-leaf nodes and the Proxy/Servers in leaf nodes. The spanning tree is configured over both secured and unsecured underlay network paths. The secured spanning tree is used to convey secured control messages (and sometimes data message extensions) between Proxy/Servers and Gateways, while the unsecured spanning tree forwards bulk data messages and/or unsecured control messages.

Each SRT segment is identified by a unique ULA prefix used by all Proxy/Servers and Gateways 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.

Note: The distinct ULA prefixes in an OMNI link domain can be carried either in a common BGP routing protocol instance for all OMNI links or in distinct BGP routing protocol instances for different OMNI links. In some SBM environments, such separation may be necessary to ensure that distinct OMNI links do not include any common infrastructure elements as single points of failure. In other environments, carrying the ULAs of multiple OMNI links within a common routing system may be acceptable.

3.2.5. Segment Routing For OMNI Link Selection

Original IPv6 sources can direct IPv6 packets/parcels 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/parcel will traverse when there may be multiple alternatives.

When an AERO node processes the SRH and forwards the original IPv6 packet/parcel 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).

3.3. OMNI Interface Characteristics

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

* ANET interfaces connect to a protected and secured ANET that is separated from open INETs by Proxy/Servers. The ANET interface may be either on the same L2 link segment as a Proxy/Server, or separated from a Proxy/Server by multiple IP hops. (Note that NATs may appear internally within an ANET and may require NAT traversal on the path to the Proxy/Server the same as for the INET case.)

* INET interfaces connect to an INET either natively or through one or several IPv4 Network Address Translators (NATs). Native INET interfaces have global IP addresses that are reachable from correspondent on the same INET. NATed INET interfaces typically have private IP addresses and connect to a private network behind one or more NATs with the outermost NAT providing INET access.

* ENET interfaces connect a Client’s downstream-attached networks, where the Client provides forwarding services for ENET Host and Client communications to remote peers. An ENET can be as simple as a small stub network that travels with a mobile Client (e.g., an Internet-of-Things) to as complex as a large private enterprise network that the Client connects to a larger ANET or INET.

* VPNed interfaces use security encapsulation over an underlay network to a Client or Proxy/Server acting as a Virtual Private Network (VPN) gateway. Other than the link-layer encapsulation format, VPNed interfaces behave the same as for Direct interfaces.

* Direct (aka "point-to-point") interfaces connect directly to a Client or Proxy/Server without engaging any forwarding devices in the path. An example is a line-of-sight link between a remote pilot and an unmanned aircraft.

OMNI interfaces use OAL encapsulation and fragmentation as discussed in Section 3.6. OMNI interfaces use L2 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 use link-layer encapsulation only (i.e., and no other L2 encapsulations) over Direct underlay interfaces or ANET interfaces when the Client and FHS Proxy/Server are known to be on the same underlay 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), Neighbor Advertisement (NA) and Redirect for neighbor cache management. In environments where spoofing may be a threat, OMNI neighbors should invoke 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, Interface Attributes and/or AERO Forwarding Parameters (AFPs) containing link information parameters for the OMNI interface’s underlay interfaces and any other per-neighbor information.

A Host’s OMNI interface is configured over an underlay interface connected to an ENET provided by an upstream Client. From the Host’s perspective, the ENET appears as an ANET and the upstream Client appears as a Proxy/Server. The Host does not provide OMNI intermediate node services and is therefore a logical termination point for the OMNI link.

A Client’s OMNI interface may be configured over multiple ANET/INET underlay 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 ANET/INET underlay 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 Interface Attributes, Traffic Selector and/or AFP sub-options with the same underlay interface ifIndex. In that case, the Client would appear to have a single underlay interface but with a dynamically changing link-layer address.

If the Client has multiple active ANET/INET underlay 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 sub-options with different underlay interface ifIndexes.

Proxy/Servers on the open Internet include only a single INET underlay interface. INET Clients therefore discover only the L2ADDR information for the Proxy/Server's INET interface. Proxy/Servers on an ANET/INET boundary include both an ANET and INET underlay interface. ANET Clients therefore must discover both the ANET and INET L2ADDR information for their Proxy/Servers.

Gateway and Proxy/Server OMNI interfaces are configured over underlay 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 a ULA-RND and acts as an OAL source to encapsulate original IP packets/parcels, then fragments the resulting OAL packets, performs L2 encapsulation and sends the resulting carrier packets over the secured or unsecured underlay paths. Note that Gateway and Proxy/Server end-to-end transport protocol sessions used by the BGP run directly over the OMNI interface and use ULA-RND source and destination addresses. The ULA-RND addresses that appear in BGP protocol session original IP packets/parcels may therefore be the same as those that appear in the OAL IPv6 encapsulation header.

3.4. OMNI Interface Initialization

AERO Proxy/Servers, Clients and Hosts 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/parcels 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 Hosts and Gateways 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 a ULA-RND appropriate for the given OMNI link SRT segment. The Proxy/Server also configures secured underlay interface tunnels and engages in BGP routing protocol sessions over the OMNI interface with one or more neighboring Gateways.

The OMNI interface provides a single interface abstraction to the IP layer, but internally serves as an NBMA nexus for sending carrier packets to OMNI interface neighbors over underlay interfaces and/or 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/ENET interfaces (see: Section 3.2.3). The Relay provisions MNPs to networks on the INET/ENET interfaces (i.e., the same as a Client would do) and advertises the MSP(s) for the OMNI link over the INET/ENET interfaces. The Relay further provides an OMNI link attachment point for non-MNP-based topologies.

3.4.2. AERO Client Behavior

When a Client enables an OMNI interface, it assigns either an XLA-MNP or a TLA and sends OMNI-encapsulated RS messages over its ANET/INET underlay interfaces to an FHS Proxy/Server, which coordinates with a Hub Proxy/Server that returns an RA message with corresponding parameters. The RS/RA messages may pass through one or more NATs in the path between the Client and FHS Proxy/Server. (Note: if the Client used a TLA in its initial RS messages, it may discover ULA-MNPs in the corresponding RAs that it receives from FHS Proxy/Servers 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 TLAs for Client-to-Client communications at least until it encounters an infrastructure element that can delegate MNPs.)

A Client can further extend the OMNI link over its (downstream) ENET interfaces where it provides a first-hop router for Hosts and other AERO Clients connected to the ENET. A downstream Client that connects via the ENET serviced by an upstream Client can in turn service further downstream ENETs that connect other Hosts and Clients. This OMNI link extension can be applied recursively over a "chain" of ENET Clients.

3.4.3. AERO Host Behavior

When a Host enables an OMNI interface, it assigns an address taken from the ENET underlay interface which may itself be a GUA delegated by the upstream Client. The Host does not assign a link-local address to the OMNI interface, since no autoconfiguration is necessary on that interface. (As an implementation matter, the Host could instead configure the "OMNI interface" as a virtual sublayer of the ENET underlay interface itself.)

The Host sends OMNI-encapsulated RS messages over its ENET underlay interface to the upstream Client, which returns encapsulated RAs and provides routing services in the same fashion that Proxy/Servers provides services for Clients. Hosts represent the leaf end systems in recursively-nested chain of concatenated ENETs, i.e., they represent terminating endpoints for the OMNI link.

3.4.4. AERO Gateway Behavior

AERO Gateways configure an OMNI interface and assign a ULA-RND and corresponding Subnet Router Anycast address for each of their OMNI link SRT segments. Gateways configure underlay interface secured tunnels with Proxy/Servers in the same SRT segment and other Gateways in the same (or an adjacent) SRT segment. Gateways 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 Client, Proxy/Server and Gateway OMNI interface maintains a network layer conceptual neighbor cache per [RFC1256] or [RFC4861] the same as for any IP interface. The OMNI interface network layer neighbor cache is maintained through static and/or dynamic neighbor cache entry configurations.

Each OMNI interface also maintains a separate internal adaptation layer conceptual neighbor cache that includes a Neighbor Cache Entry (NCE) for each of its active OAL neighbors per [RFC4861]. Throughout this document, the terms "neighbor cache" and "NCE" refer to this adaptation layer neighbor cache unless otherwise specified.

Each OMNI interface NCE is indexed by the ULA of the neighbor found in the ND message IPv6 header and determines the context for Identification verification. Clients and Proxy/Servers maintain NCEs through dynamic RS/RA message exchanges, and also maintain NCEs for any active correspondent peers through dynamic NS/NA message exchanges.

Hosts also maintain NCEs for Clients and other Hosts through the exchange of RS/RA, NS/NA or Redirect messages. Each NCE is indexed by the IP address assigned to the Host ENET interface, which is the same address used for L2 encapsulation (i.e., without the insertion of an OAL header). This encapsulation format identifies the NCE as a Host-based entry where the Host is a leaf end system in the recursively extended OMNI link.

Gateways also maintain NCEs for Clients within their local segments based on NS/NA route optimization messaging (see: Section 3.13.3). When a Gateway creates/updates a NCE for a local segment Client based on NS/NA route optimization, it also maintains AFIB state for messages destined to this local segment Client.

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 OAL packets/fragments destined to the target Client to the Client's new FHS/Hub Proxy/Server instead. It is RECOMMENDED that DEPART_TIME be set to the default constant value 10 seconds to accept any carrier packets that may be in flight. When DepartTime decrements to 0, the NCE is deleted.

Clients determine the service profiles for their FHS and Hub Proxy/ Servers by setting the N/A/U flags in RS messages and also by setting/clearing the FMT-Forward and FMT-Mode flags in the Interface Attributes sub-option. When the N/A/U flags are clear, Proxy/Servers forward all NS/NA messages to the Client, while the Client performs mobility update signaling through the transmission of uNA messages to all active neighbors following a mobility event. However, in some environments this may result in excessive NS/NA control message overhead especially for Clients connected to low-end data links.

Clients can set the N/A/U flags in RS messages they send to select their service profiles. If the N flag is set, the FHS Proxy/Server that forwards the RS message assumes the role of responding to NS messages and maintains peer NCEs associated with the NCE for this Client. If the A flag is set, the Hub Proxy/Server that processes the RS message assumes the role of responding to NS(AR) messages on behalf of this Client NCE. If the U flag is set, the Hub Proxy/ Server that processes the RS message becomes responsible for maintaining a "Report List" of sources/targets for NS(AR) messages it forwards on behalf of this Client NCE. The Hub Proxy/Server maintains each Report List entry for REPORT_TIME seconds, and sends uNA messages to each member of the Report List when it receives a Client mobility update indication (e.g., through receipt of an RS with updated Interface Attributes and/or Traffic Selectors).

Clients can also set/clear the FMT-Forward and FMT-Mode flags in the Interface Attributes sub-option of each RS message to express their desired service profile from each FHS Proxy/Server. The FHS Proxy/ Server will consider the Client’s preferences and either accept or override by setting/clearing the flags in the corresponding RA message reply. Implications for these bits are discussed in [I-D.templin-6man-omni].

Both the Client and its Hub Proxy/Server have full knowledge of the Client’s current underlay Interface Attributes and Traffic Selectors, while FHS Proxy/Servers acting in "proxy" mode have knowledge of only the individual Client underlay interfaces they service. Clients determine their FHS and Hub Proxy/Server service models by setting the N/A/U flags in the RS messages they send as discussed above.

When an Address Resolution Source (ARS) sends an NS(AR) message toward an Address Resolution Target (ART) Client/Relay, the OMNI link routing system directs the NS(AR) to a Hub Proxy/Server for the ART. The Hub then either acts as an Address Resolution Responder (ARR) on behalf of the ART or forwards the NS(AR) to the ART which acts as an ARR on its own behalf. The ARR returns an NA(AR) response to the ARS, which creates or updates a NCE for the ART while caching L3 and L2 addressing information. The ARS then (re)sets ReachableTime for the NCE to REACHABLE_TIME seconds and performs unicast NS/NA exchanges over specific underlay interface pairs to determine paths for sending carrier packets directly to the ART. The ARS 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 use ULAs instead of LLAs as IPv6 ND message source and destination addresses. This allows multiple different OMNI links to be joined into a single link at some future time without requiring a global renumbering event.

For each IPv6 ND message, the OMNI interface includes one or more OMNI options (and any other ND message options) then completely populates all option information. If the OMNI interface includes an authentication option, it first writes the value 0 into the authentication signature field then calculates the signature beginning with the first IPv6 ND message octet following the header Checksum field and continuing over the entire length of the packet or super-packet. The OMNI interface next writes the authentication signature value into the appropriate OMNI authentication option field, then calculates the IPv6 ND message checksum per [RFC4443] beginning with a pseudo-header of the IPv6 header and writes the value into the Checksum field. The IPv6 ND message checksum therefore provides integrity assurance for the message, while the authentication signature covers the entire packet or super-packet. OMNI interfaces verify integrity and authentication of each packet or super-packet received, and process the message further only following successful verification.

OMNI options include per-neighbor information that provides multilink forwarding, link-layer address and traffic selector information for the neighbor’s underlay interfaces. This information is stored in both the neighbor cache and AERO Forwarding Information Base (AFIB) as 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.

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 OMNI options and S/TLLAOs appear, the former pertains to adaptation layer to underlay interface address mappings while the latter pertains to the native L2 address format of the underlay media.

OMNI interface IPv6 ND messages may also include other IPv6 ND options. In particular, solicitation messages may include a Nonce option 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 recipient must also include a Timestamp option in its advertisement reply. All unsolicited advertisement and redirect messages should include a Timestamp option.

AERO Clients send RS messages to the link-scoped All-Routers multicast address or a ULA-RND while using unicast or anycast OAL/L2 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 Hosts and Clients on ENET underlay networks send RS messages to the link-scoped All-Routers multicast address, a ULA-RND of a remote Hub Proxy/Server or the ULA-MNP of an upstream Client while using unicast or anycast OAL/L2 addresses. The upstream AERO Client responds by returning a unicast RA message.

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

* NS/NA(AR) messages are used for address resolution and optionally to establish sequence number windows. The ARS sends an NS(AR) to the solicited-node multicast address of the ART, and an ARR with addressing information for the ART returns a unicast NA(AR) that contains current, consistent and authentic target address resolution information. NS(AR) messages include a solicited-node multicast destination address to distinguish them from ordinary NS messages. NS/NA(AR) messages must be secured.

* Ordinary NS/NA messages are used determine target reachability, establish and maintain NAT state, and/or establish AFIB state. The source sends an NS to the unicast address of the target while optionally including an OMNI AERO Forwarding Parameters (AFP) sub-option naming a specific underlay interface pair, and the target returns a unicast NA that includes a responsive AFP if necessary.

NS/NA messages that use an in-window sequence number and do not update any other state need not include an authentication signature but must include an IPv6 ND message checksum. NS/NA messages used to establish window synchronization and/or AFIB state 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 update state information must be secured.

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

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

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 forwarding nodes on the OMNI link. 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 ARR 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 XLA-MNPs must be uniquely assigned to Clients to support correct IPv6 ND protocol operation, however, no role is currently seen for assigning the same XLA-MNP 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 by including OMNI Window Synchronization sub-options in RS/RA or NS/NA message exchanges to maintain send/receive window state in their respective neighbor cache and AFIB entries as specified in [I-D.templin-6man-omni].

3.6. OMNI Interface Encapsulation and Fragmentation

When the network layer forwards an original IP packet/parcel into an OMNI interface, the interface locates or creates a Neighbor Cache Entry (NCE) that matches the destination. The OMNI interface then invokes the OMNI Adaptation Layer (OAL) as discussed in [I-D.templin-6man-omni] which encapsulates the packet/parcel in an IPv6 header to produce an OAL packet. For example, an original IP packet/parcel with source address 2001:db8:1:2::1 and destination address 2001:db8:1234:5678::1 might cause the OAL encapsulation header to include source address {XLA*}::2001:db8:1:2 (i.e., an XLA-MNP) and destination address {ULA*}::0012:3456:789a:bcde (i.e., a ULA-RND).

Following encapsulation, the OAL source then calculates a 2-octet OAL checksum, then fragments the OAL packet while including an identical Identification value for each fragment that must be within the window for the neighbor. The OAL source then appends the checksum as the final 2 octets of the final fragment, i.e., as a "trailer".

The OAL source next includes an identical Compressed Routing Header with 32-bit ID fields (CRH-32) [I-D.bonica-6man-comp-rtg-hdr] with each fragment containing one or more AERO Forwarding Vector Indices (AFVIs) if necessary as discussed in Section 3.13. The OAL source can instead invoke OAL header compression by replacing the OAL IPv6 header, CRH-32 and Fragment Header with an OAL Compressed Header (OCH).

The OAL source finally encapsulates each resulting OAL fragment in L2 headers to form a carrier packet, with source address set to its own L2 address (e.g., 192.0.2.100) and destination set to the L2 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:

```
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |           L2 Headers          |
        |       src = 192.0.2.100       |
        |        dst = 192.0.2.1        |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |        OAL IPv6 Header        |
        |  src = {XLA*}::2001:db8:1:2   |
        |dst={ULA*}::0012:3456:789a:bcde|
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |      CRH-32 (if necessary)    |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |      OAL Fragment Header      |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |       Original IP Header      |
        |     (first-fragment only)     |
        |    src = 2001:db8:1:2::1      |
        |  dst = 2001:db8:1234:5678::1  |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                               |
        ~                               ~
        ~ Original Packet Body/Fragment ~
        ~                               ~
        |                               |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |     OAL Trailing Checksum     |
        |     (final-fragment only)     |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Carrier Packet Format

(Note that carrier packets exchanged by Hosts on ENETs do not include the OAL IPv6 or CRH-32 headers, i.e., the OAL encapsulation is NULL and only the Fragment Header and L2 encapsulations are included.)

In this format, the OAL source encapsulates the original IP header and packet/parcel body/fragment 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 L2 headers are prepared as discussed in [I-D.templin-6man-omni]. The OAL source sends 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 Client XLA-MNP prefix information that may change occasionally due to regional node mobility, as well as XLA-MNP prefix information for Relay non-MNPs and per-segment ULA prefix information that rarely changes. OMNI link Gateways 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 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 an AERO Forwarding Information Base (AFIB) that caches AERO Forwarding Vectors (AFVs) which can provide both carrier packet Identification context and more direct forwarding "shortcuts" that avoid strict spanning tree paths. As a result, the spanning tree is always available but OMNI interfaces can often use the AFIB to greatly improve performance and reduce load on critical infrastructure elements.

For OAL packets/fragments undergoing L2 re-encapsulation at an OAL intermediate node, the OMNI interface removes the L2 encapsulation headers and reassembles only if the OAL packet/fragment is addressed to itself. The OMNI interface then decrements the OAL IPv6 header Hop Limit and discards the packet/fragment if the Hop Limit reaches 0. Otherwise, the OMNI interface updates the OAL addresses if necessary, recalculates the OAL checksum and re-fragments if necessary, then re-encapsulates each fragment in new L2 encapsulation headers to form carrier packets appropriate for next segment forwarding.

When an FHS Gateway forwards an OAL packet/fragment to an LHS Gateway over the unsecured spanning tree, it reconstructs the OAL header based on AFV state, inserts a CRH-32 immediately following the OAL header and adjusts the OAL payload length and destination address field. The FHS Gateway includes a single AFVI in the CRH-32 that the LHS Gateway can use to search its AFIB, then forwards the OAL packet/fragment over the unsecured spanning tree. When the LHS Gateway receives the OAL packet/fragment, it locates the AFV for the next hop based on the CRH-32 AFVI then re-applies header compression (resulting in the removal of the CRH-32) and forwards the OAL packet/fragment to the next hop.

3.7. OMNI Interface Decapsulation

When an OAL node receives OAL packets/fragments addressed to another node, it discards the L2 headers and includes new L2 headers appropriate for the next hop in the forwarding path to the OAL destination. The node then sends these new carrier packets into the next hop underlay interface.

When an OAL node receives OAL packets/fragments addressed to itself, it discards the L2 headers, verifies the Identification, reassembles to obtain the original OAL packet (or super-packet - see: [I-D.templin-6man-omni]) then finally verifies the OAL checksum.

Next, if the enclosed original IP packet(s)/parcel(s) are destined either to itself or to a destination reached via an interface other than the OMNI interface, the OAL node discards the OAL encapsulation and forwards the original IP packet(s)/parcel(s) to the network layer.

If the original IP packet(s)/parcel(s) are destined to another node reached by the OMNI interface, the OAL node instead changes the OAL source to its own address, changes the OAL destination to the ULA of the next-hop node over the OMNI interface, decrements the Hop Limit, recalculates the OAL checksum, refragments if necessary, includes new L2 headers appropriate for the next hop, then sends these new carrier packets into the next hop underlay interface.

3.8. OMNI Interface Data Origin Authentication

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

* AERO Gateways and Proxy/Servers accept carrier packets received from the secured spanning tree.

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

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

* AERO Hosts, Clients, Relays, Proxy/Servers and Gateways verify carrier packet L2 encapsulation addresses according to [I-D.templin-6man-omni].

* AERO nodes accept OAL packets/fragments with Identification values within the current window for the OAL source neighbor for a specific underlay interface pair and drop any packets with out-of-window Identification values.

AERO nodes silently drop any packets/parcels 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 underlay links with diverse MTUs while observing both a minimum and per-path Maximum Payload Size (MPS). The functions of the OAL and OMNI interface packet sizing considerations are specified in [I-D.templin-6man-omni]. (Note that the OMNI interface accommodates an assured MTU of 65535 octets due to the use of fragmentation, and can optionally expose larger MTUs to upper layers for best-effort Jumbogram services.)

When the network layer presents an original IP packet/parcel to the OMNI interface, the OAL source encapsulates and fragments the packet/parcel if necessary. When the network layer presents the OMNI interface with multiple original IP packets/parcels bound to the same OAL destination, the OAL source can concatenate them as a single OAL super-packet as discussed in [I-D.templin-6man-omni] before applying fragmentation. The OAL source then encapsulates each OAL fragment in L2 headers for transmission as carrier packets over an underlay interface connected to either a physical link (e.g., Ethernet, WiFi, Cellular, etc.) 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 Gateway in the path (see: Section 3.10.4), 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/parcels 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/parcels 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/parcels 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 underlay interfaces. (If forwarding state indicates that the original IP packet/parcel should instead be forwarded back to the network layer, the packet/parcel 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 link layer, or reassembled and forwarded to the network layer where they are subject to either local delivery or IP forwarding.

When the network layer forwards an original IP packet/parcel into the OMNI interface, it decrements the TTL/Hop Limit following standard IP router conventions. Once inside the OMNI interface, however, the OAL does not further decrement the original IP packet/parcel TTL/Hop Limit since its adaptation layer forwarding actions occur below the network layer. The original IP packet/parcel's TTL/Hop Limit will therefore be the same when it exits the destination OMNI interface as when it first entered the source OMNI interface.

When an OAL intermediate node receives a carrier packet, it discards the L2 headers to obtain the enclosed OAL packet/fragment. When the intermediate node forwards an OAL packet/fragment not addressed to itself, it decrements the OAL Hop Limit without decrementing the network layer IP TTL/Hop Limit. If decrementing would cause the OAL Hop Limit to become 0, the OAL intermediate node drops the OAL packet/fragment. This ensures that original IP packet(s)/parcel(s) cannot enter an endless loop.

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

If an OAL packet/fragment matches the Interface Attributes and/or Traffic Selectors of multiple outgoing interfaces and/or neighbor interfaces, the OMNI interface replicates the packet and sends a separate copy via each of the (outgoing / neighbor) interface pairs; otherwise, it sends a single copy via an interface with the best matching attributes/selectors. (While not strictly required, the likelihood of successful reassembly may improve when the OMNI interface sends all fragments of the same fragmented OAL packet/fragment consecutively over the same underlay interface pair to avoid complicating factors such as delay variance and reordering.) AERO nodes keep track of which underlay interfaces are currently "reachable" or "unreachable", and only use "reachable" interfaces for forwarding purposes.

The ULA Subnet ID value is used only for subnet coordination within a local OMNI link segment. When a node forwards an OAL packet/fragment addressed to a ULA with a foreign Global and/or Subnet ID value, it forwards the OAL packet/fragment based solely on the OMNI link routing information. For this reason, OMNI link routing and forwarding table entries always include both ULA-RNDs with their associated prefix lengths and XLA-MNPs which encode an MNP while leaving the Global and Subnet ID values set to 0.

The following sections discuss the OMNI interface-specific forwarding algorithms for Hosts, Clients, Proxy/Servers and Gateways. In the following discussion, an original IP packet/parcel'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 {ULA,XLA}-MNP).

3.10.1. Host Forwarding Algorithm

When an original IP packet/parcel enters a Host's OMNI interface from the network layer the Host searches for a NCE that matches the destination. If there is a matching NCE, the Host performs OMNI L2 encapsulation, fragments if necessary as discussed in Section 6.13 of [I-D.templin-6man-omni] then sends the resulting carrier packets into the ENET addressed to the L2 address of the neighbor. If there is no match, the host instead sends the carrier packets to its upstream Client.

After sending the carrier packet, the Host may receive an OAL Redirect message from its upstream Client to inform it of another AERO node on the same ENET that would provide a better first hop. The Host authenticates the Redirect message, then updates its neighbor cache accordingly.

3.10.2. Client Forwarding Algorithm

When an original IP packet/parcel enters a Client's OMNI interface from the network layer the Client searches for a NCE that matches the destination. If there is a matching NCE for a neighbor reached via an ANET/INET interface (i.e., an upstream interface), the Client selects one or more "reachable" neighbor interfaces in the entry for forwarding purposes. Otherwise, the Client performs OAL encapsulation and fragmentation, forwards the resulting OAL packet/fragment to an FHS Proxy/Server, then either invokes address resolution and multilink forwarding procedures per Section 3.13 or allows the FHS Proxy/Server to invoke these procedures on its behalf. If there is a matching NCE for a neighbor reached via an ENET interface (i.e., a downstream interface), the Client instead forwards the original IP packet/parcel to the downstream Host or Client using encapsulation and fragmentation if necessary.

When a carrier packet enters a Client's OMNI interface from the link layer, the Client discards the L2 headers to obtain the OAL packet/fragment then examines the OAL destination. 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 the matching AFV, then reassembles/decapsulates as necessary and delivers the original IP packet/parcel to the network layer. If the OAL destination matches a NCE for a peer Client on an ENET interface, the Client instead forwards the OAL packet/fragment to the peer while decrementing the OAL Hop Limit. If the OAL destination matches a NCE for a Host on an ENET interface, the Client instead reassembles then forwards the original IP packet/parcel to the Host while using L2 encapsulation and fragmentation (i.e., without invoking the OAL) if necessary. If the OAL destination does not match, the Client drops the original IP packet/parcel and MAY return a network-layer ICMP Destination Unreachable message subject to rate limiting (see: Section 3.11).

When a Client forwards an OAL packet/fragment from an ENET Host to a neighbor connected to the same ENET, it also returns a Redirect message to inform the Host that it can reach the neighbor directly as an ENET peer.

Note: Clients and their FHS Proxy/Server (and other Client) peers can exchange original IP packets/parcels over ANET underlay interfaces using OMNI L2 encapsulation without invoking the OAL, since the ANET is secured at the link and physical layers. By forwarding original IP packets/parcels without invoking the OAL, the ANET peers use the same L2 encapsulation and fragmentation procedures as specified for Hosts above.

Note: The forwarding table entries established in peer Clients of a multihop forwarding region are based on ULA-MNPs and/or TLAs used to seed the multihop routing protocols. When ULA-MNPs are used, the ULA /64 prefix provides topological relevance for the multihop forwarding region, while the 64-bit Interface Identifier encodes the Client MNP. Therefore, Clients can forward atomic fragments with compressed OAL headers that do not include ULA or AFVI information by examining the MNP-based addresses in the original IP packet/parcel header. In other words, each forwarding table entry contains two pieces of forwarding information - the ULA information in the prefix and the MNP information in the interface identifier.

3.10.3. Proxy/Server and Relay Forwarding Algorithm

When the network layer admits an original IP packet/parcel into a Proxy/Server’s OMNI interface, the OAL drops the packet/parcel to avoid looping if forwarding state indicates that it should be forwarded back to the network layer. Otherwise, the OAL examines the IP destination address to determine if it matches the ULA of a neighboring Gateway found in the OMNI interface’s network layer neighbor cache. If so, the Proxy/Server performs OAL encapsulation and fragmentation followed by L2 encapsulation then sends the resulting carrier packets to the neighboring Gateway over a secured tunnel to support the operation of the BGP routing protocol. If the destination is a non-ULA, the Proxy/Server instead assumes the Relay role and forwards the original IP packet/parcel in a similar manner as for Clients. Specifically, if there is a matching NCE the Proxy/Server selects one or more "reachable" neighbor interfaces in the entry for forwarding purposes; otherwise, the Proxy/Server performs OAL encapsulation/fragmentation followed by L2 encapsulation and sends the resulting carrier packets while invoking address resolution and multilink forwarding procedures per Section 3.13.

When the Proxy/Server receives carrier packets on underlay interfaces that contain OAL packets/fragments with both a source and destination OAL address that correspond to the same Client’s delegated MNP, the Proxy/Server drops the carrier packets regardless of their OMNI link point of origin. The Proxy/Server also drops original IP packets/parcels received on underlay interfaces either directly from an ANET Client or following reassembly of carrier packets received from an ANET/INET Client if the original IP destination corresponds to the same Client’s delegated MNP. Proxy/Servers also drop carrier packets that contain OAL packets/fragments with foreign OAL destinations that do not match their own ULA, the ULA of one of their Clients or a ULA corresponding to one of their GUA routes. These checks are essential to prevent forwarding inconsistencies from accidentally or intentionally establishing endless loops that could congest nodes and/or ANET/INET links.

Proxy/Servers process carrier packets that contain OAL packets/fragments with OCH headers or with destinations that match their ULA and also include a CRH-32 header that encodes one or more AFVIs. The Proxy/Server examines the next AFVI in the OAL headers to locate the corresponding AFV entry in the AFIB. If the carrier packets were not received from the secured spanning tree, the Proxy/Server must then verify that the L2 addresses are "trusted" according to the AFV. If the carrier packets were trusted, the Proxy/Server then forwards them according to the AFV state while decrementing the OAL packet/fragment Hop Limit.

For OAL packets/fragments with destinations that match their ULA but do not include a CRH-32/OCH, the Proxy/Server instead discards the L2 headers and performs OAL reassembly if necessary to obtain the original IP packet/parcel. For data packets/parcels addressed to their own ULA that arrived via the secured spanning tree, the Proxy/Server delivers the original IP packet/parcel to the network layer to support secured BGP routing protocol control messaging. For data packets/parcels originating from one of its dependent Clients, the Proxy/Server instead performs OAL encapsulation/fragmentation then performs L2 encapsulation and sends the resulting carrier packets while invoking address resolution and multilink forwarding procedures per Section 3.13. For IPv6 ND control messages, the Proxy/Server instead authenticates the message and processes it as specified in later sections of this document while updating neighbor cache and/or AFIB state accordingly.

When the Proxy/Server receives a carrier packet that contains an OAL packet/fragment with OAL destination set to a {ULA,XLA}-MNP 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 ULA of the new Proxy/Server, decrements the OAL Hop Limit, then supplies new OMNI L2 headers and forwards the resulting carrier packet into the spanning tree which will eventually deliver it to the new Proxy/Server. If the neighbor cache state for the Client is REACHABLE and the Proxy/Server is a Hub responsible for serving as the Client’s address resolution responder and/or default router, it submits the OAL packet/fragment for reassembly then decapsulates and processes the resulting IPv6 ND message or original IP packet/parcel accordingly. Otherwise, the Proxy/Server decrements the OAL Hop Limit, supplies new OMNI L2 headers and sends the carrier packets to the Client which must then perform data origin verification and reassembly. (In the latter case, the Client may receive fragments of the same original IP packet/parcel from different Proxy/Servers but this will not interfere with reassembly.)

When the Proxy/Server receives a carrier packet that contains an OAL packet/fragment with OAL destination set to a {ULA,XLA}-MNP that does not match the MSP, it accepts the carrier packet only if data origin authentication succeeds and if there is a network layer forwarding table entry for a GUA route that matches the MNP. The Proxy/Server then discards the L2 headers, performs OAL reassembly and decapsulation to obtain the original IP packet/parcel, then presents it to the network layer (as a Relay) where it will be delivered according to standard IP forwarding.

Clients and their FHS Proxy/Server peers can exchange original IP packets/parcels over ANET underlay interfaces using L2 encapsulation with Type-3 compressed OAL headers that include only fragmentation information and no OAL addressing information, since the ANET is secured at the link and physical layers. (For packets that do not require fragmentation, the peers can even omit the Type-3 header.) FHS Proxy/Servers will then supply a Type 0/1/2 OAL header when they forward ANET Client original IP packets/parcels toward final destinations located in other networks.

Proxy/Servers forward OAL packets/fragments received in secure control plane carrier packets via the SRT secured spanning tree and forward other OAL packets/fragments 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 applies data origin authentication itself and/or forwards the enclosed unsecured OAL contents toward the destination which must apply data origin authentication on its own behalf.

If the Proxy/Server has multiple original IP packets/parcels to send to the same neighbor, it can concatenate them as a single OAL super-packet [I-D.templin-6man-omni]. If the super-packet begins with an IPv6 ND control message to be sent over the secured spanning tree, the remainder of the super-packet also traverses the secured spanning tree.

3.10.4. Gateway Forwarding Algorithm

When the network layer admits an original IP packet/parcel into the Gateway's OMNI interface, the OAL drops the packet if routing indicates that it should be forwarded back to the network layer to avoid looping. Otherwise, the Gateway examines the IP destination address to determine if it matches the ULA of a neighboring Gateway or Proxy/Server by examining the OMNI interface's network layer neighbor cache. If so, the Gateway performs OAL encapsulation/fragmentation followed by L2 encapsulation and forwards the resulting carrier packets to the neighboring Gateway or Proxy/Server over a secured tunnel to support the operation of the BGP routing protocol between OAL neighbors.

Gateways forward OAL packets/fragments received in spanning tree carrier packets while decrementing the OAL Hop Limit but not the original IP header TTL/Hop Limit. Gateways send carrier packets that contain OAL packets/fragments with critical IPv6 ND control messages or BGP routing protocol control messages via the SRT secured spanning tree, and may send other carrier packets via the secured/unsecured spanning tree or via more direct paths according to AFIB information. When the Gateway receives a carrier packet, it removes the L2 headers to obtain the OAL packet/fragment then searches for an AFIB entry that matches the OAL header AFVI or an IP forwarding table entry that matches the OAL destination address.

Gateways process carrier packets that contain OAL packets/fragments with OAL destinations that do not match their ULA or the SRT Subnet Router Anycast address in the same manner as for traditional IP forwarding within the OAL, i.e., they forward packets not explicitly addressed to themselves. Gateways locally process OAL packets/fragments with OCH headers or full OAL headers with their ULA or the SRT Subnet Router Anycast address as the OAL destination. If the OAL packet/fragment contains an OCH or a full OAL header with a CRH-32 extension, the Gateway examines the next AFVI in the CRH-32 to locate the AFV entry in the AFIB for next hop forwarding. If an AFV is found, the Gateway uses the next hop AFVI to forward the OAL packet/fragment to the next hop while decrementing the OAL Hop Limit but without reassembling. If the Gateway has a NCE for the target Client with an entry for the target underlay interface and current L2 addresses, the Gateway instead forwards the OAL packet/fragment directly to the target Client while using the final hop AFVI instead of the next hop (see: Section 3.13.3).

If the OAL packet/fragment includes a full OAL header addressed to itself but does not include an AFVI, the Gateway instead reassembles if necessary, verifies the OAL checksum, and processes the OAL packet further. The Gateway first determines whether the OAL packet includes an NS/NA message then processes the message according to the multilink forwarding procedures discussed in Section 3.13. If the carrier packets arrived over the secured spanning tree and the enclosed OAL packets/fragments are addressed to its ULA, the Gateway instead reassembles then discards the OAL header and forwards the original IP packet/parcel to the network layer to support secured BGP routing protocol control messaging. The Gateway instead drops all other OAL packets.

Gateways forward OAL packets/fragments received in carrier packets that arrived from a first segment via the secured spanning tree to the next segment also via the secured spanning tree. Gateways forward OAL packets/fragments received in carrier packets that arrived from a first segment via the unsecured spanning tree to the next segment also via the unsecured spanning tree. Gateways configure 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 underlay interface to be used for transmission (i.e., a secured tunnel or an open INET interface).

As for Proxy/Servers, Gateways must verify that the L2 addresses of carrier packets not received from the secured spanning tree are "trusted" before forwarding according to an AFV (otherwise, the carrier packet must be dropped).

3.11. OMNI Interface Error Handling

When an AERO node admits an original IP packet/parcel 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 an underlay network 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 L2 and OAL  |   a
        |      encapsulation headers    |   c
        ~                               ~   k
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   e
        ~                               ~   t
        |original IP packet/parcel hdrs |
        |     (first-fragment only)     |   i
        ~                               ~   n
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        ~                               ~   e
        |     Portion of the body of    |   r
        |  the original IP packet/parcel|   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:

- 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].

- 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 carrier packets.

* 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 3.15.5.

* 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 underlay path as unusable and use another underlay 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 Gateway receives a carrier packet for which the network-layer destination address is covered by an MSP assigned to a black-hole route, the Gateway drops the carrier 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 OAL 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 an 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 Mobility Service Coordination

AERO nodes observes the Router Discovery and Prefix Registration specifications found in Section 15 of [I-D.templin-6man-omni]. AERO nodes further coordinate their autoconfiguration actions with the mobility service 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 ANET/INET underlay interfaces with a FHS Proxy/Server. Each FHS Proxy/Server locally services one or more of the Client’s underlay interfaces, and the Client typically selects one among them to serve as the Hub Proxy/Server (the Client may instead select a "third-party" Hub Proxy/Server that does not directly service any of its underlay interfaces). 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 other 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 ULA or link-scoped All-Routers multicast as the destination. An FHS Proxy/Server assumes the proxy role when it receives an RS message with the ULA of another Proxy/Server as the destination. (An FHS Proxy/Server can also assume the proxy role when it receives an RS message addressed to link-scoped All-Routers multicast if it can determine the ULA of another Proxy/Server to serve as a Hub.)

Hosts and Clients on ENET interfaces associate with an upstream Client on the ENET the same as a Client would associate with an ANET Proxy/Server. In particular, the Host/Client sends an RS message via the ENET which directs the message to the upstream Client. The upstream Client returns an RA message. In this way, the downstream nodes see the ENET as an ANET and see the upstream Client as a Proxy/Server for that ANET.

AERO Hosts, 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. The IPv6 ND messages are exchanged between the Client and Hub Proxy/Server (via any FHS Proxy/Servers acting as proxys) 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 XLA-MNP 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 a TLA as the source address of an RS message and with an OMNI option with prefix delegation parameters to request an MNP.

The following sections outlines Host, Client and Proxy/Server behaviors based on the Router Discovery and Prefix Registration specifications found in Section 15 of [I-D.templin-6man-omni]. These sections observe all of the OMNI specifications, and include additional specifications of the interactions of Client-Proxy/Server RS/RA exchanges with the AERO mobility service.

3.12.2. AERO Host and Client Behavior

AERO Hosts and Clients discover the addresses of candidate 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 Host/Client can discover Proxy/Server addresses through a layer 2 data link login exchange, or through an RA response to a multicast/anycast RS as described below. In the absence of other information, the Host/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"). The name resolution returns a set of resource records with Proxy/Server address information.

The Host/Client then performs RS/RA exchanges over each of its underlay interfaces to associate with (possibly multiple) FHS Proxy/Serves and a single Hub Proxy/Server as specified in Section 15 of [I-D.templin-6man-omni]. The Host/Client then sends each 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) and waits up to RetransTimer milliseconds for an RA message reply (see Section 3.12.3) while retrying up to MAX_RTR_SOLICITATIONS if necessary. If the Host/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 Proxy/Server and try another Proxy/Server.

After the Host/Client registers its underlay 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 Host/Client prepares an RS message to send over any available underlay interface as above. The RS includes an OMNI option with prefix registration/delegation information and with an Interface Attributes sub-option specific to the selected underlay interface. When the Host/Client receives the Hub Proxy/Server's RA response, it has assurance that both the Hub and FHS Proxy/Servers have been updated with the new information.

If the Host/Client wishes to discontinue use of a Hub Proxy/Server it issues an RS message over any underlay interface with an OMNI Proxy/Server Departure sub-option that encodes the (old) Hub Proxy/Server's ULA. When the Hub Proxy/Server processes the message, it releases the MNP, sets the NCE state for the Host/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. (Alternatively, when the Host/Client associates with a new FHS/Hub Proxy/Server it can include an OMNI "Proxy/Server Departure" sub-option in RS messages with the ULAs of the Old FHS/Hub Proxy/Servers.)

3.12.3. AERO Proxy/Server Behavior

AERO Proxy/Servers act as both IP routers and IPv6 ND proxys, and support a prefix delegation/registration service for Clients. Proxy/Servers arrange to add their ULAs 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 underlay network connections.

When a FHS/Hub 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, it processes the RS as specified in Section 15 of [I-D.templin-6man-omni]. When the Hub Proxy/Server receives the RS, it determines the correct MNPs to provide to the Client by processing the XLA-MNP prefix parameters and/or the DHCPv6 OMNI sub-option. When the Hub Proxy/Server returns the MNPs, it also creates an XLA-MNP forwarding table entry for the MNP resulting in a BGP update (see: Section 3.2.3). The Hub Proxy/Server then returns an RA to the Client with destination set to the source of the RS (if an FHS Proxy/Server on the return path proxys the RA, it changes the destination to the Client’s ULA-MNP).

After the initial RS/RA exchange, the Hub Proxy/Server maintains a ReachableTime timer for each of the Client’s underlay 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 XLA-MNP route from the routing system after a short delay (e.g., 2 seconds). If ReachableTime expires before a new RS is received on an individual underlay interface, the Hub Proxy/Server marks the interface as DOWN. If ReachableTime expires before any new RS is received on any individual underlay 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 XLA-MNP from the routing system.

The Hub Proxy/Server processes any IPv6 ND messages pertaining to the Client while forwarding to the Client or responding on the Client’s behalf as necessary. 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. The Hub Proxy/Server may also receive carrier packets via the secured spanning tree that contain initial data sent while route optimization is in progress. The Hub Proxy/Server reassembles the enclosed OAL packets/fragments, then re-encapsulates/re-fragments and sends the carrier packets to the target Client via an FHS Proxy/Server if necessary. Finally, If the NCE is in the DEPARTED state, the old Hub Proxy/Server forwards any OAL packets/fragments it receives from the secured spanning tree and destined to the Client to the new Hub Proxy/Server, then deletes the entry after DepartTime expires.

Note: Clients SHOULD arrange to notify former Hub Proxy/Servers of their departures, but Hub Proxy/Servers are responsible for expiring neighbor cache entries and withdrawing XLA-MNP 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. Additional Proxy/Server Considerations

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

- when FHS Proxy/Server "B" receives a Client RS message, it first verifies that the OAL Identification is within the window for the NCE that matches the {ULA,XLA}-MNP in the RS source address for this Client neighbor and authenticates the message. If no NCE was found, Proxy/Server "B" instead creates one in the STALE state and caches the Client-supplied Interface Attributes, Origin Indication and OMNI Window Synchronization sub-option parameters as well as the Client's observed L2 addresses (noting that they may differ from the Origin addresses if there were NATs on the path). Proxy/Server "B" then examines the RS destination address. If the destination address is the ULA of a different Proxy/Server "A", Proxy/Server "B" prepares a separate proxyed version of the RS message with an OAL header with source set to its own ULA and destination set to Proxy/Server B's ULA. Proxy/Server "B" also writes its own information over the Interface Attributes sub-option supplied by the Client, omits or zeros the Origin Indication sub-option then forwards the message into the OMNI link secured spanning tree.

- when Hub Proxy/Server "A" receives the RS, it assume the Hub role, delegates an MNP for the Client if necessary according to the Prefix Length sub-option included by the Client, and creates/updates a NCE indexed by the Client's XLA-MNP with FHS Proxy/Server "B"'s Interface Attributes as the link-layer address information for this FHS ifIndex. Hub Proxy/Server "A" then prepares an RA message with source set to its own ULA, destination set to the source of the RS message, and with a Prefix Length sub-option set to the actual MNP length it will delegate to the Client. Hub Proxy/Server "A" then encapsulates the RA in an OAL header with source set to its own ULA and destination set to the ULA of FHS Proxy/Server, then finally performs fragmentation if necessary and sends the resulting carrier packets into the secured spanning tree.

- when FHS Proxy/Server "B" reassembles the RA, it locates the Client NCE based on the RA destination. If the RA message includes an OMNI "Proxy/Server Departure" sub-option with non zero old FHS/Hub Proxy/Server ULAs that do not match its own ULA, FHS Proxy/Server "B" first sends a uNA to the old FHS/Hub Proxy/Servers named in the sub-option. If the RA message delegates a new XLA-MNP, Proxy/Server "B" then resets the RA destination to the corresponding ULA-MNP for this interface. Proxy/Server "B" then re-encapsulates the message with OAL source set to its own ULA and OAL destination set to ULA that appeared in the Client's RS message OAL source, with an appropriate Identification value, with an authentication signature if necessary, with the Client's Interface Attributes sub-option echoed and with the cached observed L2 addresses written into an Origin Indication sub-option. Proxy/Server "B" sets the P flag in the RA flags field to indicate that the message has passed through a proxy [RFC4389], includes responsive window synchronization parameters, then fragments the RA if necessary and returns the fragments to the Client.

* The Client repeats this process over each of its additional underlay interfaces while treating each additional FHS Proxy/ Server "C", "D", "E", etc. as a proxy to facilitate RS/RA exchanges between the Hub and the Client. The Client creates/ updates NCEs for each such FHS Proxy/Server as well as the Hub Proxy/Server in the process.

After the initial RS/RA exchanges each FHS Proxy/Server forwards any of the Client's carrier packets that contain OAL packets/fragments with destinations for which there is no matching NCE to a Gateway using OAL encapsulation with its own ULA as the source and with destination determined by the Client. The Proxy/Server instead forwards any OAL packets/fragments 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 3.13.

While the Client is still associated with FHS Proxy/Servers "B", "C", "D", etc., each FHS Proxy/Server can send NS, RS and/or unsolicited NA messages to update the neighbor cache entries of other AERO nodes on behalf of the Client based on changes in Interface Attributes, Traffic Selectors, etc. 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, FHS Proxy/ Servers "B", "C", "D" can send unsolicited RAs over the Client's underlay 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 FHS 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 XLA-MNP as the source in order to differentiate them from the IPv6 ND messages sent by a FHS Proxy/Server.

If the Client becomes unreachable over all underlay interfaces 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 OAL packets/fragments destined to the Client to a Gateway via OAL encapsulation. When DepartTime expires, the Hub Proxy/Server deletes the NCE, withdraws the XLA-MNP route and discards any further carrier packets that contain OAL packets/fragments 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/parcels without invoking the OAL so that the ANET routing system transports the original IP packets/parcels to the Proxy/Server. This can be 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 underlay 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 XLA-MNP (or to a TLA), and with destination address set to the ULA of the Client's selected Proxy/Server or to link-scoped 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 ANET 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 an underlay 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 original IPv6 ND message checksum and authentication signature values are invalidated and must be re-calculated.

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 3.13.

3.12.3.2. 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 fashion that parallels 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 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 exchanges with the Hub Proxy/Server, e.g., one exchange per second. The FHS Proxy/Server sends the NS message via the spanning tree with its own ULA as the source and the ULA of the Hub Proxy/Server as the destination, and the Hub Proxy/Server responds with an NA. 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 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 a different Proxy/Server to assume the Hub role (i.e., by sending an RS with destination set to the ULA of the new Hub).

3.12.3.3. DHCPv6-Based Prefix Registration

When a Client is not pre-provisioned with an MNP, 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 a TLA 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 creates an XLA based on the delegated MNP adds an XLA-MNP route to the routing system. The Hub Proxy/Server then sends an RA back to the Client either directly or via an FHS Proxy/Server acting as a proxy. The Proxy/Server that returns the RA directly to the Client sets the (newly-created) ULA-MNP as the destination address and with a DHCPv6-PD Reply message sub-option 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 {ULA,XLA}-MNP based on the delegated MNP.

Note: Further details of the DHCPv6-PD based MNP registration (as well as a minimal MNP delegation alternative that avoids including a DHCPv6 message sub-option in the RS) are found in [I-D.templin-6man-omni].

Note: when the Hub Proxy/Server forwards an RA to the Client via a different node acting as a FHS Proxy/Server, the Hub sets the RA destination to the same address that appeared in the RS source. The FHS Proxy/Server then subsequently sets the RA destination to the ULA-MNP when it forwards the Proxyed version of the RA to the Client - see [I-D.templin-6man-omni] for further details.

3.13. AERO Address Resolution, Multilink Forwarding and Route Optimization

AERO nodes invoke address resolution, multilink forwarding and route optimization when they need to forward initial original IP packets/parcels to new neighbors over ANET/INET interfaces and for ongoing multilink forwarding coordination with existing neighbors. Address resolution is based on an IPv6 ND NS/NA(AR) messaging exchange between an Address Resolution Source (ARS) and the target neighbor as the Address Resolution Target (ART). Either the ART itself or the ART's current Hub Proxy/Server serves as the Address Resolution Responder (ARR).

Address resolution is initiated by the first eligible ARS closest to the original source as follows:

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

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

* For Clients on INET interfaces, the Client itself is the ARS.

* For correspondent nodes on INET/ENET interfaces serviced by a Relay, the Relay is the ARS.

* For Clients that engage the Hub Proxy/Server in "mobility anchor" mode, the Hub Proxy/Server is the ARS.

* For peers within the same ANET/ENET, route optimization is through receipt of Redirect messages.

The AERO routing system directs an address resolution request sent by the ARS to the ARR. The ARR then returns an address resolution reply which must include information that is current, consistent and authentic. Both the ARS and ARR are then jointly responsible for periodically refreshing the address resolution, and for quickly informing each other of any changes. Following address resolution, the ARS and ART perform continuous unicast multilink forwarding and route optimization exchanges to maintain optimal forwarding profiles.

The address resolution, multilink forwarding and route optimization procedures are specified in the following sections.

3.13.1. Multilink Address Resolution

When one or more original IP packets/parcels from a source node destined to a target node arrives, the ARS checks for a NCE with an XLA-MNP that matches the target destination. If there is a NCE in the REACHABLE state, the ARS invokes the OAL and sends the resulting carrier packets according to the cached state then returns from processing.

Otherwise, if there is no NCE the ARS creates one in the INCOMPLETE state. The ARS then prepares an NS message for Address Resolution (NS(AR)) to send toward an ART while including the original IP packet(s)/parcel(s) as trailing data following the NS(AR) in an OAL super-packet [I-D.templin-6man-omni]. The resulting NS(AR) message must be sent securely, and includes:

* the ULA of the ARS as the source address.

* the XLA corresponding to the original IP packet/parcel's destination as the Target Address, e.g., for 2001:db8:1:2::10:2000 the Target Address is fd00::2001:db8:1:2.

* the Solicited-Node multicast address [RFC4291] formed from the lower 24 bits of the original IP packet/parcel'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, includes Interface Attributes and/or Traffic Selectors for all of the source Client's underlay interfaces and a Prefix Length sub-option with a valid length for its claimed MNP. The ARS then calculates and includes the authentication signature (if necessary) followed by the checksum, then submits the NS(AR) message for OAL encapsulation. The ARS sets the OAL source to its own ULA and sets the OAL destination according to the Client's RS message 'U' flag (see: [I-D.templin-6man-omni]). If the 'U' flag was set, the ARS sets the OAL destination to the ULA of its Hub Proxy/Server which maintains a Report List; otherwise, the ARS sets the destination to the XLA-MNP corresponding to the ART. The ARS then selects an identification value, inserts a fragment header, calculates the OAL checksum, performs fragmentation and L2 encapsulation, then sends the resulting carrier packets into the SRT secured spanning tree without decrementing the network-layer TTL/Hop Limit field.

When the ARS is a Client, it must instead use the ULA of one of its FHS Proxy/Servers as the OAL destination. The ARS Client then fragments, performs L2 encapsulation and forwards the carrier packets to the FHS Proxy/Server. The FHS Proxy/Server then discards the L2 headers, verifies the Identification, reassembles if necessary, verifies the NS(AR) checksum/authentication signature and confirms that the Client's claimed Prefix Length is valid for its ULA-MNP source address. The FHS Proxy/Server then changes the OAL source to its own ULA and changes the OAL destination to the ULA of the Hub Proxy/Server or XLA-MNP corresponding to the ART as specified above. The FHS Proxy/Server next selects an appropriate Identification, calculates the OAL checksum, re-fragments, performs L2 encapsulation and sends the resulting carrier packets into the secured spanning tree on behalf of the Client.

Note: both the source and target Client/Relay and their Hub Proxy/Servers include current and accurate information for their multilink Interface Attributes profile. The Hub Proxy/Servers can be trusted to provide an authoritative ARR response and/or mobility update message on behalf of the source/target should the need arise. While the source or target itself has no such trust basis, any attempt to mount an attack by providing false Interface Attributes information would only result in black-holing of return traffic, i.e., the "attack" could only result in denial of service to the source/target itself. Therefore, the source/target's asserted Interface Attributes need not be validated by the Hub Proxy/Server.

3.13.1.1. ARS Hub Proxy/Server NS(AR) Processing

If the ARS Client's Hub Proxy/Server maintains a Report List, the carrier packets containing the NS(AR) will first arrive at the at the Hub due to the OAL destination address supplied by the ARS (see above). This source Hub then discards the L2 headers, reassembles then records the NS Target Address in the Report List for this source Client. The Hub then leaves the OAL source address unchanged, but changes the OAL destination address to the XLA corresponding to the NS Target Address. The Hub then decrements the OAL header Hop Limit, includes an appropriate Identification, recalculates the OAL checksum, refragments, performs L2 encapsulation and sends the resulting carrier packets into the secured spanning tree.

3.13.1.2. Relaying the NS(AR)

When a Gateway receives carrier packets containing the NS(AR), it discards the L2 headers and determines the next hop by consulting its standard IPv6 forwarding table for the OAL header XLA destination address. The Gateway next decrements the OAL header Hop Limit, then performs L2 encapsulation and sends the carrier packet(s) via the secured spanning tree the same as for any IPv6 router, where they may traverse multiple OMNI link segments. The final-hop Gateway will deliver the carrier packets via the secured spanning tree to the Hub Proxy/Server (or Relay) that services the ART.

3.13.1.3. NS(AR) Processing at the ARR/ART

When the Hub Proxy/Server of the ART receives the NS(AR) secured carrier packets with the XLA-MNP of the ART as the OAL destination, it discards the L2 headers, verifies the Identification, reassembles if necessary, verifies the OAL checksum then either forwards the NS(AR) to the ART or processes it locally if it is acting as a Relay or as the ART's designated ARR. The Hub Proxy/Server processes the message as follows:

* if the NS(AR) target matches a Client NCE in the DEPARTED state, the (old) Hub Proxy/Server resets the OAL destination address to the ULA of the Client's new Hub Proxy/Server. The old Hub Proxy/Server then decrements the OAL header Hop Limit, recalculates the OAL checksum, re-fragments, includes appropriate L2 headers then forwards the resulting carrier packets over the secured spanning tree.

* If the NS(AR) target matches a Client NCE in the REACHABLE state, the Hub Proxy/Server notes whether the NS(AR) arrived from the secured spanning tree. If the message arrived via the secured spanning tree the Hub Proxy/Server verifies the NS checksum only; otherwise, it must also verify the message authentication signature. If the Hub Proxy/Server maintains a Report List for the ART, it next records the NS source address in the Report List for this ART. If the Hub Proxy/Server is the ART's designated ARR, it prepares to return an NA(AR) as discussed below; otherwise, the Hub Proxy/Server determines the underlay interface for the ART and proceeds as follows:

  - If the Hub Proxy/Server is also the FHS Proxy/Server on the underlay interface used to convey the NS(AR) to the ART, it includes an authentication signature if necessary then recalculates the NS(AR) checksum. The Hub then changes the OAL source to its own ULA and OAL destination to the ULA-MNP of the ART, decrements the OAL Hop Limit, includes a suitable identification value, recalculates the OAL checksum, re-fragments if necessary, includes appropriate L2 headers and sends the resulting carrier packets over the underlay interface to the ART.

  - If the Hub Proxy/Server is not the FHS Proxy/Server on the underlay interface used to convey the NS(AR) to the ART, it instead recalculates the NS(AR) checksum, changes the OAL source to its own ULA and changes the OAL destination to the ULA of the FHS Proxy/Server for this ART interface. The Hub Proxy/Server next decrements the OAL Hop Limit, includes a suitable Identification value, recalculates the OAL checksum, re-fragments if necessary, includes appropriate L2 headers and sends the resulting carrier packets over the secured spanning tree.

  - When the FHS Proxy/Server receives the carrier packets, it discards the L2 headers, reassembles and verifies the OAL and NS(AR) checksums, then forwards to the ART the same as described above.

* If the NS(AR) target matches one of its non-MNP routes, the Hub Proxy/Server serves as both a Relay and an ARR, since the Relay forwards original IP packets/parcels toward the (fixed network) target at the network layer.

If the ARR is a Relay or the ART itself, it first creates or updates a NCE for the NS(AR) source address while caching all Interface Attributes and Traffic Selector information. Next, the ARR prepares a (solicited) NA(AR) message to return to the ARS with the source address set to the ART’s XLA, the destination address set to the NS(AR) ULA source address and the Target Address set to the same value that appeared in the NS(AR) Target Address.

The ARR then includes Interface Attributes and Traffic Selector sub-options for all of the ART’s underlay interfaces with current information for each interface and includes a Prefix Length sub-option with the length to apply to the ART’s MNP. The ARR next 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 1 (as an authoritative responder). The ARR finally includes an authentication signature if necessary, calculates the NA message checksum, then submits the NA(AR) for OAL encapsulation with source set to its own ULA and destination set to the ULA that appeared in the NS(AR) OAL source and selects an appropriate Identification. The ARR then calculates the OAL checksum, fragments, includes appropriate L2 headers and forwards the resulting (L2-encapsulated) carrier packets.

When the ART Proxy/Server receives carrier packets sent by an ART acting as an ARR on its own behalf, it reassembles if necessary and verifies the checksum/authentication signature. The Proxy/Server then verifies that the Prefix Length is acceptable, changes the OAL source address to its own ULA and changes the OAL destination to the ULA corresponding to the NA(AR) destination. The Proxy/Server next decrements the OAL Hop Limit, includes an appropriate Identification, recalculates the NA and OAL checksums and fragments if necessary. The Proxy/Server finally includes appropriate L2 headers and sends the carrier packets into the secured spanning tree.

3.13.1.4. Relaying the NA(AR)

When a Gateway receives NA(AR) carrier packets, it discards the L2 headers and determines the next hop by consulting its standard IPv6 forwarding table for the OAL header destination address. The Gateway then decrements the OAL header Hop Limit, re-encapsulates the carrier packets with new L2 headers and forwards them via the SRT secured spanning tree where they may traverse multiple OMNI link segments. The final-hop Gateway will deliver the carrier packets via the secured spanning tree to a Proxy/Server for the ARS.

3.13.1.5. Processing the NA(AR) at the ARS

When the ARS receives the NA(AR) message, it first searches for a NCE that matches the NA(AR) target address. The ARS then processes the message the same as for standard IPv6 Address Resolution [RFC4861]. In the process, it caches all OMNI option information in the NCE for the ART (including Interface Attributes, Traffic Selectors, etc.), and caches the NA(AR) XLA source address as the address of the ART.

When the ARS is a Client, the SRT secured spanning tree will first deliver the solicited NA(AR) message to the FHS Proxy/Server, which re-adjusts the OAL header 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 NA(AR) without needing an additional authentication signature; 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 ULA as the OAL source and the ULA-MNP of the Client as the OAL destination when it forwards the NA(AR). The Proxy/Server then decrements the OAL Hop Limit, includes an appropriate Identification, recalculates the OAL checksum, re-fragments, includes appropriate L2 headers and sends the carrier packets over the underlay interface to the Client.

3.13.1.6. Reliability

After the ARS transmits the first NS(AR), it should wait up to RETRANS_TIMER seconds to receive a responsive NA(AR). The ARR can then retransmit the NS(AR) up to MAX_UNICAST_SOLICIT times before giving up.

3.13.2. Multilink Forwarding

Following address resolution, the ARS and ART (or their Proxy/ Servers) can assert multilink forwarding paths through underlay interface pairs serviced by the same source/destination ULAs by sending unicast NS/NA messages with OMNI AERO Forwarding Parameter (AFP) sub-options. The unicast NS/NA messages establish multilink forwarding state in OAL intermediate nodes in the path between the ARS and ART. Note that either the ARS or ART can independently initiate multilink forwarding by sending unicast NS messages on behalf of specific underlay interface pairs. (Underlay interface directionality (i.e., in/out) must also be factored into the paths established for multilink forwarding.)

Nodes that configure OMNI interfaces include an additional forwarding table termed the AERO Forwarding Information Base (AFIB) that supports OAL packet/fragment forwarding based on OMNI neighbor underlay interface pairs. The AFIB contains per-interface-pair AERO Forwarding Vectors (AFVs) identified by locally-unique 4-octet values known as AFV Indexes (AFVIs). The AFVs cache uncompressed OAL header information as well as the previous/next-hop addressing and AFVI information. The AFVs also cache window synchronization state for the specific underlay interface pair. Using the window synchronization state, simple Identification-based data origin authentication is enabled at each OAL source, intermediate and target node.

OMNI interfaces manage the AFIB in conjunction with their internal Neighbor Cache. OMNI interface NCEs link to (possibly) multiple AFVs, with one AVF per underlay interface pair (according to directionality). When OMNI interface peers need to coordinate, they locate a NCE for the peer then use the NCE as a nexus that aggregates potentially many AVFs. In particular, the NCE caches the AFVI to be used to index the local AFV at the head end of the path.

OAL source, intermediate and target nodes create AFVs/AFVIs when they process an NS message with an AFP sub-option with Job code '00' (Initialize; Build B) or a solicited NA message with Job code '01' (Follow B; Build A) (see: [I-D.templin-6man-omni]). The OAL source of the NS (which is also the OAL destination of the solicited NA) is considered to reside in the "First Hop Segment (FHS)", while the OAL destination of the NS (which is also the OAL source of the solicited NA) is considered to reside in the "Last Hop Segment (LHS)".

The FHS and LHS roles are determined on a per-interface-pair basis. After address resolution, either peer is equally capable of initiating multilink forwarding on behalf of a specific FHS/LHS underlay interface pair. The peer that sends the initiating NS with Job code '00' message for a specific pair becomes the FHS peer while the one that returns the NA response becomes the LHS peer for that pair only. It is therefore quite possible (and even commonplace) that both peers may assume the FHS role for some pairs while assuming the LHS role for other pairs, i.e., even though each peer maintains only a single NCE.

When an OAL node initiates or forwards an NS with Job code ’00’, it creates an AFV, records the NS source and destination ULAs then generates and assigns a locally-unique "B" AFVI (while also caching the "B" values for all previous OAL hops on the path from the FHS OAL source). When the OAL node receives future OAL packets/fragments that include "B", it can unambiguously locate the correct AFV and determine directionality without examining addresses. When the AFV is indexed by its "B" AFVI, it returns the ULAs in (dst,src) order the opposite of how they appeared in the OAL header of the original NS to support full header reconstruction for reverse-path forwarding. (If the NS message included a nested OAL encapsulation, the ULAs of both OAL headers are returned.)

When an OAL node initiates or forwards a solicited NA with Job code ’01’, it uses the "B" AFVI to locate the AFV created by the NS then generates and assigns a locally-unique "A" AFVI (while also caching the "A" values for all previous OAL hops on the path from the LHS OAL source). When the OAL node receives future carrier packets that include "A", it can unambiguously locate the correct AFV and determine directionality without examining addresses. When the AFV is indexed by its "A" AFVI, it returns the ULAs in (src,dst) order the same as they appeared in the OAL header of the original NS to support full header reconstruction for forward-path forwarding. (If the NS message included a nested OAL encapsulation, the ULAs of both OAL headers are returned.)

OAL nodes generate random non-zero 32-bit values as candidate AFVIs which must first be tested for local uniqueness. If a candidate AFVI s already in use, the OAL node repeats the random generation process until it obtains a unique non-zero value. Since the number of AFVs 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. Since the uniqueness property is node-local only, an AFVI locally generated by a first OAL node must not be tested for uniqueness by other OAL nodes.

OAL nodes cache AFVs for up to ReachableTime seconds following their initial creation. If the node processes another NS or NA message specific to an AFV, it resets ReachableTime to REACHABLE_TIME seconds, i.e., the same as for NCEs. If ReachableTime expires, the node deletes the AFV and frees its associated AFVIs so they can be reused for future AFVs.

The following sections provide the detailed specifications of these NS/NA exchanges for all nodes along the forward and reverse paths.

3.13.2.1. FHS Client-Proxy/Server NS Forwarding

When an FHS OAL source has an original IP packet/parcel to send toward an LHS OAL target, it first performs multilink address resolution resulting in the creation of a NCE for the XLA of the target then selects a source and target underlay interface pair. The FHS source uses its cached information for the target interface as LHS information then prepares an NS message with an AFP sub-option with Job code ’00’, includes window synchronization information, then sets the NS source to the XLA of the FHS Client and the NS target to the XLA of the LHS Client. The FHS source next creates an AFV then generates and assigns a locally-unique "B" AFVI to the AFV while also including it as the first "B" entry in the AFP AFVI List. The FHS source then includes any FHS/LHS addressing information it knows locally in the AFP sub-option, i.e., based on information discovered through address resolution.

If the FHS source is the FHS Proxy/Server, it then examines the LHS FMT-Forward code. If FMT-Forward is clear the FHS Proxy/Server sets the NS destination to the ULA of the LHS Proxy/Server; otherwise, it sets the NS destination to the same address as the target. The FHS Proxy/Server then performs OAL encapsulation while setting the OAL source to its own ULA and setting the OAL destination to the FHS Subnet Router Anycast ULA determined by applying the FHS SRT prefix length to its ULA. The FHS Proxy/Server then selects an appropriate Identification value, calculates the OAL checksum, fragments if necessary, encapsulates in appropriate L2 headers then sends the carrier packets into the secured spanning tree which will deliver them to a Gateway interface that assigns the FHS Subnet Router Anycast ULA.

If the FHS source is the FHS Client, it instead includes an authentication signature if necessary. If LHS FMT-Forward is clear, the FHS Client sets the NS destination to the ULA of the LHS Proxy/Server; otherwise, it sets the NS destination to the same address as the target. The FHS Client then calculates the NS message checksum, performs OAL encapsulation, sets the OAL source to its own ULA-MNP and sets the OAL destination to the ULA of the FHS Proxy/Server. The FHS Client finally selects an appropriate Identification value for the FHS Proxy/Server, calculates the OAL checksum, fragments if necessary, encapsulates in appropriate L2 headers then sends the carrier packets to the FHS Proxy/Server.

When the FHS Proxy/Server receives the carrier packets, it discards the L2 headers then verifies the Identification, reassembles if necessary, verifies the OAL checksum and verifies the NS checksum/ authentication signature. The FHS Proxy/Server then creates an AFV (i.e., the same as the FHS Client had done) while caching the AFP

AFVI List "B" entry along with the FHS Client addressing information as previous hop information for this AFV. The FHS Proxy/Server next generates a new locally-unique "B" AFVI, then both assigns it as the AFV index and writes it as the next "B" entry in the AFP AFVI List (while also writing any FHS Client and Proxy/Server addressing information). The FHS Proxy/Server then calculates the NS checksum and sets the OAL source address to its own ULA and destination address to the FHS Subnet Router Anycast ULA. The FHS Proxy/Server finally decrements the OAL Hop Limit, includes an Identification appropriate for the secured spanning tree, calculates the OAL checksum and re-fragments if necessary. The FHS Proxy/Server finally includes appropriate L2 headers and sends the carrier packets into the secured spanning tree.

3.13.2.2. Gateway NS Forwarding

Gateways in the spanning tree forward OAL packets/fragments 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 Gateway receives an OAL packet/fragment over the secured spanning tree addressed to its ULA or the FHS Subnet Router Anycast ULA, it instead reassembles to obtain the NS then verifies the OAL and NS checksums. The FHS Gateway next creates an AFV (i.e., the same as the FHS Proxy/Server had done) while caching the AFP FHS Client and Proxy/Server addressing information, window synchronization information and corresponding AFVI List "B" values in the AFV to enable future reverse path forwarding to this FHS Client. The FHS Gateway then generates a locally-unique "B" AFVI for the AFV and also writes it as the next "B" entry in the NS AFP AFVI List.

The FHS Gateway then examines the SRT prefixes corresponding to both FHS and LHS. If the FHS Gateway has a local interface connection to both the FHS and LHS (whether they are the same or different segments), the FHS/LHS Gateway caches the NS AFP LHS information in the AFV, writes its LHS ULA and L2ADDR into the NS AFP LHS fields, then sets its LHS ULA as the OAL source and the ULA of the LHS Proxy/Server as the OAL destination. If the FHS and LHS prefixes are different, the FHS Gateway instead sets its FHS ULA as the OAL source and the LHS Subnet Router Anycast ULA as the OAL destination. The FHS Gateway then decrements the OAL Hop Limit, selects an appropriate Identification, recalculates the NS and OAL checksums, re-fragments if necessary, then finally includes appropriate L2 headers and sends the carrier packets into the secured spanning tree.

When the FHS and LHS Gateways are different, the LHS Gateway will receive carrier packets over the secured spanning tree from the FHS Gateway, noting there may be many intermediate Gateways in the path between FHS and LHS which will simply forward the enclosed OAL packets/fragments without further processing. The LHS Gateway then reassembles to obtain the NS, verifies the OAL and NS checksums then creates an AFV (i.e., the same as the FHS Gateway had done) while caching the AFP "B" AFVIs and addressing information of previous OAL forwarding hops along with window synchronization information. In particular, the LHS Gateway caches the ULA of the FHS Gateway as the spanning tree address for the previous-hop, caches the LHS information then generates a locally-unique "B" AFVI for the AFV. The LHS Gateway then writes its own LHS ULA and L2ADDR into the AFP sub-option while also writing "B" as the next entry in the AFP AFVI List. The LHS Gateway then sets its own ULA as the OAL source and the ULA of the LHS Proxy/Server as the OAL destination, decrements the OAL Hop Limit, selects an appropriate Identification, recalculates the NS and OAL checksums, re-fragments if necessary, then finally includes appropriate L2 headers and sends the carrier packets into the secured spanning tree.

3.13.2.3. LHS Proxy/Server-Client NS Receipt and NA Forwarding

When the LHS Proxy/Server receives the carrier packets from the secured spanning tree, it discards the L2 headers, reassembles if necessary, verifies the OAL and NS checksums 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 AFV and caches the AFP "B" AFVIs and addressing information of previous OAL forwarding hops the same as for the prior hop. The LHS Proxy/Server next caches the NS window synchronization parameters in the AFV. If the NS destination is the XLA of the LHS Client, the LHS Proxy/Server also generates a locally-unique "B" AFVI and assigns it both to the AFV and as the next "B" entry in the NS AFVI List.

If the NS destination matches its own ULA, the LHS Proxy/Server next prepares to return a solicited NA with Job code ’01’. The LHS Proxy/Server next creates or updates an NCE for the NS source address (if necessary) with state set to STALE and with an AFVI pointer to the new AFV state. When the LHS Proxy/Server forwards future carrier packets based on the cached information, it can populate forwarding information in a CRH-32 routing header to enable forwarding based on the cached AFVI List "B" entries instead of ULA addresses.

The LHS Proxy/Server then creates an NA with Job code ’01’ while copying the NS AFP sub-option into the NA and including responsive window synchronization information. The LHS Proxy/Server then generates a locally-unique "A" AFVI and both assigns it to the AFV and includes it as the first "A" entry in the AFP sub-option AFVI List (see: [I-D.templin-6man-omni] for details on AFVI List A/B processing). The LHS Proxy/Server then encapsulates the NA with OAL source set to its own ULA and OAL destination set to the ULA of the LHS Gateway. The LHS Proxy/Server then selects an appropriate Identification value, calculates the NA and OAL checksums, fragments if necessary then finally includes appropriate L2 headers and forwards the carrier packets into the secured spanning tree.

If the NS destination was the XLA of the LHS Client, the LHS Proxy/ Server includes an authentication signature in the NS if necessary, then recalculates the NS checksum, changes the OAL source to its own ULA and changes the OAL destination to the ULA-MNP of the LHS Client. The LHS Proxy/Server then decrements the OAL Hop Limit, selects an appropriate Identification value, calculates the OAL checksum, fragments if necessary then finally includes appropriate L2 headers and sends the carrier packets to the LHS Client. When the LHS Client receives the carrier packets, it discards the L2 headers, verifies the Identification, reassembles if necessary, then verifies the OAL checksum and NS checksum/authentication signature. The LHS Client then creates a NCE for the NS ULA source address (if necessary) in the STALE state and examines the AFP sub-option. The Client then caches the NS OMNI AFP sub-options in the NCE corresponding to the NS ULA source, then creates an AFV, caches the addressing information and "B" entries of the previous OAL hops then finally generates and assigns a locally-unique "A" AFVI the same as for previous hops. The Client finally caches the new AFVI in the NCE so that future communications can locate the correct AFV.

The LHS Client then prepares an NA using exactly the same procedures as for the LHS Proxy/Server above (while including responsive window synchronization information), except that it uses its XLA as the NA source and the NS source as the NA destination. The LHS Client also includes an authentication signature if necessary, calculates the NA message checksum, then encapsulates the NA with OAL source set to its own ULA-MNP and OAL destination set to the ULA of the LHS Proxy/ Server. The LHS Client finally includes an appropriate Identification, calculates the OAL checksum, fragments if necessary then includes appropriate L2 headers and sends the carrier packets to the LHS Proxy/Server. When the LHS Proxy/Server receives the carrier packets, it discards the L2 headers verifies the Identifications, reassembles if necessary, verifies the OAL checksum and NA checksum/ authentication signature, then uses the current AFP AFVI List "B" entry to locate the AFV. The LHS Proxy/Server then caches the addressing and "A" information for the LHS Client in the AFV, then generates a locally-unique "A" AFVI and both assigns it to the AFV and writes it as the next AFP AFVI List "A" entry. The LHS Proxy/ Server then calculates the NA checksum, sets the OAL source to its own ULA and destination to the ULA of the LHS Gateway, decrements the OAL Hop Limit, includes an appropriate Identification, calculates the

OAL checksum, re-fragments if necessary then finally includes appropriate L2 headers and sends the carrier packets into the secured spanning tree.

3.13.2.4. Gateway NA Forwarding

When the LHS Gateway receives the carrier packets containing the NA message, it discards the L2 headers, reassembles if necessary, verifies the OAL and NA checksums then uses the current NA AFP AFVI List "B" entry to locate the AFV. The LHS Gateway then caches the AFP addressing and AFVI List "A" information for the previous hops in the AFV, then generates a locally-unique "A" AFVI and both assigns it to the AFV and writes it as the next AFP AFVI List "A" entry. The LHS Gateway then recalculates the NA checksum. If the LHS Gateway is connected directly to both the FHS and LHS segments (whether the segments are the same or different), the LHS Gateway will have already cached the FHS/LHS information based on the original NS; the LHS Gateway then sets the OAL source to its FHS ULA and OAL destination to the ULA of the FHS Proxy/Server. Otherwise, the LHS Gateway sets the OAL source to its LHS ULA and OAL destination to the ULA of the FHS Gateway. The LHS Gateway then decrements the OAL Hop Limit, selects an appropriate Identification, recalculates the OAL checksum, re-fragments if necessary, includes appropriate L2 headers and finally sends the carrier packets into the secured spanning tree.

When the FHS and LHS Gateways are different, the FHS Gateway will receive carrier packets containing the NA message from the LHS Gateway over the secured spanning tree, where there may have been many intermediate Gateway forwarding hops. The FHS Gateway then discards the L2 headers, reassembles if necessary, verifies the OAL and NA checksums and locates the AFV based on the current AFP AFVI List "B" entry. The FHS Gateway then caches the addressing and "A" information for the previous hops in the AFV and generates a locally-unique "A" AFVI. The FHS Gateway then assigns the new "A" value to the AFV, records "A" in the AFP AFVI List then writes its FHS ULA and L2ADDR into the AFP FHS Gateway fields. The FHS Gateway then recalculates the NA checksum, sets its FHS ULA as the OAL source and sets the ULA of the FHS Proxy/Server as the OAL destination. The FHS Gateway then decrements the OAL Hop Limit, selects an appropriate Identification value, recalculates the OAL checksum, re-fragments if necessary, includes appropriate L2 headers and finally sends the carrier packets into the secured spanning tree.

3.13.2.5. FHS Proxy/Server-Client NA Receipt

When the FHS Proxy/Server receives the carrier packets from the secured spanning tree, it discards the L2 headers, reassembles if necessary, verifies the OAL and NA checksums then locates the AFV based on the current AFP AFVI List "B" entry. The FHS Proxy/Server then caches the AFP addressing and "A" information for the previous hops. If the NA destination matches its own ULA, the FHS Proxy/Server locates the NCE for the ULA of the LHS Proxy/Server or XLA of the LHS Client and sets the state to REACHABLE. The FHS Proxy/Server then caches the window synchronization parameters and prepares to return an acknowledgement, if necessary.

If the NA destination is the XLA of the FHS Client, the FHS Proxy/Server instead generates a locally-unique "A" AFVI and assigns it both to the AFV and as the next AFP AFVI List "A" entry, then includes an authentication signature/checksum in the NA message. The FHS Proxy/Server then sets the OAL source to its own ULA and sets the OAL destination to the ULA-MNP of the FHS Client. The FHS Proxy/Server then decrements the OAL Hop Limit, selects an appropriate Identification value, recalculates the OAL checksum, re-fragments if necessary, includes appropriate L2 headers and finally sends the carrier packets to the FHS Client.

When the FHS Client receives the carrier packets, it discards the L2 headers, verifies the Identification, reassembles if necessary, verifies the OAL checksum and NA checksum/authentication signature, then locates the AFV based on the current AFP AFVI List "B" entry. The FHS Client then caches the previous hop addressing and "A" information the same as for prior hops. The FHS Client then locates the NCE for the NS source address and sets the state to REACHABLE, then caches the window synchronization parameters and prepares to return an unsolicited NA (uNA) acknowledgement, if necessary.

3.13.2.6. Returning Window Acknowledgements

If either the FHS Client or FHS Proxy/Server needs to return an acknowledgement to complete window synchronization, it prepares a uNA message with an AFP sub-option with Job code set to ’10’ (Follow A; Record B). The FHS node sets the uNA source to its own ULA or XLA, then sets the uNA destination to the ULA or XLA of the LHS node. The FHS node next sets the AFP AFVI List to the cached list of "A" entries received in the Job code ’01’ NA, but need not set any other FHS/LHS information. The FHS node then encapsulates the uNA message in an OAL header with its own ULA as the OAL source. If the FHS node is the Client, it next sets the ULA of the FHS Proxy/Server as the OAL destination, includes an authentication signature/checksum, selects an appropriate Identification value, calculates the OAL checksum, fragments if necessary, includes appropriate L2 headers and finally sends the carrier packets to the FHS Proxy/Server. The FHS Proxy/Server then verifies the Identification, reassembles if necessary, verifies the OAL checksum and uNA checksum/authentication signature, then uses the current AFVI List "A" entry to locate the AFV.

The FHS Proxy/Server then writes its "B" AFVI as the next AFP AFVI List "B" entry, recalculates the uNA checksum then sets its own ULA as the OAL source and the ULA of the FHS Gateway as the OAL destination, The FHS Proxy/Server finally decrements the OAL Hop Limit, selects an appropriate Identification, recalculates the OAL checksum, includes appropriate L2 headers and finally sends the carrier packets into the secured spanning tree. When the FHS Gateway receives the carrier packets, it discards the L2 headers, reassembles if necessary, verifies the OAL and uNA checksums then uses the current AFVI List "A" entry to locate the AFV. The FHS Gateway then writes its "B" AFVI as the next AFP AFVI List "B" entry, then sets the OAL source to its own ULA. If the FHS Gateway is also the LHS Gateway, it sets the OAL destination to the ULA of the LHS Proxy/Server; otherwise it sets the OAL destination to the ULA of the LHS Gateway. The FHS Gateway recalculates the uNA checksum then decrements the OAL Hop Limit, selects an appropriate Identification, recalculates the OAL checksum, re-fragments if necessary, includes appropriate L2 headers and finally sends the carrier packets into the secured spanning tree. If an LHS Gateway receives the carrier packets, it processes them exactly the same as the FHS Gateway had done while re-setting the OAL destination to the ULA of the LHS Proxy/Server.

When the LHS Proxy/Server receives the carrier packets, it discards the L2 headers, verifies the Identification, reassembles if necessary then verifies the OAL and uNA checksums. The LHS Proxy/Server then locates the AFV based on the current AFP AFVI List "A" entry. If the uNA destination matches its own ULA, the LHS Proxy/Server next updates the NCE/AFV for the source ULA based on the uNA window synchronization parameters and MAY compare the AFVI List to the version it had cached in the AFV based on the original NS.

If the uNA destination is the XLA of the LHS Client, the LHS Proxy/Server instead writes its "B" AFVI as the next AFP AFVI List "B" entry and includes an authentication signature/checksum. The LHS Proxy/Server then writes its own ULA as the OAL source and the ULA-MNP of the Client as the OAL destination, then decrements the OAL Hop Limit, selects an appropriate Identification and recalculates the OAL checksum. The LHS Proxy/Server finally re-fragments if necessary, includes appropriate L2 headers and sends the resulting carrier packets to the LHS Client. When the LHS Client receives the carrier packets, it discards the L2 headers, verifies the Identification, reassembles if necessary, verifies the OAL checksum and uNA checksum/authentication signature then processes the message exactly the same as for the LHS Proxy/Server case above.

Note: If either the LHS Client or LHS Proxy/Server needs to return an acknowledgement to complete window synchronization, it prepares a uNA message with an AFP sub-option with Job code set to '11' (Follow B; Record A). All other procedures are exactly the opposite as per the FHS case specified above.

3.13.2.7. OAL End System Exchanges Following Synchronization

Following the initial NS/NA exchange with AFP sub-options, OAL end systems can begin exchanging ordinary carrier packets that include "A/B" AFVIs and with Identification values within their respective send/receive windows without requiring security signatures and/or secured spanning tree traversal. OAL end systems and intermediate nodes can also consult their AFIBs when they receive carrier packets that contain OAL packets/fragments with "A/B" AFVIs to unambiguously locate the correct AFV and can use any discovered "A/B" values of other OAL nodes to forward OAL packets/fragments to nodes that configure the corresponding AFVIs. OAL end systems must then perform continuous NS/NA exchanges to update window state, register new interface pairs for optimized multilink forwarding, confirm reachability and/or refresh AFIB cache state in the path before ReachableTime expires.

While the OAL end systems continue to actively exchange OAL packets, they are jointly responsible for updating cache state and per-interface reachability before expiration. Window synchronization state is performed on a per-interface-pair basis and tracked in the AFVs which are also linked to the appropriate NCE. However, the window synchronization exchange only confirms target Client reachability over the specific underlay interface pair. Reachability for other underlay interfaces that share the same window synchronization state must be determined individually using additional NS/NA messages.

To update AFIB state in the path, the FHS node that sent the original NS message with AFP Job code '00' can send additional NS messages with AFP sub-options with Job code '10' (Follow "A"; Record "B") and with window synchronization parameters. The message will be processed by all intermediate OAL nodes which will refresh AFV timers, cache window synchronization parameters and forward the NS onward toward the LHS node that returned the original NA message. When the LHS node receives the NS, it returns an NA message with AFP Job code '11' (Follow "B"; Record "A").

At the same time, the LHS node that received the original NS message with Job code ’00’ can send additional NS messages with Job code ’11’ in order to cause the FHS node to return an NA message with AFP Job code ’10’. The process can therefore be coordinated asynchronously with the FHS/LHS nodes initiating an NS/NA exchange independently of one another. The exchanges will succeed as long as the AFIB state in the path remains active. Note that all intermediate node processing of Job code ’10’ and ’11’ NS/NA messages is conducted the same as for the initial NS/NA exchange according to the detailed specifications above.

OAL sources can also begin including CRH-32s in OAL packets/fragments with a list of "A/B" AFVIs that OAL intermediate nodes can use for shortest-path forwarding based on AFVIs instead of spanning tree addresses. OAL sources and intermediate nodes can instead forward OAL packets/fragments with OAL compressed headers termed "OCH" (see: [I-D.templin-6man-omni]) that include only a single "A/B" AFVI meaningful to the next hop, since all OAL nodes in the path up to (and sometimes including) the OAL destination have already established AFVs. Note that when an FHS OAL source receives a solicited NA with Job code ’01’, the AFP sub-option will contain an AFVI 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 AFP AFVI List "A" entries last-to-first when it populates a CRH-32, or must select the correct "A" entry to include in an OCH header based on the intended OAL intermediate node or destination.

When a Gateway receives unsecured carrier packets that contain OAL packets/fragments destined to a local SRT segment Client that has asserted direct reachability, the Gateway performs direct forwarding while bypassing the local Proxy/Server based on the Client’s advertised AFVIs and discovered NATed L2ADDR information (see: Section 3.13.3). If the Client cannot be reached directly (or if NAT traversal has not yet converged), the Gateway instead forwards OAL packets/fragments directly to the local segment Proxy/Server.

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

When a source Client forwards OAL packets/fragments it can employ header compression according to the AFVIs established through an NS/NA exchange with a remote or local peer. When the source Client forwards to a remote peer, it can forward OAL packets/fragments to a local SRT Gateway (following the establishment of L2ADDR information) while bypassing the Proxy/Server following route optimization (see: Section 3.13.3). When a target Client receives carrier packets that contain OAL packets/fragments that match a local AFV, 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 original IP packet/parcel 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, they can exchange carrier packets that contain OAL packets/fragments directly with header compression using AFVIs discovered as above (see: Section 3.13.4). The FHS Client will have cached the "A" AFVI for the LHS Client, which will have cached the "B" AFVI for the FHS Client.

When the FHS Client or FHS Proxy/Server sends an NS for the purpose of establishing multilink forwarding state, it should wait up to RETRANS_TIMER seconds to receive a responsive NA. The FHS node can then retransmit the NS up to MAX_UNICAST_SOLICIT times before giving up. Note that each successive attempt establishes new AFV state in the OAL intermediate nodes, but that any abandoned stale AFV state will be quickly reclaimed.

3.13.2.8. Rapid Commit Multilink Forwarding

Multilink forwarding can often be invoked simultaneously with Address Resolution in order to reduce control message overhead and round-trip delays. When an ART acting as an ARR receives an NS(AR) with a set of Interface Attributes for the ARS source Client, it can perform "rapid commit" by immediately invoking multilink forwarding as above at the same time as returning the NA(AR).

In order to perform rapid commit, the ARR includes an AFP sub-option with Job code ’00’ and a Window Synchronization sub-option as though it were initiating a multilink coordination NS/NA exchange as specified above. The ARR then includes any Interface Attributes and/or Traffic Selector sub-options as necessary to satisfy the address resolution request, and can also include ordinary original IP packets/parcels as additional super-packet extensions to this NA(AR) message if it has immediate data to send to the ARS. The ARR then returns the NA(AR) to the ARS using the same hop-by-hop OAL addressing disciplines as specified above for an ordinary multilink NS/NA exchange. This will cause the NA(AR) to visit all OAL intermediate nodes on the path towards the ARS.

When the NA(AR) traverses the return path to the ARS, OAL intermediate nodes in the path process the NS AFP information exactly the same as for an ordinary multilink forwarding exchange as specified above, i.e., without examining the remaining NA(AR) message contents. This results in the ARR node now assuming the FHS role and the ARS assuming the LHS role from the perspective of multilink forwarding coordination. When the NA(AR) arrives, the ARS processes the AFP and window synchronization parameters while also processing all other NA(AR) OMNI option information, thereby eliminating an extraneous message transmission and associated delay. The ARS (now acting as an LHS peer) then completes the exchange by returning a responsive NA with an AFP sub-option with Job code ’01’; if no NA response is received within RETRANS_TIMER seconds, the ARR can retransmit the NA(AR) up to MAX_NEIGHBOR_ADVERTISEMENT times before giving up.

This very importantly implies that the type of IPv6 ND message used to convey an AFP with Job codes ’00’ and ’01’ (i.e., NS or NA) is unimportant from the perspective of multilink forwarding. This means that Job code ’00’ serves as the solicitation indication and Job code ’01’ serves as the response such that either an NS or NA message carrying an AFP with Job code ’00’ will invoke a responsive NA message carrying an AFP with Job code ’01’.

3.13.3. Client/Gateway Route Optimization

Following multilink route optimization for specific underlay interface pairs, FHS/LHS Clients located on open INETs can invoke Client/Gateway route optimization to improve performance and reduce load and congestion on their respective Proxy/Servers. To initiate Client/Gateway route optimization, the Client prepares an NS message with its own XLA address as the source and the ULA of its Gateway as the destination while creating a NCE for the Gateway if necessary. The NS message must be no larger than the minimum MPS and encapsulated as an atomic fragment.

The Client then includes an Interface Attributes sub-option for its underlay interface as well as an authentication signature but does not include window synchronization parameters. The Client then performs OAL encapsulation with its own ULA-MNP as the source and the ULA of the Gateway as the destination while including a randomly-chosen Identification value, then performs L2 encapsulation on the atomic fragment and sends the resulting carrier packet directly to the Gateway.

When the Gateway receives the carrier packet, it removes the L2 headers, verifies the NS checksum/authentication signature then creates a NCE for the Client. The Gateway then caches the L2 encapsulation addresses (which may have been altered by one or more NATs on the path) as well as the Interface Attributes for this Client ifIndex, and marks this Client underlay interface as "trusted". The Gateway then prepares an NA reply with its own ULA as the source and the XLA of the Client as the destination where the NA again must be no larger than the minimum MPS.

The Gateway then echoes the Client’s Interface Attributes, includes an Origin Indication with the Client’s observed L2 addresses and includes an authentication signature. The Gateway then performs OAL encapsulation with its own ULA as the source and the ULA-MNP of the Client as the destination while using the same Identification value that appeared in the NS, then performs L2 encapsulation on the atomic fragment and sends the resulting carrier packet directly to the Client.

When the Client receives the NA reply, it caches the carrier packet L2 source address information as the Gateway target address via this underlay interface while marking the interface as "trusted". The Client also caches the Origin Indication L2 address information as its own (external) source address for this underlay interface.

After the Client and Gateway have established NCEs as well as "trusted" status for a particular underlay interface pair, each node can begin sending ordinary carrier packets intended for this multilink route optimization directly to one another while omitting the Proxy/Server from the forwarding path while the status is "trusted". The NS/NA messaging will have established the correct state in any NATs in the path so that NAT traversal is naturally supported. The Client and Gateway must maintain a timer that watches for activity on the path; if no carrier packets and/or NS/NA messages are sent or received over the path before NAT state is likely to have expired, the underlay interface pair status becomes "untrusted".

Thereafter, when the Client sends a carrier packet that contains an OAL packet/fragment toward the Gateway as the next hop, the Client includes the AFVI for the Gateway (discovered during multilink route optimization) instead of the AFVI for its Proxy/Server; the Gateway will accept the OAL packet/fragment from the Client if and only if the AFVI matches the correct AFV and the underlay interface status is trusted. (The same is true in the reverse direction when the Gateway sends carrier packets directly to the Client.)

Note that the Client and Gateway each maintain a single NCE, but that the NCE may aggregate multiple underlay interface pairs. Each underlay interface pair may use differing source and target L2 addresses according to NAT mappings, and the "trusted/untrusted" status of each pair must be tested independently. When no "trusted" pairs remain, the NCE is deleted.

Note that the above method requires Gateways to participate in NS/NA message authentication signature application and verification. In an alternate approach, the Client could instead exchange NS/NA messages with authentication signatures via its Proxy/Server but addressed to the ULA of the Gateway, and the Proxy/Server and Gateway could relay the messages over the secured spanning tree. However, this would still require the Client to send additional messages toward the L2 address of the Gateway to populate NAT state; hence the savings in complexity for Gateways would result in increased message overhead for Clients.

3.13.4. Client/Client Route Optimization

When the FHS/LHS Clients are both located on the same SRT segment, Client-to-Client route optimization is possible following the establishment of any necessary state in NATs in the path. Both Clients will have already established state via their respective shared segment Proxy/Servers (and possibly also the shared segment Gateway) and can begin sending carrier packets directly via NAT traversal while avoiding any Proxy/Server and/or Gateway hops.

When the FHS/LHS Clients on the same SRT segment perform the initial NS/NA exchange to establish AFIB state, they first examine the FMT-Forward and FMT-Mode settings to determine whether direct-path forwarding is even possible for one or both Clients (direct-path forwarding is only possible for one or both when FMT-Forward and FMT-Mode are both 1). The NS/NA messages then include an Origin Indication (i.e., in addition to an AFP sub-option) with the mapped addresses discovered during the RS/RA exchanges with their respective Proxy/Servers. After the AFV paths have been established, both Clients can begin sending carrier packets via strict AFV paths while establishing a direct path for Client-to-Client route optimization.

To establish the direct path, either Client (acting as the source) transmits a bubble to the mapped L2 address for the target Client which primes its local chain of NATs for reception of future carrier packets from that L2 address (see: [RFC4380] and [I-D.templin-6man-omni]). The source Client then prepares an NS message with its own XLA as the source, with the XLA of the target as the destination and with an OMNI option with an Interface Attributes sub-option. The source Client then encapsulates the NS in an OAL header with its own ULA-MNP as the source, with the ULA-MNP of the target Client as the destination and with an in-window Identification for the target. The source Client then fragments and encapsulates in L2 headers addressed to its Proxy/Server then sends the resulting carrier packets to the Proxy/Server.

When the Proxy/Server receives the carrier packets, it re-encapsulates and sends them as unsecured carrier packets according to AFIB state where they will eventually arrive at the target Client which can verify that the identifications are within the acceptable window and reassemble if necessary. Following reassembly, the target Client prepares an NA message with its own XLA as the source, with the XLA of the source Client as the destination and with an OMNI option with an Interface Attributes sub-option. The target Client then encapsulates the NA in an OAL header with its own ULA-MNP as the source, with the ULA-MNP of the source Client as the destination and with an in-window Identification for the source Client. The target Client then fragments and encapsulates in L2 headers addressed to the source Client’s Origin addresses then forwards the resulting carrier packets directly to the source Client.

Following the initial NS/NA exchange, both Clients mark their respective (source, target) underlay interface pairs as "trusted" for no more than ReachableTime seconds. While the Clients continue to exchange carrier packets via the direct path avoiding all Proxy/Servers and Gateways, they should perform additional NS/NA exchanges via their local Proxy/Servers to refresh NCE state as well as send additional bubbles to the peer's Origin address information if necessary to refresh NAT state.

Note that these procedures are suitable for a widely-deployed but basic class of NATs. Procedures for advanced NAT classes are outlined in [RFC6081], which provides mechanisms that can be employed equally for AERO using the corresponding sub-options specified by OMNI.

Note also that each communicating pair of Clients may need to maintain NAT state for peer to peer communications via multiple underlay interface pairs. It is therefore important that Origin Indications are maintained with the correct peer interface and that the NCE may cache information for multiple peer interfaces.

Note that the source and target Client exchange Origin information during the secured NS/NA multilink route optimization exchange. This allows for subsequent NS/NA exchanges to proceed using only the Identification value as a data origin confirmation. However, Client-to-Client peerings that require stronger security may also include authentication signatures for mutual authentication.

3.13.5. Client-to-Client OMNI Link Extension

Clients may be recursively nested within the ENETs of other Clients. When a Client is the downstream-attached ENET neighbor of an upstream Client, it still supports the route optimization functions discussed above by maintaining an AFIB and assigning AFVI values. When the Client processes an IPv6 ND NS/NA message that includes an AFP sub-option, it writes its AFVI information as the first/last AFVI list entry the same as for the single Client case discussed above.

The Client then forwards the NS/NA message to the next Client in the extended OMNI link toward the FHS/LHS Proxy/Server, which records the AFVI value then overwrites the AFVI list entry with its own AFVI value. This process iteratively continues until the Client that will forward the NS/NA message to the FHS/LHS Proxy/Server is reached, at which point the NS/NA AFVI list entries are populated by the intermediate nodes on the path to the LHS/FHS the same as discussed above.

In this way, each Client in the extended OMNI link discovers the A/B AFVIs of the next/previous Client without intruding into the AFP sub-option AFVI list. Therefore the list can remain fixed at 5 entries even though the Client-to-Client OMNI link extension can be arbitrarily long. Therefore, route optimization is not possible between consecutive Client members of the extended OMNI link but becomes possible at the Internetworking border that separates the FHS and LHS elements.

3.13.6. Intra-ANET/ENET Route Optimization for AERO Peers

When a Client forwards an OAL packet (or original IP packet/parcel) from a Host or another Client connected to one of its downstream ENETs to a peer within the same downstream ENET, the Client returns an IPv6 ND Redirect message to inform the source that that target can be reached directly. The contents of the Redirect message are the same as specified in [RFC4861], and should also include a Prefix Length sub-option with the length of the MNP found in the Target Address field.

In the same fashion, when a Proxy/Server forwards an OAL packet (or original IP packet/parcel) from a Host or Client connected to one of its downstream ANETs to a peer within the same downstream ANET, the Proxy/Server returns an IPv6 ND Redirect message.

All other route optimization functions are conducted per the NS/NA messaging discussed in the previous sections.

3.14. 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 exchanges over the OMNI link secured spanning tree (i.e. the same as described above) to test reachability without risk of DoS attacks from nodes pretending to be a neighbor. These NS/NA messages use the unicast XLAs/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 exchanges directly between neighbors without employing the secured spanning tree as long as they include in-window Identifications and an authentication signature/checksum.

After route optimization directs a source FHS peer to a target LHS peer with one or more link-layer addresses, either node may invoke multilink forwarding state initialization to establish authentic intermediate node state between specific underlay interface pairs which also tests their reachability. Thereafter, either node acting as the source may perform additional reachability probing through NS messages over the SRT secured or unsecured spanning tree, or through NS messages sent directly to an underlay interface of the target itself. While testing a target underlay interface, the source can optionally continue to forward OAL packets/fragments via alternate interfaces, maintain a small queue of carrier packets until target reachability is confirmed or include them as trailing data with the NS in an OAL super-packet [I-D.templin-6man-omni].

NS messages are encapsulated, fragmented and transmitted as carrier packets the same as for ordinary original IP data packets/parcels, however the encapsulated destinations are either the ULA or XLA of the source and either the ULA of the LHS Proxy/Server or the XLA of the target itself. The source encapsulates the NS message the same as described in Section 3.13.2 and includes an Interface Attributes sub-option with ifIndex set to identify its underlay interface used for forwarding. The source then includes an in-window Identification, fragments the OAL packet, includes L2 encapsulations and sends the resulting carrier packets into the unsecured spanning tree, either directly to the target if it is in the local segment or directly to a Gateway in the local segment.

When the target receives the NS carrier packets, it discards the L2 headers, verifies that it has a NCE for this source and that the Identification is in-window then reassembles if necessary. The target next verifies the NS checksum/authentication signature, then searches for Interface Attributes in its NCE for the source that match the NS for the NA reply. The target then prepares the NA with the source and destination addresses reversed, encapsulates and sets the OAL source and destination, includes an Interface Attributes sub-option in the NA to identify the ifIndex of the underlay interface the NS arrived on and sets the Target Address to the same value included in the NS. 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 source and performs fragmentation. The node then performs L2 encapsulation and sends the carrier packets into the unsecured spanning tree either directly to the source if it is in the local segment or directly to a Gateway in the local segment.

When the source receives the NA, it marks the target underlay interface tested as "trusted". Note that underlay interface states are maintained independently of the overall NCE REACHABLE state, and that a single NCE may have multiple target underlay interfaces in various "trusted/untrusted" states while the NCE state as a whole remains REACHABLE.

3.15. Mobility Management and Quality of Service (QoS)

AERO is a fully Distributed Mobility Management (DMM) service in which each Proxy/Server is responsible for only a small subset of the Clients on the OMNI link. This is in contrast to a Centralized Mobility Management (CMM) service where there are only one or a few network mobility collective entities for large Client populations. 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 a designated router service for their dependent Clients, while FHS Proxy/Servers provide a proxy conduit between the Client and both the Hub and OMNI link in general. 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 underlay interface attributes change, the Client is responsible for updating the Hub Proxy/Server through new RS/RA exchanges 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 underlay interface.

Note: when a Client’s underlay interface address changes, the Client and/or its (former) FHS Proxy/Server for this interface must invalidate any AFVs based on the (changed) interface. Future data packet forwarding will then trigger a new multilink forwarding NS/NA exchange to re-seed new AFVs in the path.

Mobility management considerations are specified in the following sections.

3.15.1. Mobility Update Messaging

Mobile Clients (and/or their Hub Proxy/Servers) accommodate mobility and/or multilink change events by sending secured uNA messages to each active neighbor. When a node sends a uNA message to each specific neighbor on behalf of a mobile Client, it sets the IPv6 source address to its own ULA or XLA, sets the destination address to the neighbor's ULA or XLA and sets the Target Address to the mobile Client's XLA. The uNA also includes an OMNI option with OMNI Interface Attributes and Traffic Selector sub-options for the mobile Client's underlay interfaces and includes an authentication signature if necessary. The node 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 ULA and destination set to either the specific neighbor's ULA or the FHS Proxy/Server's ULA. The uNA message will then follow the secured spanning tree and arrive at the specific neighbor.

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 node can optionally send up to MAX_NEIGHBOR_ADVERTISEMENT uNAs to each neighbor to increase the likelihood that at least one will be received. Alternatively, the node can set the PNG flag in the uNA OMNI option header to request a uNA acknowledgement as specified in [I-D.templin-6man-omni].

When the FHS/LHS Proxy/Server receives a secured uNA message prepared as above, if the uNA destination was its own ULA the Proxy/Server uses the included OMNI option information to update its NCE for the target but does not reset ReachableTime since the receipt of a uNA message does not provide confirmation that any forward paths to the target Client are working. If the destination was the XLA of the FHS/LHS Client, the Proxy/Server instead changes the OAL source to its own ULA, includes an authentication signature if necessary, and includes an in-window Identification for this Client. Finally, if the uNA message PNG flag was set, the node that processes the uNA returns a uNA acknowledgement as specified in [I-D.templin-6man-omni].

3.15.2. Announcing Link-Layer Information Changes

When a Client needs to change its underlay Interface Attributes and/or Traffic Selectors for one or more underlay interfaces (e.g., due to a mobility event), the Client sends RS messages to its Hub Proxy/Server (via first-hop FHS Proxy/Servers if necessary). Each RS includes an OMNI option with Interface Attributes and/or Traffic Selector sub-options for the ifIndex in question.

Note that the first FHS Proxy/Server may change due to the underlay interface change. If the Client RS includes an OMNI Proxy/Server Departure sub-option for the former FHS Proxy/Server, the new FHS Proxy/Server can send a departure indication (see Section 3.15.5); otherwise, any stale state in the former FHS Proxy/Server 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.

After performing the RS/RA exchange, the Client sends uNA messages to all neighbors the same as described in the previous section.

3.15.3. Bringing New Links Into Service

When a Client needs to bring new underlay 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 underlay interface (if necessary) with an OMNI option that includes an Interface Attributes sub-option with appropriate link quality values and with link-layer address information for the new link. The Client then again sends uNA messages to all neighbors the same as described above.

3.15.4. Deactivating Existing Links

When a Client needs to deactivate an existing underlay interface, it sends a uNA message toward the Hub Proxy/Server via an FHS Proxy/Server with an OMNI option with appropriate Interface Attributes 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 uNA messages over an underlay interface other than the one being deactivated, it MUST include Interface Attributes with appropriate link quality values for any underlay interfaces being deactivated. The Client then again sends uNA messages to all neighbors the same as described above.

Note that when a Client deactivates an underlay interface, neighbors that receive the ensuing uNA messages need not purge all references for the underlay interface from their neighbor cache entries. The Client may reactivate or reuse the underlay interface and/or its ifIndex 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.

3.15.5. 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 a Client associates with a new Hub Proxy/Server, it sends RS messages to register its underlay interfaces with the new Hub while including the old Hub’s ULA in the "Old Hub Proxy/Server ULA" field of a Proxy/Server Departure OMNI sub-option. When the new Hub Proxy/Server returns the RA message via the FHS Proxy/Server (acting as a proxy), the FHS Proxy/Server sends a uNA to the old Hub Proxy/Server (i.e., if the ULA is non-zero and different from its own). The uNA has the XLA of the Client as the source and the ULA of the old hub as the destination and with an OMNI Proxy/Server Departure sub-option as above. The FHS Proxy/Server encapsulates the uNA in an OAL header with the ULA of the new Hub as the source and the ULA of the old Hub as the destination, the fragments, performs L2 encapsulation and sends the resulting carrier packets via the secured spanning tree.

When the old Hub Proxy/Server receives the carrier packets, it decapsulates and reassembles if necessary to obtain the uNA then changes the Client’s NCE state to DEPARTED, resets DepartTime and caches the new Hub Proxy/Server ULA. After a short delay (e.g., 2 seconds) the old Hub Proxy/Server withdraws the Client’s MNP from the routing system. While in the DEPARTED state, the old Hub Proxy/Server forwards any carrier packets received via the secured spanning tree destined to the Client’s ULA-MNP to the new Hub Proxy/Server’s ULA. After DepartTime expires, the old Hub Proxy/Server deletes the Client’s NCE.

Mobility events may also cause a Client to change to a new FHS Proxy/Server over a specific underlay interface at any time such that a Client RS/RA exchange over the underlay interface will engage the new

FHS Proxy/Server instead of the old. The Client can arrange to inform the old FHS Proxy/Server of the departure by including a Proxy/Server Departure sub-option with a ULA for the "Old FHS Proxy/ Server ULA", and the new FHS Proxy/Server will issue a uNA using the same procedures as outlined for the Hub above while using its own ULA as the source address. This can often result in successful delivery of carrier packets that would otherwise be lost due to the mobility event.

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 become unresponsive, topological movements of significant distance, movement to a new geographic region, movement to a new OMNI link segment, etc.

3.16. Multicast

Clients provide an IGMP (IPv4) [RFC2236] or MLD (IPv6) [RFC3810] proxy service for its ENETs 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 underlay 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 FHS Proxy/Server then acts as an ARS to send NS(AR) messages to an ARR for the multicast source. Clients on INET and ANET underlay interfaces without native multicast services instead send NS(AR) messages as an ARS to cause their FHS Proxy/Server to forward the message to an ARR. When the ARR prepares 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.

3.16.1. Source-Specific Multicast (SSM)

When an ARS "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 3.13) using its own ULA or XLA as the source address, the solicited node multicast address corresponding to S as the destination and the XLA 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 ARR "Y" that services S. Y will then return an NA(AR) that includes an OMNI option with Interface Attributes for any underlay interfaces that are currently servicing S.

When X processes the NA(AR) it selects one or more underlay interfaces for S and performs an NS/NA multilink forwarding 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 XLA of X as the next hop in the reverse path. Since Gateways 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 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 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 3.15. When X receives the uNA message, it updates its NCE for the XLA for source S and sends new Join messages in NS/NA exchanges addressed to the new target Client underlay 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.

3.16.2. Any-Source Multicast (ASM)

When an ARS "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 address resolution 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 message with its own ULA or XLA as the source address and the ULA or XLA for R as the destination address, then encapsulates the NS message in an OAL header with its own ULA as the source and the 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/parcels in PIM Register messages, includes the PIM Register messages in the OMNI options of uNA messages, performs OAL encapsulation and fragmentation with Identification values within the receive window for Client R* that aggregates R, then performs L2 encapsulation and sends the resulting carrier packets. 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/parcel; 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/parcels 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 3.16.1. Once the (S,G) tree is established, the listeners can send (S, G) Prune messages to R so that multicast original IP packets/parcels 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.

3.16.3. 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.

3.17. 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 Gateways 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 3.2.4).

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/parcels 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 Gateways.

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 Gateways on the link. Correspondent nodes can then perform Segment Routing to select the correct SRT, which will then direct the original IP packet/parcel over multiple hops to the target.

3.18. 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/parcels 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.

3.19. 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 Gateways on each INET partition, with each Gateway 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 Gateways.

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

3.20. Proxy/Server-Gateway Bidirectional Forwarding Detection

In environments where rapid failure recovery is required, Proxy/Servers and Gateways 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 Gateways maintain BFD sessions in parallel with their BGP peerings. If a Proxy/Server or Gateway fails, BGP peers will quickly re-establish routes through alternate paths the same as for common BGP deployments.

3.21. 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 XLA) 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.

Many AERO/OMNI functions are implemented and undergoing final integration. OAL fragmentation/reassembly buffer management code has been cleared for public release.

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 Gateways configure underlay interface 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 Gateways of all OMNI link segments in turn configure underlay interface secured tunnels with neighboring AERO Gateways for other OMNI link segments 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. (Note that this inter-segment Gateway arrangement mirrors the "half-gateway" model discussed in the original Catenet proposal.)

To prevent unauthorized local applications from congesting the secured spanning tree, Proxy/Servers and Gateways should configure local firewall settings to permit only the BGP protocol service daemon to source routing protocol control messages with the ULA assigned to the OMNI interface as the source and the ULA of a neighboring Proxy/Server or Gateway as the destination. This could be implemented as a port/address filtering configuration that permits only TCP port 179 (as defined in the IANA "Service Names and Port Numbers" registry) when using the ULA assigned to the OMNI interface. To prevent malicious Clients from congesting the secured spanning tree, Proxy/Servers should also rate-limit the secured IPv6 ND NS/NA messages they process for the same (source, target) pair, e.g., by applying IPv6 ND MAX_UNICAST_SOLICIT; MAX_NEIGHBOR_ADVERTISEMENT limits. This is especially true for NS/NA messages that include ordinary original IP data packets/parcels as part of a super-packet.

To prevent spoofing vectors, Proxy/Servers MUST discard without responding to any unsecured IPv6 ND messages that include OMNI sub-options that would affect state. Also, Proxy/Servers MUST discard without forwarding any original IP packets/parcels 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 sending any carrier packets that include an OAL packet/fragment with 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 the secured spanning tree, 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, or to use the QUIC-TLS protocol to establish a secured connection [RFC9000][RFC9001][RFC9002].

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 secured 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 QUIC-TLS, 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 Gateways 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 Gateways 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 carrier 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 ENETs 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, all OAL nodes SHOULD employ Identification window synchronization and OAL end systems SHOULD configure an (end-system-based) firewall.

SRH authentication facilities are specified in [RFC8754]. Security considerations for accepting link-layer ICMP messages and reflected carrier 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, Eliot Lear, 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 Akash Agarwal, Kyle Bae, M. Wayne Benson, Dave Bernhardt, Cam Brodie, John Bush, Balaguruna Chidambaram, Irene Chin, Bruce Cornish, Claudiu Danilov, Sean Dickson, 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, Satish Raghavendran, Vijay Rajagopalan, Kristina Ross, Greg Saccone, Bhargava Raman Sai Prakash, Rod Santiago, Madhanmohan Savadamuthu, 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. Akash Agarwal, Kyle Bae, Wayne Benson, Madhuri Madhava Badgandi, Vijayasarathy Rajagopalan, Bhargava Raman Sai Prakash, Katie Tran and Eric Yeh are especially acknowledged for their work on the AERO implementation. Chuck Klabunde is honored for his support and guidance, 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 proxys. 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., "Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces", Work in Progress, Internet-Draft, draft-templin-6man-omni-73, 11 October 2022, <https://www.ietf.org/archive/id/draft-templin-6man-omni-73.txt>.

[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.

[EUI] IEEE, I., "Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID, https://standards.ieee.org/wp-content/uploads/import/documents/tutorials/eui.pdf", 3 August 2017.

[I-D.bonica-6man-comp-rtg-hdr] Bonica, R., Kamite, Y., Alston, A., Henriques, D., and L. Jalil, "The IPv6 Compact Routing Header (CRH)", Work in Progress, Internet-Draft, draft-bonica-6man-comp-rtg-hdr-28, 18 May 2022, <https://www.ietf.org/archive/id/draft-bonica-6man-comp-rtg-hdr-28.txt>.

[I-D.bonica-6man-crh-helper-opt] Li, X., Bao, C., Ruan, E., and R. Bonica, "Compressed Routing Header (CRH) Helper Option", Work in Progress, Internet-Draft, draft-bonica-6man-crh-helper-opt-04, 11 October 2021, <https://www.ietf.org/archive/id/draft-bonica-6man-crh-helper-opt-04.txt>.

[I-D.ietf-intarea-frag-fragile]
Bonica, R., Baker, F., Huston, G., Hinden, R. M., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", Work in Progress, Internet-Draft, draft-ietf-intarea-frag-fragile-17, 30 September 2019, <https://www.ietf.org/archive/id/draft-ietf-intarea-frag-fragile-17.txt>.

[I-D.ietf-intarea-tunnels]
Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-10, 12 September 2019, <https://www.ietf.org/archive/id/draft-ietf-intarea-tunnels-10.txt>.

[I-D.ietf-ipwave-vehicular-networking]
Jeong, J. P., "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases", Work in Progress, Internet-Draft, draft-ietf-ipwave-vehicular-networking-29, 19 May 2022, <https://www.ietf.org/archive/id/draft-ietf-ipwave-vehicular-networking-29.txt>.

[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", Work in Progress, Internet-Draft, draft-ietf-rtgwg-atn-bgp-18, 14 June 2022, <https://www.ietf.org/archive/id/draft-ietf-rtgwg-atn-bgp-18.txt>.

[I-D.templin-6man-dhcpv6-ndopt]
Templin, F. L., "A Unified Stateful/Stateless Configuration Service for IPv6", Work in Progress, Internet-Draft, draft-templin-6man-dhcpv6-ndopt-11, 1 January 2021, <https://www.ietf.org/archive/id/draft-templin-6man-dhcpv6-ndopt-11.txt>.

[I-D.templin-intarea-parcels]
Templin, F., "IP Parcels", Work in Progress, Internet-Draft, draft-templin-intarea-parcels-16, 6 October 2022, <https://www.ietf.org/archive/id/draft-templin-intarea-parcels-16.txt>.

[I-D.templin-intarea-seal]
Templin, F. L., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", Work in Progress, Internet-Draft, draft-templin-intarea-seal-68, 3 January 2014, <https://www.ietf.org/archive/id/draft-templin-intarea-seal-68.txt>.

[I-D.templin-intarea-vet]
Templin, F. L., "Virtual Enterprise Traversal (VET)", Work in Progress, Internet-Draft, draft-templin-intarea-vet-40, 3 May 2013, <https://www.ietf.org/archive/id/draft-templin-intarea-vet-40.txt>.

[I-D.templin-ipwave-uam-its]
Templin, F. L., "Urban Air Mobility Implications for Intelligent Transportation Systems", Work in Progress, Internet-Draft, draft-templin-ipwave-uam-its-04, 4 January 2021, <https://www.ietf.org/archive/id/draft-templin-ipwave-uam-its-04.txt>.

[I-D.templin-ironbis]
Templin, F. L., "The Interior Routing Overlay Network (IRON)", Work in Progress, Internet-Draft, draft-templin-ironbis-16, 28 March 2014, <https://www.ietf.org/archive/id/draft-templin-ironbis-16.txt>.

[I-D.templin-v6ops-pdhost]
Templin, F. L., "IPv6 Prefix Delegation and Multi-Addressing Models", Work in Progress, Internet-Draft, draft-templin-v6ops-pdhost-27, 1 January 2021, <https://www.ietf.org/archive/id/draft-templin-v6ops-pdhost-27.txt>.

[IEN48] Cerf, V., "The Catenet Model For Internetworking, https://www.rfc-editor.org/ien/ien48.txt", July 1978.

[IEN48-2] Cerf, V., "The Catenet Model For Internetworking (with figures), http://www.postel.org/ien/pdf/ien048.pdf", July 1978.

[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>.

[RFC1256] Deering, S., Ed., "ICMP Router Discovery Messages", RFC 1256, DOI 10.17487/RFC1256, September 1991, <https://www.rfc-editor.org/info/rfc1256>.

[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>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[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>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/info/rfc9001>.

[RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <https://www.rfc-editor.org/info/rfc9002>.

[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

Address resolution and route optimization as discussed in Section 3.13 results in the creation of NCEs. 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 messages 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 messages for (ReachableTime - 5) seconds. If any data messages have been sent to the neighbor within this timeframe, then send an NS(AR) to receive a new NA(AR). If no data messages 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 messages are sent within the 5 second window, reset the NCE state to STALE.

The monitoring of the neighbor data traffic therefore becomes an ongoing process during the NCE lifetime. If the NCE expires, future data messages will trigger a new NS/NA(AR) exchange while the messages 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 carrier packets over a single interface at a time, and the neighbor always observes carrier packets arriving from the Client from the same L2 source address.

If the Client’s underlay 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 carrier 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 original IP packets/parcels without any encapsulation. In that case, the Client sends packets/parcels over the Direct link according to traffic selectors. If the Direct interface is selected, then the Client’s packets/parcels are transmitted directly to the peer without traversing an ANET/INET. If other interfaces are selected, then the Client’s packets/parcels are transmitted via a different interface, which may result in the inclusion of Proxy/Servers and Gateways in the communications path. Direct interfaces must be tested periodically for reachability, e.g., via NUD.

A.4. AERO Critical Infrastructure Considerations

AERO Gateways can be either Commercial off-the Shelf (COTS) standard IP routers or virtual machines in the cloud. Gateways must be provisioned, supported and managed by the INET administrative authority, and connected to the Gateways of other INETs via inter-domain peerings. Cost for purchasing, configuring and managing Gateways 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 underlay 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 ENETs that provide forwarding services for non-MNP destinations. The Relay connects to the OMNI link and engages in eBGP peering with one or more Gateways 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 ARS/ARR 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, peer carrier packet forwarding to Clients will continue by virtue of the neighbor cache entries that have already been established through address resolution and route optimization. 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 carrier 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 ULAs 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 earlier versions:

* Submit for RFC publication.

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
United States of America
Email: fltemplin@acm.org