Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 15 December 2025

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

13 June 2025

Automatic Extended Route Optimization (AERO)
draft-templin-6man-aero3-45

Abstract

This document specifies an Automatic Extended Route Optimization (AERO) service for IP internetworking over Overlay Multilink Network (OMNI) Interfaces. AERO/OMNI uses IPv6 Neighbor Discovery (IPv6 ND) for control plane 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 path selection, multinet traversal, mobility management, multicast forwarding, multihop operation and route optimization are naturally supported through dynamic neighbor cache updates on a per flow basis. Both Provider-Aggregated (PA) and Provider-Independent (PI) addressing services are supported. AERO is a widely-applicable service especially well-suited for air/land/sea/space mobility applications including aviation, intelligent transportation systems, mobile end user devices, space exploration and many others.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 15 December 2025.

Copyright Notice

Table of Contents

1. Introduction 3
2. Terminology 7
3. Requirements 19
4. Automatic Extended Route Optimization (AERO) 19
4.1. AERO Node Types 19
4.2. The AERO Service over OMNI Links 20
4.2.1. AERO/OMNI Reference Model 20
4.2.2. AERO Addressing 24
4.2.3. AERO Routing System 26
4.2.4. Segment Routing Topologies (SRTs) 28
4.2.5. Segment Routing For OMNI Link Selection 29
4.3. OMNI Interface Characteristics 29
4.4. OMNI Interface Initialization 31
4.4.1. AERO Gateway Behavior 32
4.4.2. AERO Proxy/Server and Relay Behavior 32
4.4.3. AERO Client Behavior 32
4.5. OMNI Interface Neighbor Cache Maintenance 33
4.5.1. AERO/OMNI Control Plane Messages 36
4.5.2. OMNI Neighbor Window Synchronization 39
4.6. OMNI Interface Encapsulation and Fragmentation 40
4.7. OMNI Interface Decapsulation 42
4.8. OMNI Interface Data Origin Authentication 43
4.9. OMNI Interface MTU 43
4.10. OMNI Interface Forwarding Algorithm 44
4.10.1. Client Forwarding Algorithm 46
4.10.2. Proxy/Server and Relay Forwarding Algorithm 47
4.10.3. Gateway Forwarding Algorithm 49
4.11. OMNI Interface Error Handling 51
4.12. AERO Mobility Service Coordination 54
4.12.1. AERO Service Model 54
4.12.2. AERO Client Behavior 56
4.12.3. AERO Proxy/Server Behavior 57
4.13. AERO Address Resolution, Multilink Forwarding and Route Optimization 61
4.13.1. Multilink Address Resolution 63
4.13.2. Multilink Forwarding 69
4.13.3. Mobile Ad-hoc Network (MANET) Forwarding 80

4.13.4. Proxy/Server-to-Proxy/Server Route Optimization . . 81
4.13.5. Gateway-to-Proxy/Server Route Optimization 82
4.13.6. Client-to-Client Route Optimization 82
4.13.7. Intra-(M)ANET/ENET Route Optimization 84
4.14. Neighbor Unreachability Detection (NUD) 84
4.15. Mobility Management and Quality of Service (QoS) 86
4.15.1. Mobility Update Messaging 87
4.15.2. Announcing Link-Layer Information Changes 88
4.15.3. Bringing New Links Into Service 88
4.15.4. Deactivating Existing Links 88
4.15.5. Moving Between Proxy/Servers 89
4.15.6. Accommodating Path Changes 90
4.16. Multicast 91
4.16.1. Source-Specific Multicast (SSM) 92
4.16.2. Any-Source Multicast (ASM) 93
4.16.3. Bi-Directional PIM (BIDIR-PIM) 94
4.17. Operation over Multiple OMNI Links 94
4.18. DNS Considerations 95
4.19. Transition/Coexistence Considerations 95
4.20. Proxy/Server-Gateway Bidirectional Forwarding Detection 96
4.21. Time-Varying MNPs 96
5. Implementation Status 96
6. IANA Considerations 97
7. Security Considerations 97
8. Acknowledgements 100
9. References 102
9.1. Normative References 102
9.2. Informative References 104
Appendix A. Non-Normative Considerations 110
A.1. Implementation Strategies for Route Optimization 110
A.2. Implicit Mobility Management 111
A.3. Direct Underlying Interfaces 111
A.4. AERO Critical Infrastructure Considerations 112
A.5. AERO Server Failure Implications 112
A.6. AERO Client / Server Architecture 113
Appendix B. Change Log 115
Author's Address 116

1. Introduction

Automatic Extended Route Optimization (AERO) fulfills the requirements of Distributed Mobility Management (DMM) [RFC7333] and route optimization [RFC5522] for air/land/sea/space mobility applications including aeronautical networking intelligent transportation systems, home network users, enterprise mobile device users, space exploration and many others. AERO is a secure internetworking and mobility management service that employs the

Overlay Multilink Network Interface (OMNI) [I-D.templin-6man-omni3] with its Non-Broadcast, Multiple Access (NBMA) virtual link model.

The OMNI link is an adaptation layer virtual overlay manifested by IPv6 encapsulation over a network-of-networks concatenation of underlay Internetworks. Nodes on the link can exchange original IP packets or parcels (see: [I-D.templin-6man-parcels2] and [I-D.templin-intarea-parcels2]) 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 Clients as OMNI link end systems via Proxy/Servers and Relays as intermediate systems 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 supports the operation of IPv6 Neighbor Discovery (IPv6 ND) [RFC4861] as the mobility service control message protocol. 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 virtual link overlay service where mobile node Clients use Proxy/Servers acting as proxys and/or designated routers while correspondent nodes on foreign networks may use any Relay on the link for efficient communications. Foreign network correspondent nodes forward original IP packets destined to other AERO nodes via the nearest Relay, which forwards them through the cloud. Mobile node Clients discover shortest paths to OMNI link neighbors through AERO route optimization. Both unicast and multicast communications are supported.

AERO supports both Provider-Aggregated (PA) and Provider-Independent (PI) addressing. Correspondent nodes on foreign networks configure PA addresses from Foreign Network Prefixes (FNPs) advertised by Relays. AERO Clients instead obtain stable PA addresses from Stable Network Prefixes (SNPs) assigned to and managed by First Hop Segment (FHS) Proxy/Servers. Mobile node Clients can also register PI Mobile Network Prefixes (MNPs) with Mobility Anchor Point (MAP) Proxy/Servers to support Internetworking for mobile routers.

AERO Clients receive SNP (PA) addresses and optionally also MNP (PI) prefix delegations through control message exchanges with Proxy/Servers over their local networks. Proxy/Servers provide anchor points for both local network PA operation and global mobility. By binding mobile PI prefixes to stable PA addresses, the AERO service supports the best aspects of PA/PI working together.

AERO Gateways peer with Proxy/Servers in a secured private BGP overlay routing instance to establish a Segment Routing Topology (SRT) virtual 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 a set of Mobility Service Prefixes (MSPs) common to all mobile nodes. Relays provide an optimal route from correspondent nodes on foreign links/networks to mobile or fixed nodes on the local 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 is compatible 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 and/or peers. 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 (MN) that can connect airborne Internet of Things (IoT) sub-networks. 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 Virtual Private Network (VPN) or open Internetwork services enabled according to the appropriate security model. AERO also supports terrestrial vehicular, urban air mobility and mobile pedestrian communication services for intelligent transportation systems [RFC9365]. Other applicable use cases including home and small office networks, enterprise networks and many others represent additional large classes of potential AERO/OMNI users.

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 Client peer-to-peer relaying capability useful when multiple forwarding hops between peers may be necessary to reach a target peer or an infrastructure access point connection to the OMNI link.

6. (Performance) Maximization - the ability to exchange large packets between peers without loss due to a link size restriction, and to adaptively adjust packet sizes to maintain the best performance profile 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 OMNI specification terminology [I-D.templin-6man-omni3] and the IPv6 Neighbor Discovery (IPv6 ND) [RFC4861] node variables, protocol constants and message types (including Router Solicitation (RS), Router Advertisement (RS), Neighbor Solicitation (NS), Neighbor Advertisement (NA), unsolicited NA (uNA) and Redirect) are cited extensively throughout. AERO further introduces new "pseudo" IPv6 ND message types Multilink Initiate (MI), Multilink Respond (MR) and Multilink Control (NC) with formats identical to the standard uNA message but with different Code values. These messages are used to control adaptation layer functions only and are never exposed to the network layer.

Throughout the document, the simple terms "(Proxy/)Client", "Proxy/Server", "Gateway" and "Relay" refer to "AERO/OMNI (Proxy/)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 (OAL).

The terms "All-Routers multicast", "All-Nodes multicast", "Solicited-Node multicast" and "Subnet-Router anycast" are defined in [RFC4291].

The term "IP" refers generically to either Internet Protocol version (IPv4 [RFC0791] or IPv6 [RFC8200]) for specification elements that apply equally to both.

The terms "application layer (L5 and higher)", "transport layer (L4)", "network layer (L3)", "(data) link layer (L2)" and "physical layer (L1)" are used consistently with common Internetworking terminology, with the understanding that reliable delivery protocol users of UDP are considered as transport layer elements. The OMNI specification further introduces an "adaptation layer" positioned below the network layer but above the link layer, which may include physical links and Internet- or higher-layer tunnels. A (network) interface is a node's attachment to a link (via L2), and an OMNI interface is therefore a node's attachment to an OMNI link (via the adaptation layer).

The terms "IP jumbogram", "advanced jumbo (AJ)" and "IP parcel" refer to special packet formats that enable a new link model for the Internet as discussed in [I-D.templin-6man-parcels2] [I-D.templin-intarea-parcels2].

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-omni3].

IPv6 Prefix Delegation (IPv6 PD)
: a networking service for delegating IPv6 prefixes to nodes on the link. AERO nodes apply the IPv6 PD service provided by DHCPv6 [I-D.ietf-dhc-rfc8415bis] in conjunction with OMNI interface IPv6 ND.

GUA, ULA, LLA, MLA
: A Globally-Unique (GUA), Unique-Local (ULA) or Link-Local (LLA) Address per the IPv6 addressing architecture [RFC4193] [RFC4291], or a Multilink-Local Address (MLA) per [I-D.templin-6man-mla]. IPv4 prefixes other than those reserved for special purposes [RFC6890] are also considered as GUA prefixes.

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
: An encapsulation 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 network layer as "L3" and sees all link 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, corporate enterprise network, satellite service provider network, cellular operator network, residential WiFi network, etc.) that connects Clients to the Mobility Service over the OMNI link. Physical and/or data link level security is assumed and sometimes referred to as "protected spectrum" for wireless domains. 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.

Mobile Ad-hoc NETwork (MANET)
: a connected ANET region for which links often have undetermined connectivity properties, lower layer security services cannot always be assumed and multihop forwarding between Clients acting as MANET routers may be necessary.

Internetwork (INET)
: a connected network region with a coherent IP addressing plan that provides transit forwarding services between (M)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 the network and/or higher layers. The global public Internet itself is an example.

End-user Network (ENET)
: a simple or complex "downstream" network tethered to a Client as a single logical unit that travels together. The ENET could be as simple as a single link connecting a single end system, or as complex as a large network with many links, routers, bridges and end user devices. The ENET provides an "upstream" link for arbitrarily many low-, medium- or high-end devices dependent on the Client for their upstream connectivity, i.e., as Internet of Things (IoT) entities. ENETs can also support a recursively-descending chain of additional Clients such that the ENET of an upstream Client is seen as the ANET of a downstream Client.

*NET
a "wildcard" term used when a given specification applies equally to all MANET/ANET/INET cases. From the Client's perspective, *NET interfaces are "upstream" interfaces that connect the Client to the Mobility Service, while ENET interfaces are "downstream" interfaces that the Client uses to connect downstream *NETs which may connect other Clients. Local communications between correspondents within the same *NET can often be conducted based on IPv6 ULAs [RFC4193] or MLAs [I-D.templin-6man-mla].

underlay network/interface
a *NET or 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/ virtual communications media or coordinates with another node that hosts the media.

MANET Interface
a node's underlay interface to a local network with indeterminant neighborhood properties over which multihop relaying may be necessary. All MANET interfaces used by AERO/OMNI are IPv6 interfaces and therefore must configure a Maximum Transmission Unit (MTU) no smaller than the IPv6 minimum MTU (1280 octets) even if lower-layer fragmentation is needed.

OMNI link
the same as defined in [I-D.templin-6man-omni3]. The OMNI link employs IPv6 encapsulation to traverse intermediate systems 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][RFC8754] to navigate between different OMNI links, and/or to cause packets to visit selected waypoints within the same OMNI link.

OMNI link segment
a Proxy/Server and all of its constituent Clients within any attached *NETs is considered as a leaf OMNI link segment, with each leaf interconnected via links and "bridge" nodes in intermediate OMNI link segments. When the *NETs of multiple leaf segments overlap (e.g., due to network mobility), they can combine to form larger *NETs with no changes to Client-to-Proxy/Server relationships. The OMNI link consists of the concatenation of all OMNI link leaf and intermediate segments as a loop-free spanning tree.

OMNI interface
: a node's virtual Ethernet (veth) interface to an OMNI link, and configured over one or more underlay interfaces. If there are multiple OMNI links in an OMNI domain, a separate OMNI interface is configured for each link. The OMNI interface configures a Maximum Transmission Unit (MTU) and an Effective MTU to Receive (EMTU_R) the same as any interface. The OMNI interface assigns an LLA the same as for any IPv6 interface and assigns an MLA for adaptation layer addressing over its underlay networks. The OMNI interface further assigns any unicast or anycast ULA/GUA addresses acquired through address autoconfiguration. Since OMNI interface addresses are managed for uniqueness, OMNI interfaces do not require Duplicate Address Detection (DAD) and therefore set the administrative variable 'DupAddrDetectTransmits' to zero [RFC4862].

OMNI Adaptation Layer (OAL)
: an OMNI interface sublayer service that encapsulates original IP packets 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 Option
: a pseudo IPv6 ND option providing multilink parameters for the OMNI interface. The OMNI option is appended to the end of an IPv6 ND message during OAL encapsulation such that it appears immediately following the final message option.

(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 *NET or ENET 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 regarded as an extension of the IP L2ADDR.)

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

OAL packet
: an original IP packet encapsulated in an OAL IPv6 header with an IPv6 Extended Fragment Header extension that includes an 8-octet (64-bit) OAL Identification value. Each OAL packet is then subject to fragmentation by the source and reassembly by the destination.

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 an IPv6 Extended Fragment Header with an 8-octet (64-bit) OAL Identification value and with Index and More Fragments both set to 0. (Note that control message atomic fragments also omit the Extended Fragment Header over secured spanning tree links.)

(L2) 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 systems. OAL intermediate systems re-encapsulate OAL packets/fragments during forwarding by removing the L2 headers of carrier packets from a previous hop underlay network and replacing them with new L2 headers for the next hop underlay network. Carrier packets may themselves be subject to fragmentation and reassembly in L2 underlay networks at a layer below the OAL. Carrier packets sent over unsecured paths use OMNI protocol L2 encapsulations, while those sent over secured paths use L2 security encapsulations such as IPsec [RFC4301].

OAL source
an OMNI interface acts as an OAL source when it encapsulates original IP packets to form OAL packets, then performs OAL fragmentation and L2 encapsulation to create carrier packets. Every OAL source is also an OAL end system.

OAL destination
an OMNI interface acts as an OAL destination when it decapsulates carrier packets, then performs OAL reassembly/decapsulation to restore the original IP packet. Every OAL destination is also an OAL end system.

OAL intermediate system
an OMNI interface acts as an OAL intermediate system when it decapsulates carrier packets received from a first segment to obtain the OAL packet/fragment, then re-encapsulates in new L2 headers and sends these new carrier packets into the next segment. OAL intermediate systems decrement the Hop Limit in OAL packets/fragments during forwarding, and discard the OAL packet/fragment if the Hop Limit reaches 0. OAL intermediate systems do not decrement the TTL/Hop Limit of the original IP packet, which can only be updated by the network and higher layers. OAL intermediate systems along the path explicitly addressed by the OAL IPv6 Destination (e.g., Proxys, etc.) are regarded as "endpoint" intermediate systems while those not explicitly addressed (e.g., MANET routers, AERO Gateways, etc.) are regarded as "transit" intermediate systems.

Mobility Service Prefix (MSP)
an aggregated IP GUA prefix (e.g., 2001:db8::/32, 2002:192.0.2.0::/40, etc.) assigned to the OMNI link and from which more-specific Mobile and Stable Network Prefixes (MNPs/SNPs) are delegated, where IPv4 MSPs are represented as "6to4 prefixes" per [RFC3056]. 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-omni3]). OMNI links that connect to the global Internet advertise their MSPs to interdomain routing peers.

Mobile Network Prefix (MNP)
a longer IP GUA prefix derived from an MSP (e.g., 2001:db8:1000:2000::/56, 2002:192.0.2.8::/48, etc.) and delegated to an AERO Client.

Stable Network Prefix (SNP)
a global and unique-local IP prefix pair assigned to one or more Proxy/Servers that connect local *NET Client groups to the rest of the OMNI link. Clients request address delegations from the SNP

that can be used to support PA communications. Clients communicate internally within (M)ANETs and INET groups using IPv6 ULAs [RFC4193] assigned in 1x1 correspondence to SNP GUAs [RFC4291] made visible to external peers through IP network address/prefix translation [RFC6145][RFC6146][RFC6147] [RFC6296].

Foreign Network Prefix (FNP)
: a global IP prefix not covered by a MSP and assigned to a link or network outside of the AERO/OMNI domain. Relays advertise any of their associated FNPs into the AERO/OMNI routing system and forward packets between MNP/SNP mobile or fixed nodes on the OMNI link and FNP correspondent nodes on other links.

Subnet Router Anycast (SRA) Address
: An IPv6 address taken from an FNP/MNP/SNP in which the remainder of the address beyond the final bit of the prefix is set to the value "all-zeros". For example, the SRA for 2001:db8:1::/48 is simply 2001:db8:1:: (i.e., with the 80 least significant bits set to 0). For IPv4, the IPv6 SRA corresponding to the IPv4 prefix 192.0.2.0/24 is 2002:192.0.2.0::/40 per [RFC3056].

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

Provider-Aggregated (PA) Address
: a ULA/GUA address pair delegated to a Client from an FHS Proxy/Server SNP is considered Provider-Aggregated (PA) or "Proxy/Server-Aggregated". The Client either assigns the GUA PA address to its own OMNI interface or allows the FHS Proxy/Server to supply the address via Network Prefix Translation for IPv6 (NPTv6) [RFC6296].

Provider-Independent (PI) Address
: a GUA allocated from an MNP delegated to a Client via a MAP Proxy/Server is considered Provider-Independent (PI) or "Proxy/Server-Independent". The Client assigns PI addresses to (downstream) ENET interfaces and can sub-delegate the MNP to downstream ENET nodes.

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

(AERO) (Proxy/)Client
: an AERO node that configures an OMNI interface over one or more underlay interfaces and requests SNP address and/or MNP prefix delegations from AERO Proxy/Servers. The Client assigns a variety

of IPv6 address types to the OMNI interface for use in IPv6 ND exchanges with other AERO nodes and forwards original IP packets to correspondents according to OMNI interface neighbor cache state. The Client coordinates with Proxy/Servers and/or other Clients over upstream ANET/INET interfaces and may also provide Proxy services for other Clients over downstream interfaces.

(AERO) Proxy/Server
an AERO node that provides a proxying service between AERO Clients and external peers on its Client-facing (M)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 (M)ANET interfaces.) The Proxy/Server configures an OMNI interface and maintains BGP peerings with Gateways to provide a local anchor point for its stable and/or mobile Clients. All Proxy/Servers configure a Stable Network Prefix (SNP) and manage 1x1 mappings of internal ULAs and external GUAs according to Network Prefix Translation for IPv6 (NPTv6) [RFC6296].

(AERO) Relay
an AERO Proxy/Server that provides forwarding services between nodes reached via the OMNI link and correspondents on foreign links/networks. AERO Relays maintain BGP peerings with Gateways the same as Proxy/Servers. Relays also run a dynamic routing protocol to discover any Foreign Network Prefix (FNP) routes in service on other links/networks, advertise OMNI link MSP(s) to other links/networks, and redistribute FNPs 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 Internets can also be configured to advertise "default" routes into the OMNI link BGP routing system.)

(AERO) 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 systems 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 FNP/MNP/SNP 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. It is important to note that all Gateways are also Proxy/Servers, but only those Proxy/Servers configured as intermediate nodes in the spanning tree are considered Gateways.

First-Hop Segment (FHS) Client
: a Client that initiates communications with a target peer by sending control messages to establish reverse-path multilink forwarding state in OMNI link intermediate systems on the path to the target. Note that in some arrangements the Client's (FHS) Proxy/Server (and not the Client itself) initiates the exchange.

Last-Hop Segment (LHS) Client
: a Client that responds to a communications request from a source peer's initiation by returning a response message to establish forward-path multilink forwarding state in OMNI link intermediate systems on the path to the source. Note that in some arrangements the Client's (LHS) Proxy/Server (and not the Client itself) returns the response.

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 systems to facilitate RS/RA exchanges between a Client and its MAP 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.

Mobility Anchor Point (MAP) Proxy/Server
: a Proxy/Server selected by a Client that injects the Client's MNP into the BGP routing system and provides both forwarding and mobility services for any *NET underlay interfaces that register the MNP. Clients often select the first FHS Proxy/Server they coordinate with to serve in the MAP role as all FHS Proxy/Servers are equally capable candidates to serve as a MAP. The Client can instead select any available Proxy/Server for the OMNI link as there is no requirement that the MAP must also be one of the Client's FHS Proxy/Servers. This flexible arrangement supports a fully distributed mobility management service.

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 communicating peer nodes. The SRT maintains a

spanning tree established through BGP peerings between Gateways and Proxy/Servers. Each SRT leaf 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 and its connected IoT sub-networks.

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

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 MAP Proxy/Server. Note that a MAP Proxy/Server can assume the ARR role even if it is located on a different SRT segment than the ART. The MAP Proxy/Server assumes the ARR role only when it receives an RS message from the ART with the 'ARR' flag set (see: [I-D.templin-6man-omni3]).

Potential Router List (PRL)
: a geographically and/or topologically referenced list of addresses of all Proxy/Servers within the same OMNI link segment. Each OMNI link segment 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.

flow
: a sequence of packets sent from a particular source to a particular unicast, anycast, or multicast destination that a node desires to label as a flow. The 3-tuple of the Flow Label, Source Address and Destination Address fields enable efficient IPv6 flow classification. The IPv6 Flow Label Specification is observed per [RFC6437] [RFC6438].

AERO Flow Information Base (AFIB)
: A multilink forwarding table on each OAL source, destination and intermediate system that includes AERO Flow Vectors (AFV) with both next hop forwarding instructions and context for reconstructing compressed headers for specific underlay interface pairs used to transport flows from a source to a destination.

AERO Flow Vector (AFV)
: An AFIB entry that includes soft state for each underlay interface pairwise communication flow from source to destination. AFVs are identified by an AFV Index (AFVI) paired with the previous hop L2 address, with the pair established based on an IPv6 ND solicitation and solicited IPv6 ND advertisement response. The AFV also caches underlay interface pairwise Identification sequence number parameters to support carrier packet filtering.

AERO Flow Vector Index (AFVI)
: A 2-octet or 4-octet integer value supplied by a previous hop OAL node when it requests a next hop OAL node to create an AFV. (The AFVI is always processed as a 4-octet value, but compressed headers may omit the 2 most significant octets when they encode the value 0.) The next hop OAL node caches the AFVI and L2 address supplied by the previous hop as header compression/ decompression state for future OAL packets with compressed headers. The previous hop OAL node must ensure that the AFVI values it assigns to the next hop via a specific underlay interface are distinct and reused only after their useful lifetimes expire. The special value 0 means that no AFVI is asserted.

3. Requirements

OMNI interfaces should limit the size of their IPv6 ND control plane messages (plus any original IP packet attachments) to the adaptation layer path MTU which may be as small as the minimum IPv6 link MTU minus encapsulation overhead. If there are sufficient OMNI parameters and/or IP packet attachments that would exceed this size, the OMNI interface forwards the information as multiple smaller IPv6 ND messages and the recipient accepts the union of all information received. This allows the messages to travel without loss due to a size restriction over secured control plane paths that include IPsec tunnels [RFC4301], secured direct point-to-point links and/or unsecured paths that require an authentication signature.

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.

4. Automatic Extended Route Optimization (AERO)

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

4.1. AERO Node Types

AERO (Proxy/)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 receive PA SNP addresses from their Proxy/Servers. AERO Clients that obtain PI MNPs register them with the AERO service, and distribute the MNPs to ENETs (which may connect other Clients). AERO Clients provide Proxy services for Clients on downstream-attached networks.

AERO Gateways, Proxy/Servers and Relays are infrastructure elements in *NET boundary (or standalone INET) deployments and hence have INET addresses that rarely (if ever) change. Together, they provide access to the AERO service OMNI link virtual overlay for connecting AERO Clients. AERO Gateways (together with Proxy/Servers and Relays) provide the secured backbone supporting infrastructure for a Segment Routing Topology (SRT) spanning tree for the OMNI link.

AERO Gateways are Proxy/Servers deployed as OMNI link intermediate systems that 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). The OMNI interface and OAL provide an adaptation layer forwarding service that the network layer perceives as L2 bridging, since the inner IP TTL/Hop Limit is not decremented. Each Gateway peers with Proxy/Servers, Relays and other Gateways in a dynamic routing protocol instance as a Distributed Mobility Management (DMM) service for the list of active MNPs (see: Section 4.2.3). Gateways assign one or more Mobility Service Prefixes (MSPs) to the OMNI link and configure IPsec tunnels with Proxy/Servers, Relays and other Gateways; they further maintain forwarding table entries for each FNP/MNP/SNP 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 acts as either an OMNI link intermediate system or end system according to the service model selected by the Client. Proxy/Servers also peer with Gateways in an adaptation layer dynamic routing protocol instance to advertise its list of associated MNPs (see Section 4.2.3). MAP Proxy/Servers provide prefix delegation 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 *NET boundaries provide a primary forwarding service for (M)ANET Client communications with peers in external INETs. Proxy/Servers in open INETs provide an authentication service for IPv6 ND messages but should be used only as a last resort data plane forwarding service when a Client cannot forward directly to an INET peer. Source Clients securely coordinate with target Clients by sending control messages via a First-Hop Segment (FHS) Proxy/Server which forwards them over the SRT spanning tree to a Last-Hop Segment (LHS) Proxy/Server which finally forwards them to the target.

AERO Relays are Proxy/Servers that provide forwarding services to exchange original IP packets between the OMNI link and fixed or mobile nodes on other links/networks. Relays run a dynamic routing protocol to discover any FNP prefixes in service on foreign 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 FNPs via OMNI link Gateway BGP peerings.

4.2. The AERO Service over OMNI Links

4.2.1. AERO/OMNI Reference Model

Figure 1 presents the basic OMNI link reference model:

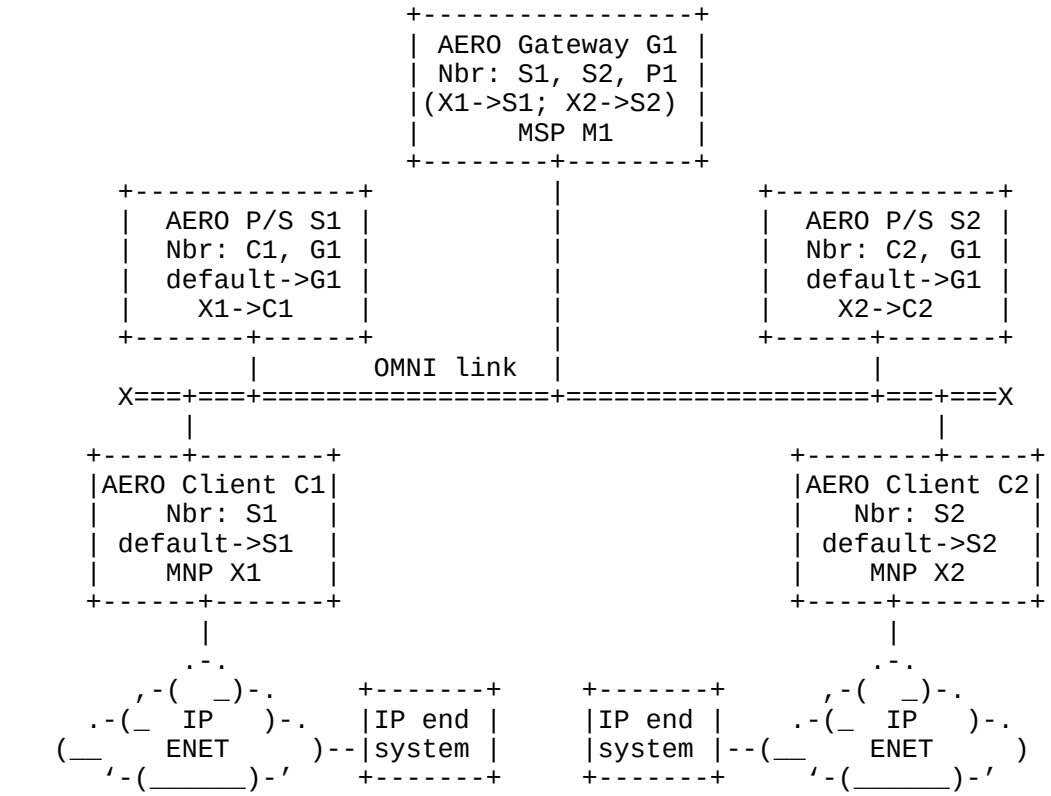

Figure 1: AERO/OMNI Reference Model

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 other Gateways in the SRT (not shown). Together, the set of all 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 FNP 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. (While not shown, AERO Clients can also be recursively nested in an arbitrarily-deep chain of (Proxy/)Clients between a Proxy/Server and the ultimate IP end systems.)

* IP end systems attach to the ENETs served by Clients C1 and C2, respectively. (Although not depicted here, there may be multiple Proxy/Client intermediate systems between Clients C1 and C2 and the ultimate IP end systems.)

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

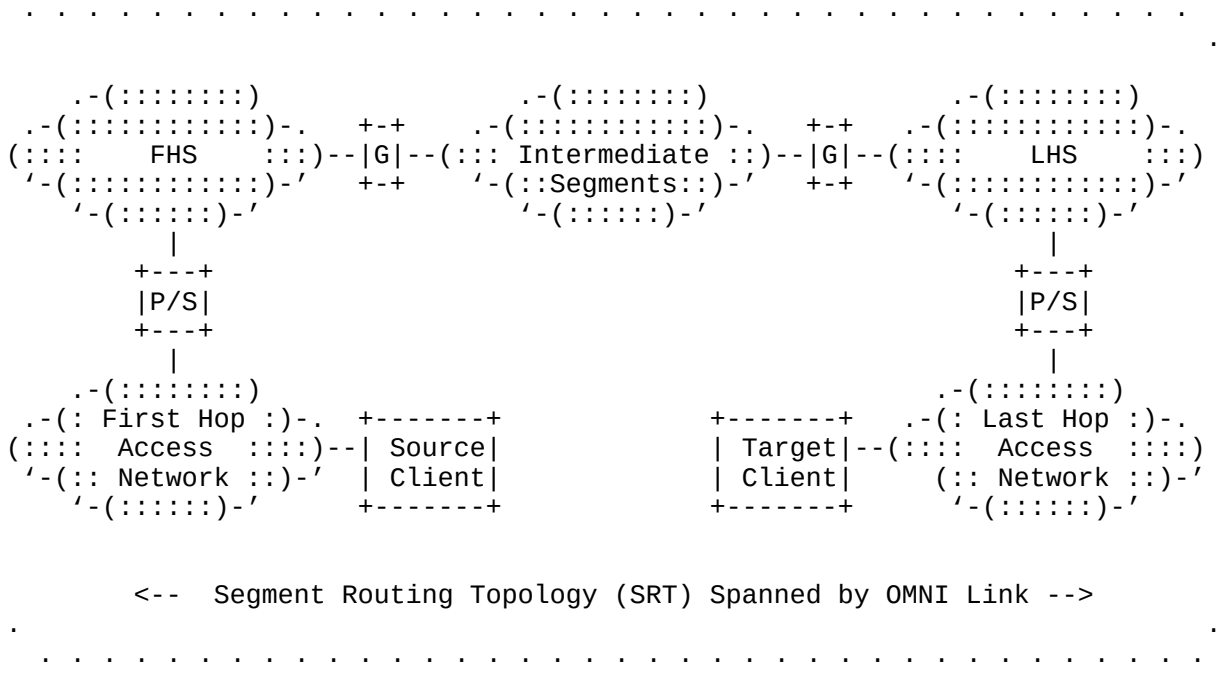

Figure 2: OMNI Link Segment Routing Topology (SRT)

In the Segment Routing Topology, a source Client connects via a first hop access network served by a First Hop Segment (FHS) Proxy/Server. The FHS Proxy/Server then forwards to an FHS Gateway which connects to an arbitrarily complex set of Intermediate Segments. Adjacent intermediate Segments are joined by intermediate Gateways (not shown) that serve as adaptation layer IPv6 routers, with the final segment connected by a Last Hop Segment (LHS) Gateway. The LHS Gateway then forwards to an LHS Proxy/Server which in turn connects to the last hop access network where the target Client resides.

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 examined 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 network of networks concept emerged from the earliest days of Internetworking beginning in the 1960's [KAHN]. The concept has carried forward to the present day where the Internet has become successful beyond measure. The AERO Multinet service concatenates SRT segments through Gateway-to-Gateway peerings as suggested in the "Catenet Model for Internetworking (IEN48)" [CERF]. Catenet suggested a concatenation of independent and diverse Internetwork "segments" to form a much larger network supporting end-to-end services.

The Catenet vision originally articulated in the 1970's faded into obscurity as the Internet evolved in the decades that followed, and the adaptation layer was omitted from the architecture. As a result, the Internet has evolved to become a monolithic public routing and addressing service interconnecting private domains leading to the rise of the middle and a diminished role for end-to-end [RFC3724]. The adaptation layer manifested by AERO and OMNI now promises to restore the best aspects of end-to-end envisioned by Catenet through incremental deployment in the modern Internet.

4.2.2. AERO Addressing

AERO nodes on OMNI links use the Link-Local Address (LLA) prefix fe80::/64 to assign an LLA with randomized EUI-64 interface identifier to the OMNI interface per [RFC4291]. AERO Proxy/Servers use their LLAs as the Source Address for Router Advertisement and Redirect messages as required by [RFC4861], while AERO Clients use their LLAs as Source/Destination Address for NS/NA/uNA messages. The OMNI interface maintains an internal adaptation layer mapping cache that translates the LLAs seen by the network layer into Multilink Local Addresses (MLAs) included in actual IPv6 ND message exchanges with neighbors. (See: [I-D.templin-6man-omni3] for a detailed specification of the OMNI interface adaptation layer mapping cache for LLAs/MLAs and Source/Target Link Layer Address Options (S/TLLAO).)

AERO nodes assign a unique MLA to the OMNI interface per [I-D.templin-6man-mla]. The node assigns an MLA to an OMNI interface the same as suggested for "sites" in the IPv6 scoped addressing architecture [RFC4007], i.e., as a single adaptation layer address assigned to a virtual interface configured over potentially multiple underlying interfaces.

MLAs are considered as adaptation layer addresses in the architecture, but nodes may also use them as the Source and Destination Addresses of original IP packets exchanged between peers in isolated MANETs with intermittent connection to the global Internet. Each original IP packet with MLA addresses is subject to OAL encapsulation with an IPv6 header that also uses MLA addresses.

AERO Clients receive Globally Unique Address (GUA) prefixes during Proxy/Server RS/RA exchanges and configure per underlay interface GUA addresses from the prefixes. AERO Clients also receive a 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]. AERO Proxy/Servers assign ULAs to Clients as *NET internal addresses in 1x1 correspondence with GUAs as *NET external addresses according to NPTv6 [RFC6296].

AERO MSPs, MNPs, FNPs and SNPs are typically based on GUAs, but in some cases may be based on IPv4 private addresses [RFC1918] or IPv6 ULA-C's [RFC4193].

AERO address selection rules are conducted per [RFC6724] as updated by [I-D.ietf-6man-rfc6724-update].

AERO Clients and Proxy/Servers use their MLAs as OAL Source and Destination Addresses within the FHS *NET. FHS Proxy/Servers rewrite OAL MLA Source and Destination Addresses as SNP SRA GUAs before forwarding packets over intervening Gateways on the paths to LHS Proxy/Servers. LHS Proxy/Servers in turn rewrite OAL SNP SRA GUA Source and Destination Addresses as MLAs for forwarding within the LHS *NET.

See [I-D.templin-6man-omni3] for a full discussion of the various unicast, anycast and multicast addresses used by AERO nodes on OMNI links.

4.2.3. AERO Routing System

The AERO routing system comprises a private Border Gateway Protocol (BGP) [RFC4271] service coordinated between Gateways as interior nodes and Proxy/Servers and Relays as leaf nodes of a spanning tree. 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 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 for the SNP prefixes assigned to Proxy/Servers and the set of all FNP/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 flushed from the routing system. 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 foreign 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 the MSP known as the "SRT prefix". For example, a first segment could assign 2001:db8::/48, a second could assign 2001:db8:1::/48, a third could assign 2001:db8:2::/48, etc. Within each segment, each Proxy/Server and Gateway configures an SNP within the segment’s SRT prefix, e.g., the SNPs 2001:db8::/64, 2001:db8:0:1::/64 2001:db8:0:2::/64 all belong to the SRT prefix 2001:db8::/48.

The administrative authorities for each segment must therefore coordinate to assure mutually-exclusive SNP assignments, but internal provisioning of SNPs is an independent local consideration for each administrative authority. For each SRT prefix, the Gateway(s) that connect that segment assign the all-zero's address of the prefix as a Subnet Router Anycast (SRA) address. For example, the SRA address for 2001:db8::/48 is simply 2001:db8::. All Proxy/Servers also assign the SRA address taken from their uniquely-assigned more-specific SNP, e.g., the SRA address for the SNP 2001:db8:0:1::/64 is simply 2001:db8:0:1::.

SRT prefixes (and their SNP sub-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 SRT prefixes either via standard BGP routing or static routes. For example, if three Gateways ('A', 'B' and 'C') from different segments serviced 2001:db8::/48, 2001:db8:1::/48 and 2001:db8:2::/48 respectively, then the forwarding tables in each gateway appear as follows:

A: 2001:db8::/48->local, 2001:db8:1::/48->B, 2001:db8:2::/48->C

B: 2001:db8::/48->A, 2001:db8:1::/48->local, 2001:db8:2::/48->C

C: 2001:db8::/48->A, 2000:db8:1::/48->B, 2001:db8:2::/48->local

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

FNP and MNP routes are instead dynamically advertised in the AERO routing system by Proxy/Servers and Relays that provide anchor points for their corresponding prefixes. For example, if three Proxy/Servers ('D', 'E' and 'F') service the MNPs 2001:db8:1000:1::64/, 2001:db8:1000:2::/64 and 2001:db8:1000:2::/48 then the routing system would include:

D: 2001:db8:1000:1::/64

E: 2001:db8:1000:2::/64

F: 2001:db8:1000:3::/64

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

4.2.4. Segment Routing Topologies (SRTs)

The distinct GUA 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 GUAs, 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 GUA prefix and assigns an IPv6 SRA address used for OMNI interface determination in Safety-Based Multilink (SBM) as discussed in [I-D.templin-6man-omni3]. 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 GUA 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 GUA 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 GUAs of multiple OMNI links within a common routing system may be acceptable.

4.2.5. Segment Routing For OMNI Link Selection

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

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

4.3. OMNI Interface Characteristics

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

* (M)ANET interfaces connect to a protected and secured ANET or an open MANET that connects to an INET via Proxy/Servers. The (M)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 a (M)ANET and may require NAT traversal on the path to the Proxy/Server the same as for the INET case.) MANETs are special cases of ANETs in which adaptation layer multihop forwarding may be necessary, and protected secured underlay links cannot always be assumed.

* 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 end system communications to remote peers. An ENET can be as simple as a small IoT sub-network that travels with a mobile Client to as complex as a large private enterprise network that the Client connects to a larger ANET or INET.

* VPN interfaces use security encapsulations (e.g. IPsec tunnels) over underlay networks to connect Clients, Proxy/Servers and/or Gateways. VPN interfaces provide security services at lower layers of the architecture (L2/L1) the same as for Direct point-to-point interfaces.

* Direct point-to-point interfaces securely connect Clients, Proxy/Servers and/or Gateways over physical or virtual media that does not transit any open Internetwork paths. Examples include a line-of-sight link between a remote pilot and an unmanned aircraft, a fiberoptic link between Gateways, etc.

OMNI interfaces use OAL encapsulation and fragmentation as discussed in Section 4.6. OMNI interfaces use L2 encapsulation (see: Section 4.6) to exchange carrier packets with OMNI link neighbors over INET interfaces and IPsec tunnels as well as over ANET interfaces for which the Client and neighbor 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 (M)ANET interfaces when the Client and neighbor are known to be on the same underlay link.

OMNI interfaces maintain an adaptation layer view of the neighbor cache for tracking per-neighbor state. IP nodes that configure OMNI interfaces use ND messages including Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS), Neighbor Advertisement (NA), unsolicited Neighbor Advertisement (uNA) and Redirect to manage both the network and adaptation layer views of the neighbor cache. The adaptation layer further uses a multilink forwarding message set termed Multilink Initiate (MI), Multilink Respond (MR) and Multilink Control (MC) which use the same ICMPv6 Type value as the standard NA message but with different Code values. OMNI neighbors invoke per-flow OAL Identification window synchronization in their ND message exchanges to enable Source Address verification, header compression and robust fragmentation/reassembly.

OMNI interfaces include OMNI options formatted as specified in [I-D.templin-6man-omni3] in the IP ND messages they forward on behalf of the network layer. The OMNI option includes prefix registration information, Interface Attributes and/or Neighbor Synchronization parameters for coordinating the OMNI interface’s underlay interfaces.

A Client’s OMNI interface may be configured over multiple *NET 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 *NET 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 Neighbor Synchronization 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 *NET 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 a (M)ANET/INET boundary include both (M)ANET and INET underlay interfaces. (M)ANET Clients therefore must discover both the (M)ANET and INET L2ADDR information for their Proxy/Servers.

Gateway and Proxy/Server OMNI interface connections to the SRT are configured over both secured IPsec tunnels for carrying IPv6 ND and BGP protocol control plane messages and open INET paths for carrying unsecured data plane messages. The OMNI interface configures an MLA and acts as an OAL source to encapsulate original IP packets, 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 MLA IPv6 Source and Destination Addresses.

4.4. OMNI Interface Initialization

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

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

4.4.1. AERO Gateway Behavior

AERO Gateways configure an OMNI interface and assign both MLAs and SNPs with corresponding SRA GUAs for 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 an adaptation layer BGP routing protocol session with neighbors over the secured spanning tree (see: Section 4.2.3).

4.4.2. AERO Proxy/Server and Relay Behavior

When a Proxy/Server enables an OMNI interface, it assigns both an LLA and MLA plus one or more SNP ULA/GUA prefix pairs. The Proxy/Server then configures SRA GUAs appropriate for the given OMNI link SRT segment externally and configures SRA ULAs appropriate for the locally attached *NET internally. The Proxy/Server also configures secured underlay interface tunnels and engages in adaptation layer BGP routing protocol sessions over the OMNI interface with one or more neighboring Gateways.

The OMNI interface provides a single interface abstraction to the network layer, but internally serves as an NBMA nexus for exchanging carrier packets with other OMNI nodes 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 (NCEs) 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 foreign networks/links (see: Section 4.2.3). The Relay provisions MNPs and advertises the MSP(s) for the OMNI link over its foreign network interface attachments. The Relay further provides an OMNI link attachment point for FNP-based topologies.

4.4.3. AERO Client Behavior

When a Client enables an OMNI interface, it assigns an LLA and a unique MLA to the OMNI interface. The Client then sends OMNI-encapsulated RS messages to FHS Proxy/Servers which allocate an SNP ULA/GUA address pair and optionally coordinate with a MAP Proxy/Server that delegates one or more MNPs. The MAP/FHS Proxy/Servers then return an RA message to the Client which may pass through one or more NATs in the path.

When the Client sends initial RS messages, it will discover ULAs/GUAs in the corresponding RAs that it receives from FHS Proxy/Servers and can then assign the ULAs/GUAs to the OMNI interface. If the Client is operating outside the context of AERO infrastructure, however, it may continue using MLAs over its underlay or OMNI interfaces for peer-to-peer communications within the local *NET. The Client can then continue indefinitely or at least until it encounters an infrastructure element that can delegate SNP ULA/GUA pairs and/or MNPs.)

A Client can further extend the OMNI link over its (downstream) ENET interfaces where it provides a first-hop router for end systems and other 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 end systems and Clients. This OMNI link extension can be applied recursively over a "chain" of ENET Clients acting as Proxys.

4.5. OMNI Interface Neighbor Cache Maintenance

Each Client and Proxy/Server OMNI interface maintains a network layer conceptual Neighbor and Destination Cache per [RFC1256][RFC4861] the same as for any IP interface. The OMNI interface neighbor cache is maintained through static and/or dynamic neighbor cache entry configurations. The IP layer initiates and terminates IP ND messaging exchanges to manage the network layer view of the neighbor cache.

Each OMNI interface also maintains an internal adaptation layer view of the neighbor cache that includes a Neighbor Cache Entry (NCE) for each of its active OAL neighbors per [RFC4861]. IPv6 ND messages that update the adaptation layer neighbor cache include an OMNI option with zero or more sub-options.

Each OMNI interface NCE is indexed by the IPv6 MLA of a neighbor found in an ND message 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 IPv6 ND message exchanges.

Clients establish NCEs for their associated FHS and MAP Proxy/Servers through the exchange of RS/RA messages. When a Client and Proxy/Server establish NCEs, they set a ReachableTime timer to REACHABLE_TIME seconds. Clients determine the service profiles for their FHS and MAP Proxy/Servers by setting the OMNI Neighbor Synchronization sub-option NUD/ARR/RPT flags in RS messages and also by setting/clearing the FMT-Forward and FMT-Mode flags in the Interface Attributes sub-option. When the NUD/ARR/RPT flags are

clear, Proxy/Servers forward all Address Resolution (NS/NA(AR)) and Neighbor Unreachability Detection (NS/NA(NUD)) 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 IPv6 ND control message overhead especially for Clients connected to low-end data links.

Clients can therefore set the NUD/ARR/RPT flags in RS messages they send to request their desired Proxy/Server service profiles. If the NUD flag is set, the FHS Proxy/Server that forwards the RS message assumes the role of responding to NS(AR/DAD) messages and maintains peer NCEs associated with the NCE for this Client. If the ARR flag is set, the MAP Proxy/Server that processes the RS message assumes the role of responding to NS(AR) and NS(DAD) messages on behalf of this Client NCE. If the RPT flag is set, the MAP Proxy/Server that processes the RS message becomes responsible for maintaining a "Report List" for each Client NCE for the Source Addresses of NS(AR) messages it forwards or responds to on behalf of this Client.

When a Client sets the RPT flag, the MAP Proxy/Server maintains Report List entries based on a ReportTime timer initialized to REACHABLE_TIME seconds upon receipt of an NS(AR) and decremented once per second while no additional NS(AR)s arrive. The MAP Proxy/Server then sends uNA messages to each Report List entry when it receives a Client mobility update indication (e.g., through receipt of an RS with updated Interface Attributes and/or Traffic Selectors). When a Report List entry ReportTime timer expires, the MAP Proxy/Server deletes the entry. When a Client NCE timer expires, the MAP Proxy/Server deletes the NCE along with its associated Report List.

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 for a specific underlay interface. 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 bit settings are discussed in [I-D.templin-6man-omni3].

Both the Client and its MAP 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 request their desired FHS and MAP Proxy/Server service models by setting the NUD/ARR/RPT 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 MAP Proxy/Server for the ART. The MAP 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 multilink forwarding ND message 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 ND messages arrive.

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/MAP Proxy/Server instead.

It is RECOMMENDED that REACHABLE_TIME be set to the default constant value 30 seconds as specified in [RFC4861]. 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 ReachableTime or DepartTime decrement to 0, the NCE is deleted.

AERO nodes also use the value MAX_UNICAST_SOLICIT to limit the number of NS(NUD) 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 solicited IPv6 ND advertisements 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.

4.5.1. AERO/OMNI Control Plane Messages

OMNI interfaces use IPv6 ND messages as the secured control plane messaging service for all adaptation layer neighbor coordination exchanges. OMNI interfaces forward IPv6 ND messages to and from the IP layer the same as for standard IPv6 ND, but during IPv6 ND message encapsulation also append a trailing OMNI pseudo-option [I-D.templin-6man-omni3].

For each IPv6 ND message, the OMNI interface includes a trailing OMNI option following any other ND message options then completely populates all sub-option information. If the OMNI interface includes an Authentication sub-option, it calculates and includes a digital signature per the OMNI specification. OMNI interfaces verify integrity and authentication of each message 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 Flow Information Base (AFIB) as basis for the forwarding algorithm specified in Section 4.10. 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 IPv6 ND message options including the Source/Target Link-Layer Address Option (S/TLLAO) prepared according to the appropriate IPv6 over specific link layer specification (e.g., [RFC2464]). The OMNI option pertains to the adaptation layer to underlay interface address mappings while the S/TLLAO pertains to the network layer to adaptation layer mapping. The adaptation layer appends an OMNI option when it forwards an IPv6 ND message from the network layer to external peers. The adaptation layer translates the S/TLLAO into a local representation of the address and removes the OMNI option when it forwards an IPv6 ND message from external peers to the network layer.

OMNI interface IPv6 ND messages may also include additional OMNI sub-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 must include a Timestamp option. (Note that the OMNI option itself includes Nonce and Timestamp sub-options that will often be used instead of the corresponding IPv6 ND options.)

AERO Clients send RS messages with Source Address set to their own LLA and Destination Address set to link-scoped All-Routers multicast address or the LLA of a Proxy/Server. The OMNI adaptation layer then translates the LLAS to MLAs while using unicast or anycast OAL addresses and appropriate L2 addresses. AERO Proxy/Servers respond by returning RA messages with a unicast LLA Source Address that is translated to the corresponding MLA by the adaptation layer. During RS/RA exchanges, AERO Clients and Proxy/Servers include state synchronization parameters to establish Identification windows and other state.

AERO nodes use NS/NA messages as follows:

* NS/NA(AR) messages are used for address resolution. When an ARS prepares an NS(AR) it sets the IPv6 Source Address to its LLA which the OMNI interface rewrites as its MLA. The ARS also sets the Target Address to the IP Destination Address of the invoking packet and sets the Destination Address to the solicited-node multicast address corresponding to the (unicast) Target Address. After the ARS sends the NS(AR), an ARR with addressing information for the ART returns a unicast NA(AR) that contains current, consistent and authentic Target Address resolution information. The ARR sets the NA(AR) Source Address to its OMNI interface LLA (for translation to its MLA), sets the Destination Address to the Source Address of the NS(AR) and sets the Target Address to the Target Address of the NS(AR). NS/NA(AR) messages must be secured.

* Other NS/NA message exchanges are used to determine target reachability (NS/NA(NUD)). The source sends an NS to the unicast address of the target while optionally including an OMNI Neighbor Synchronization sub-option naming a specific underlay interface pair, and the target returns a responsive NA. 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 and OMNI option checksum. NS/NA messages used to establish or update NCE and/or AFIB state must be secured.

* Unsolicited NA messages (uNAs) are used to update a neighbor's cache when an underlay interface address changes due to a mobility event. Nodes also use uNAs during Route Optimization.

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

AERO introduces three special-purpose IPv6 ND messages for Multilink Forwarding. These messages use the same Type value and message formatting specifications as the standard NA messages but include different Code values. The messages are:

* Multilink Initiate (MI) (Type 136; Code TBD1) - sent as an adaptation layer control message used to initiate state needed to support multilink forwarding. Recipients of MI messages respond by returning a Multilink Respond (MR).

* Multilink Respond (MR) (Type 136; Code TBD2) - an adaptation layer control message multilink forwarding response to an MI message. Recipients of MR messages optionally return a Multilink Control (MC).

* Multilink Control (MC) (Type 136; Code TBD3) - a muti-purpose secured standalone adaptation layer control message used to confirm a previous MI/MR exchange, establish multilink forwarding state, forward error/informational messages or transport passenger packets.

Unlike standard IPv6 ND messages, the pseudo message set (MI/MR/MC) is used internally within the adaptation layer only and the messages are never exposed to the network layer; any MI/MR/MC messages accidentally exposed to the network layer would be silently discarded due to RS message validation rules per [RFC4861] since they include non-zero Code values.

IPv6 ND (pseudo) messages sent on OMNI links that must be examined by transit OAL intermediate systems on the path require a special codepoint for recognition other than the IPv6 Destination Address. The OAL source therefore sets the DSCP field in the IPv6 OAL encapsulation header of such messages to the special value '111111' (see: [I-D.templin-6man-omni3]). The control planes of transit OAL intermediate systems can then intercept and process these messages before forwarding them to the next OAL hop.

IPv6 ND (pseudo) messages that require explicit multihop forwarding guidance include a Segment Routing Header (SRH) [RFC8754] extension to the OAL IPv6 header that includes a Segment List with IPv6 addresses of endpoint intermediate OAL systems on the path. The

Segment List begins with the MLAs of Proxy/Clients on the path from the original source Client to its FHS Proxy/Server, followed by the SNP SRA GUAs of the FHS and LHS Proxy/Servers then finally followed by the MLAs of Proxy/Clients on the path from the LHS Proxy/Server to the final destination Client. (The addressing information is gleaned from Interface Attributes for the FHS and LHS Client interfaces selected to transport the flow as determined through router discovery and address resolution.) The SRH then guides the forwarding algorithm for OMNI link traversal.

IPv6 ND pseudo messages MI and MC set the Target Address to the Source Address of the subject packet, set the Source Address to the MLA of the source peer and set the Destination Address to the Destination Address of the subject packet. IPv6 ND pseudo message MR sets the Target Address to the Destination Address of the subject packet, sets the Source Address to the MLA of the target peer and sets the Destination Address to the Source Address of the subject packet.

IPv6 ND pseudo messages include an SRH extension to the OAL IPv6 header with Segment List addresses included as discussed above. The IPv6 ND pseudo message R/S/O flags are unused; they should be set to 0 on transmission and ignored on reception. As a result, the pseudo messages are distinguished from ordinary uNA messages by the message Code field value alone.

4.5.2. 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 AERO control messages without including Identification values. In environments where spoofing is considered a threat, OMNI interface neighbors instead invoke Identification window synchronization by including OMNI Neighbor Synchronization sub-options in IPv6 ND message exchanges to maintain send/receive window state in their respective neighbor caches as well as in AFIB entries of all OAL intermediate nodes in the forward and reverse paths.

In common arrangements, OAL Identification window synchronization is necessary for Client to Client, Client to Proxy/Server or Proxy/Server to Proxy/Server message exchanges conducted over unsecured Internetwork paths. Conversely, Proxy/Server to Proxy/Server, Proxy/Server to Gateway and Gateway to Gateway message exchanges carried over the secured spanning tree do not require window synchronization.

OAL end system and intermediate nodes verify Identification values of OAL packets that traverse the unsecured spanning tree according to their populated AFIB state. This allows each OAL node to exclude spurious packets injected into the OMNI link from an off-path adversary.

4.6. OMNI Interface Encapsulation and Fragmentation

When the network layer forwards an original IP packet into an OMNI interface, the interface locates a NCE corresponding to the OAL destination. The OMNI interface then invokes the OAL as discussed in [I-D.templin-6man-omni3] which removes the virtual Ethernet header and encapsulates the packet in an IPv6 header to form an OAL packet.

Following encapsulation, the OAL source then fragments the OAL packet while including an identical Identification value for each fragment that must be within the window for the flow over the interface pair selected for the neighbor. The OAL source includes any necessary OAL IPv6 extension headers including an identical SRH with each fragment. The OAL source can instead invoke OAL header compression by replacing the full OAL IPv6 header, SRH and Extended Fragment Header with an OAL Compressed Header (OCH) that includes an AERO Flow Vector Index (AFVI) (see: [I-D.templin-6man-omni3]).

For messages that will traverse unsecured paths, the OAL source finally performs L2 encapsulation on each resulting OAL fragment to form a carrier packet, with Source Address set to its own L2 address (e.g., 192.0.2.100) and Destination Address set to the L2 address of the next hop OAL intermediate system or destination (e.g., 192.0.2.1). The carrier packet encapsulation format in the above example is shown in Figure 3:

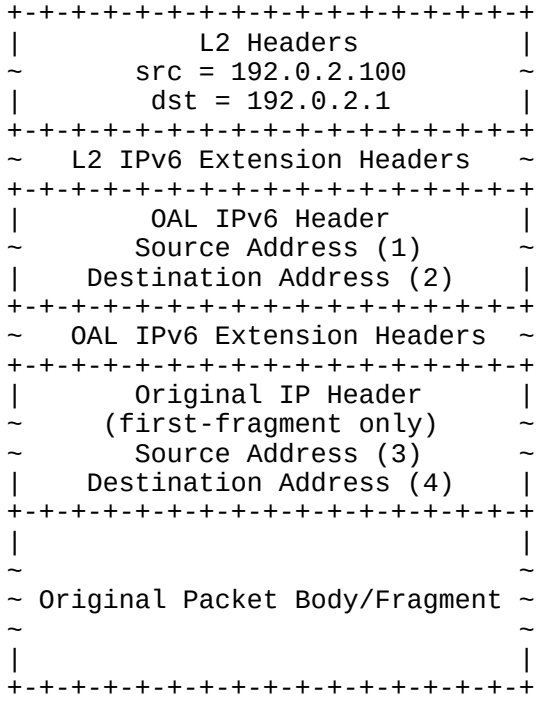

Figure 3: Carrier Packet Format

In this format, the OAL source encapsulates the original IP header and packet body/fragment in an OAL IPv6 header. The OAL source then adds an SRH plus Extended Fragment Header as OAL IPv6 header extensions for each fragment and prepends L2 headers prepared as discussed in [I-D.templin-6man-omni3]. The OAL source sends each such carrier packet into the SRT unsecured spanning tree, where they may be forwarded over multiple OAL intermediate systems until they arrive at the OAL destination. These carrier packets may themselves be subject to L2 fragmentation and reassembly along the concatenated network path segments.

The OMNI link control plane service distributes Client MNP prefix information that may change occasionally due to regional node mobility, as well as more static information for Relay FNPs and per-segment SNPs that rarely change. 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 Flow Information Base (AFIB) that caches AERO Flow 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 entries established through route optimization to greatly improve performance and reduce load on critical infrastructure elements.

For OAL packets/fragments undergoing L2 re-encapsulation at an OAL intermediate system, the OMNI interface performs L2 decapsulation followed by Identification verification and OAL reassembly 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, includes an appropriate Identification, performs OAL fragmentation then for each OAL fragment performs L2 encapsulation to produce a carrier packet appropriate for next segment forwarding.

4.7. 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 performs L2 decapsulation, verifies the Identification, then performs OAL reassembly/decapsulation to obtain the original OAL packet or composite packet (see: [I-D.templin-6man-omni3]). Next, if the enclosed original IP packet(s) are addressed either to itself or to a destination reached via an interface other than the OMNI interface, the OAL node replaces the OAL encapsulation IPv6 header with a virtual Ethernet header and forwards the original IP packet(s) to the network layer.

If the original IP packet(s) are destined to another node reached by the OMNI interface, the OAL node instead changes the OAL Source Address to its own address, changes the OAL Destination Address to the address of the next-hop node over the OMNI interface, decrements the Hop Limit, then performs L2 encapsulation and forwards these new carrier packets into an underlay interface for the next segment.

Further OMNI link decapsulation details are specified in [I-D.templin-6man-omni3]. Further OMNI link forwarding procedures are specified in Section 4.10.

4.8. 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 that originate from within the same secured ANET.

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

* AERO Clients, Relays, Proxy/Servers and Gateways verify carrier packet L2 encapsulation addresses according to [I-D.templin-6man-omni3].

* OAL end systems and intermediate systems forward/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 that do not satisfy the above data origin authentication procedures. Further security considerations are discussed in Section 7.

4.9. OMNI Interface MTU

The OMNI interface observes the link nature of tunnels, including the Maximum Transmission Unit (MTU), Effective MTU to Receive (EMTU_R) and the role of fragmentation and reassembly [I-D.ietf-intarea-tunnels]. The OMNI interface employs the OAL to accommodate multiple underlay links with diverse MTUs. OMNI interface packet sizing considerations are specified in [I-D.templin-6man-omni3], where the OMNI interface MTU can essentially be considered "unlimited".

When the network layer presents an original IP packet to the OMNI interface, the OAL source encapsulates and fragments the packet if necessary. When the network layer presents the OMNI interface with multiple original IP packets addressed to the same IPv6 flow, the OAL source can concatenate them as a single OAL composite packet as discussed in [I-D.templin-6man-omni3] before applying fragmentation. The OAL source then submits each OAL fragment for L2 encapsulation for transmission as a carrier packet 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.

4.10. OMNI Interface Forwarding Algorithm

Original IP packets enter a node's OMNI interface either from the network layer (i.e., from a local application or the IP forwarding system) while carrier packets enter from the link layer (i.e., from an OMNI interface neighbor). All original IP packets and carrier packets entering a node's OMNI interface first undergo data origin authentication as discussed in Section 4.8. Those that satisfy data origin authentication are processed further, while all others are dropped silently.

Original IP packets that enter the OMNI interface from the network layer are forwarded to an OMNI interface neighbor using OAL encapsulation and fragmentation to produce carrier packets for transmission over underlay interfaces. (If forwarding state indicates that the original IP packet should instead be forwarded back to the network layer, the packet is dropped to avoid looping). Carrier packets that enter the OMNI interface from the link layer are either re-encapsulated and re-admitted into the 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 of a router forwards an original IP packet 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 TTL/Hop Limit since its adaptation layer forwarding actions occur below the network layer. The original IP packet'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 system receives a carrier packet, it performs L2 decapsulation to obtain the enclosed OAL packet/fragment. When the intermediate system forwards an OAL packet/fragment not addressed to itself (or one addressed to itself but that also includes an SRH with Segments Left greater than 0), 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 system drops the OAL packet/fragment. This ensures that original IP packet(s) cannot enter an endless loop.

OMNI interfaces may have multiple underlay interfaces and/or NCEs for neighbors with multiple underlay interfaces (see Section 4.3). 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 use only "reachable" interfaces for forwarding purposes.

In addition to standard forwarding based on Interface Attributes and/ or Traffic Selectors, nodes may employ a policy engine that would provide further guidance to the forwarding algorithm. For example the policy engine may suggest a load balancing profile over multiple underlay interface pairs, with portions of a traffic flow spread between multiple paths according to Equal Cost MultiPath or Link Aggregation Groups (LAGs) [RFC6438] (note that Interface Attributes include an underlay interface group identifier). Other policies may suggest the use of paths with the least cost, best performance, etc. This document therefore specifies mechanisms without mandating any particular policies.

All Clients, Proxy/Servers and Gateways serve as OAL intermediate nodes for the purpose of forwarding OAL packets/fragments that include an SRH or OCH with non-zero AFVI over the unsecured spanning tree based on AFIB entries. When an OAL intermediate node forwards an OAL packet/fragment with an L2 Source Address and AFVI that matches an AFV, the node first verifies that the Identification is in sequence. The OAL intermediate node then rewrites the packet's AFVI with a value that will be recognized by the next OAL hop and forwards the packet. (For OAL packets/fragments with uncompressed headers and

with AFVI set to 0, the OAL intermediate node instead forwards based on matching the OAL IPv6 Destination Address with a standard IPv6 forwarding table entry after applying SRH processing if necessary.) The chain of OAL source, intermediate and destination nodes may therefore traverse many (Proxy/)Clients, Proxy/Servers and Gateways on the path.

The following sections discuss the OMNI interface-specific forwarding algorithms for Clients, Proxy/Servers and Gateways. In the following discussion, an original IP packet's Destination Address is said to "match" if it is the same as a cached address, or if it is covered by a cached FNP/SNP/MNP.

4.10.1. Client Forwarding Algorithm

When an original IP packet enters a Client's OMNI interface from the network layer the Client searches for a NCE that matches the corresponding OAL destination. If there is a matching NCE for a neighbor reached via a *NET 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 if necessary, forwards the resulting OAL packet/fragments to an FHS Proxy/Server, then either invokes address resolution and multilink forwarding procedures per Section 4.13 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 to the downstream end system or Client using L2 encapsulation and fragmentation if necessary.

When a carrier packet enters a Client's OMNI interface from the link layer, the Client performs L2 decapsulation if necessary to obtain the OAL packet/fragment then examines the OAL Destination Address (i.e., after locating the correct AFV if the OAL packet header is OCH). If the OAL Destination Address matches one of the Client's addresses and the packet includes an SRH with Segments Left greater than 0, the Client rewrites the OAL Destination Address and forwards the packet to the peer Client indicated by the next hop SRH address. Otherwise, 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 to the network layer. If the OAL Destination Address does not match, the Client drops the original IP packet and MAY return a network layer ICMP Destination Unreachable message subject to rate limiting (see: Section 4.11).

Note: The forwarding table entries established in peer Clients are based on MLAs which also appear as OAL Source and Destination Addresses within (M)ANETs but may be rewritten as GUAs over INETs. The original IP packet Source and Destination Addresses instead use LLAs, ULAs or GUAs. When ULAs are used, the subnet ID in the ULA /64 prefix provides topological relevance for the multihop forwarding region, while the 64-bit Interface Identifier encodes the 1x1 mapping of the MANET-internal ULA to the MANET-external GUA maintained by the Proxy/Server that configures the ULA/GUA SNP.

Note: Clients within MANETs support Client-to-Client multihop forwarding when necessary to reach destinations or FHS Proxy/Servers that may be multiple OAL hops away. In this way, forwarding Clients act as OAL intermediate nodes and forward using OCH compression based on AFV state that is indexed by the AFVIs included in each OAL packet/fragment. ULA-based communications are sufficient for Client-to-Client communications within a MANET, while packets that enter or exit the MANET via a FHS Proxy/Server may be subject to NPTv6 [RFC6296].

4.10.2. Proxy/Server and Relay Forwarding Algorithm

When the network layer admits an original IP packet into a Proxy/Server's OMNI interface, the OAL drops the packet 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 SNP SRA GUA of a neighboring Gateway found in the OMNI interface's network layer neighbor cache. If so, the Proxy/Server performs OAL encapsulation and fragmentation then performs L2 encapsulation and forwards the resulting carrier packet to the Gateway over a secured link (e.g., an IPsec tunnel, Direct link, etc.) to support control plane functions such as the operation of the BGP routing protocol. If the IP Destination Address matches an FNP/SNP/MNP associated with a (foreign) Proxy/Server or Client, the (local) Proxy/Server instead assumes the Relay role and forwards the original IP packet in the same manner as for Client forwarding. 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 forwards the resulting carrier packets while invoking address resolution and multilink forwarding procedures per Section 4.13.

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 MLA, the Proxy/Server drops the carrier packets regardless of their OMNI link point of origin. The Proxy/Server also drops original IP packets received

on underlay interfaces either directly from a *NET Client if the original IP Destination Address corresponds to the same Client's delegated MNP or SNP ULA/GUA. Proxy/Servers also drop carrier packets that contain OAL packets/fragments with foreign OAL Destination Addresses (MLAs) that do not match one of their local *NET Clients. These checks are essential to prevent forwarding inconsistencies from accidentally or intentionally establishing endless loops that could congest nodes and/or *NET links.

Proxy/Servers process carrier packets that contain OAL packets/fragments with OCH headers or with Destination Addresses that match their SNP SRA GUA and also include an SRH. In the first case, the Proxy/Server examines the L2 Source Address and AFVI to locate the corresponding AFV entry in the AFIB. In the second case, the Proxy/Server applies standard SRH processing procedures. The Proxy/Server then forwards them according to the AFV or IPv6 routing state while decrementing the OAL packet/fragment Hop Limit.

For OAL packets/fragments with Destination Addresses that match their MLA or SNP SRA prefix and also include an SRH, the Proxy/Server performs any necessary local processing then rewrites the OAL Destination Address according to the next hop SRH address. For those that do not include an OCH or SRH with additional next hop addresses, the Proxy/Server instead performs L2 decapsulation, verifies the Identification and performs OAL reassembly to obtain the original IP packet. For data packets addressed to its own SNP SRA GUA that arrived via the secured spanning tree, the Proxy/Server delivers the original IP packet to the network layer to support secured BGP routing protocol control messaging. For data packets originating from one of its dependent Clients, the Proxy/Server instead performs OAL encapsulation/fragmentation followed by L2 encapsulation and sends the resulting carrier packets while invoking address resolution and multilink forwarding procedures per Section 4.13. 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 carrier packets that contain OAL packets with OAL Destination Address set to an MLA or SNP ULA/GUA of one of its Client neighbors established through RS/RA exchanges, it accepts the carrier packets only if data origin authentication succeeds. If the NCE state is DEPARTED, the Proxy/Server changes the OAL Destination Address to the SNP SRA GUA of the new Proxy/Server, decrements the OAL Hop Limit, then performs L2 encapsulation and forwards the resulting carrier packets into the spanning tree which will eventually deliver them to the new Proxy/Server. If the neighbor cache state for the Client is REACHABLE and the Proxy/Server is a MAP responsible for serving as the Client's address resolution

responder and/or default router, it verifies the Identification then submits the OAL packet/fragment for reassembly. The Proxy/Server then decapsulates and processes the resulting IPv6 ND message or original IP packet accordingly. Otherwise, the Proxy/Server decrements the OAL Hop Limit, performs L2 encapsulation and forwards the carrier packet to the Client which then performs data origin verification and reassembly. (In the latter case, the Client may receive fragments of the same original IP packet from different Proxy/Servers but this will not interfere with correct reassembly.)

When the Proxy/Server receives carrier packets that contain OAL packets with OAL Destination Address set to a FNP address that does not match the MSP, it accepts the carrier packets only if data origin authentication succeeds and if there is a network layer forwarding table entry for the FNP. The Proxy/Server then performs L2 decapsulation, verifies the Identification, performs OAL reassembly/ decapsulation to obtain the original IP packet, then presents it to the network layer (as a Relay) where it will be delivered according to standard IP forwarding.

When a Proxy/Server receives a carrier packet from the secured spanning tree, it considers the message as authentic without having to verify network or higher layer authentication signatures.

If the Proxy/Server has multiple original IP packets to send to the same neighbor, it can concatenate them as a single OAL composite packet [I-D.templin-6man-omni3].

4.10.3. Gateway Forwarding Algorithm

When the network layer admits an original IP packet 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 SNP SRA GUA 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 link (e.g., an IPsec tunnel, etc.) to support the operation of control plane functions (including the BGP routing protocol) between OAL neighbors.

Gateways forward OAL packets/fragments reassembled from 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 decapsulates to obtain the OAL packet/fragment then searches for an AFIB entry that matches the OAL AFVI or an IPv6 forwarding table entry that matches the OAL Destination Address.

Gateways process carrier packets containing OAL packets/fragments with OAL Destination Addresses that do not match their SNP/SRT SRA GUA 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 SNP/SRT SRA GUA as the OAL Destination Address. If the OAL packet/fragment contains an OCH or a full OAL header with an SRH extension, the Gateway forwards the OAL packet/fragment to the next hop while decrementing the OAL Hop Limit but without reassembling. When the Gateway forwards the OAL packet/fragment, it either rewrites the OCH AFVI with the value it will represent to the next OAL hop or follows standard SRH processing procedures.

If the OAL packet/fragment includes a full OAL header but does not include an AFVI, the Gateway instead examines the OAL packet. The Gateway first determines whether the OAL packet includes an MI/MR/MC message then processes the message according to the multilink forwarding procedures discussed in Section 4.13. If the carrier packets arrived over the secured spanning tree and the enclosed OAL packets/fragments are addressed to its SNP/SRT SRA GUA, the Gateway instead reassembles then discards the OAL header and forwards the original IP packet 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 determines the next hop for a given OAL Destination Address, where the secured/unsecured spanning tree is determined through the selection of the underlay interface to be used for transmission (e.g., an IPsec tunnel or an open INET interface).

As for Proxy/Servers, Gateways must verify that the L2 Source 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).

4.11. OMNI Interface Error Handling

When an AERO node admits an original IP packet into the OMNI interface, it may receive link and/or network layer error indications. The AERO node may also receive OMNI link error indications in OAL-encapsulated MC 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 next OAL hop or by the next OAL hop 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", "Packet Too Big", "Time Exceeded", "Parameter Problem" etc. [RFC0792][RFC4443].

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:

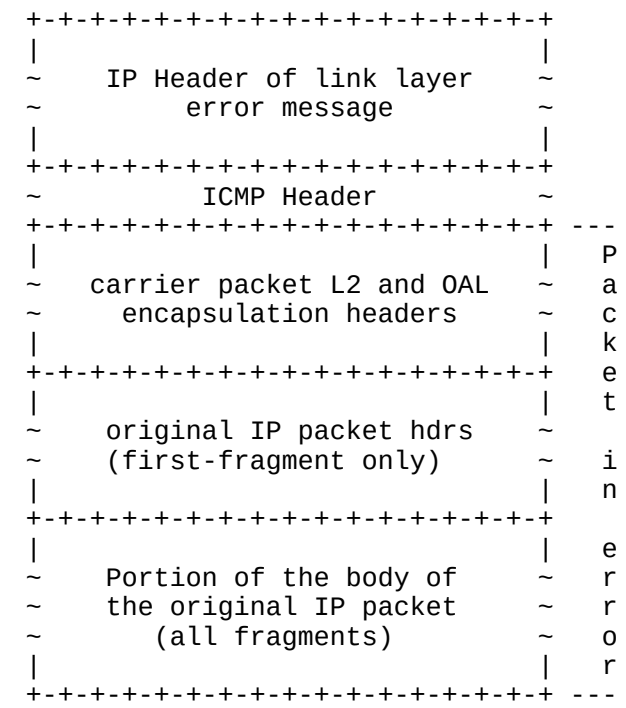

Figure 4: OMNI Interface Link-Layer Error Message Format

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 Packet Too Big messages, there may be a restricting link on the path or the next OAL hop may be experiencing reassembly cache congestion. In both cases, the node should adaptively decrease the size of the OAL fragments it sends to this OAL next hop (note that the PTB messages could indicate either "hard" or "soft" errors).

* 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 adaptively decrease the size of the OAL fragments it sends to this OAL next hop.

* 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 4.15.5.

* 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 re-converge 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.

AERO nodes include ICMPv6 error messages intended for an OAL source as sub-options in the OMNI option of secured MC messages. When the OAL source receives the MC 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.

An AERO/OMNI intermediate system may discover that a transit packet has no matching AFIB state to support forwarding to the next adaptation layer hop. In that case, the intermediate system should return a Destination Unreachable error sub-option in a secured MC

message. The OAL source should process the message as an indication that AFIB multilink forwarding state for a particular flow must be refreshed.

4.12. AERO Mobility Service Coordination

AERO nodes observes the Router Discovery and Prefix Registration specifications found in [I-D.templin-6man-omni3]. AERO nodes further coordinate their autoconfiguration actions with the mobility service as discussed in the following sections.

4.12.1. AERO Service Model

Each AERO Proxy/Server on the OMNI link is configured to respond to Client address delegation requests for Provider Aggregated (PA) addressing. Each Proxy/Server aggregates a unique PA prefix that it does not coordinate with other Proxy/Servers, and ensures that only unique PA addresses are delegated to requesting Clients. Each Proxy/Server runs its own independent DHCPv6 server that shares operational fate with the Proxy/Server itself. If the Proxy/Server goes down, the DHCPv6 service is also disabled and the lease database must be refreshed after the Proxy/Server reboots. Clients assign their PA address delegations to the OMNI interface in association with the corresponding underlay interface for each Proxy/Server.

Each AERO Proxy/Server on the OMNI link is configured to respond to Client prefix delegation/registration requests for Provider Independent (PI) addressing also based on the DHCPv6 service. 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 PI service regardless of the Proxy/Servers they select and provision their PI prefixes for downstream-attached node addressing on ENET interfaces. (Note: an OMNI link can instead delegate non-correlated MNPs to Clients instead of maintaining a common synchronized database. In that case, each Client may receive a different MNP delegation each time it registers with the OMNI domain and may need to renumber its downstream-attached ENETs.)

Clients associate each of their *NET underlay interfaces with FHS Proxy/Servers. 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 MAP Proxy/Server (the Client may instead select a "third-party" MAP 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 MAP Proxy/Server and Client without assuming the MAP role functions themselves.

Each Client typically associates with a single MAP Proxy/Server, while all other Proxy/Servers are candidates for providing the MAP role for other Clients. A Client can select both an FHS and MAP Proxy/Server in a single message by including an SRH in the RS message OAL header when it already knows the FHS and MAP addresses. An FHS Proxy/Server assumes the MAP role when it receives an RS message with a Destination Address that matches its own MLA, or link-scoped All-Routers multicast. An FHS Proxy/Server assumes the proxy role when it receives an RS message with a Destination Address that matches the MLA of another Proxy/Server. (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 SNP SRA GUA of another Proxy/Server to serve as a MAP.)

AERO Clients and Proxy/Servers use IPv6 ND messages to maintain adaptation layer NCEs. 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 FHS/MAP Proxy/Servers include SNP ULA/GUA address delegation (and optionally also MNP prefix delegation) DHCPv6 parameters in RS/RA messages. The IPv6 ND messages are exchanged between the Client and any FHS Proxy/Servers acting as proxys for the MAP Proxy/Server as specified in [I-D.templin-6man-omni3] according to the address/prefix management schedule required by the service. If the Client knows its MNP in advance, it can include the MNP in its DHCPv6 prefix delegation request. If the MAP Proxy/Server accepts the Client's MNP assertion (or if it delegates a new MNP for the Client), it injects the MNP into the routing system and establishes the necessary neighbor cache state.

AERO Clients and their FHS Proxy/Servers on MANETs and open INETs must establish and maintain Identification synchronization windows in their RS/RA exchanges. The window synchronization provides a well-managed Identification value that the Client and Proxy/Server can use for validating IPv6 ND messages with authentication signatures.

All Client and Proxy/Server behaviors for the exchange of RS/RA messages are conducted according to the Router Discovery and Prefix Delegation specifications found in [I-D.templin-6man-omni3]. The following 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.

4.12.2. AERO Client Behavior

AERO Clients discover the addresses of candidate FHS Proxy/Servers as specified in the section on "Router Discovery and Prefix Delegation" in [I-D.templin-6man-omni3]. The Client then performs RS/RA exchanges over each of its underlay interfaces to associate with an FHS Proxy/Server for each interface and a single MAP Proxy/Server if necessary. The Client sends each RS (either directly via Direct interfaces, via an IPsec tunnel for VPN interfaces, via an access router for (M)ANET interfaces or via INET encapsulation for INET interfaces) and waits up to RetransTimer milliseconds for an RA message reply (see Section 4.12.3) while retrying up to MAX_RTR_SOLICITATIONS if necessary. If the Client receives no RAs, or if it receives an RA with Router Lifetime set to 0, the Client SHOULD abandon attempts through the first candidate Proxy/Server and try another Proxy/Server.

After the 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 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 Client receives the MAP Proxy/Server's RA response, it has assurance that both the MAP and FHS Proxy/Servers have been updated with the new information.

If the Client wishes to discontinue use of a MAP Proxy/Server it issues an RS message over any underlay interface with an OMNI Proxy/Server Departure sub-option with an L3ADDR that encodes the (old) MAP Proxy/Server's SNP SRA GUA. When the MAP Proxy/Server processes the message, it releases any MNPs, sets the NCE state for the Client to DEPARTED and returns an RA reply with Router Lifetime set to 0. After a short delay (e.g., 2 seconds), the MAP Proxy/Server withdraws the MNP from the routing system. (Alternatively, when the Client associates with a new FHS/MAP Proxy/Server it can include an OMNI "Proxy/Server Departure" sub-option in RS messages with an L3ADDR that encodes the SNP SRA GUAs of the Old FHS/MAP Proxy/Servers.)

4.12.3. AERO Proxy/Server Behavior

AERO Proxy/Servers act as both IP routers and IPv6 ND proxys, to support address and prefix delegation services for Clients. When a FHS/MAP Proxy/Server receives a prospective Client's secured 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 and performs DHCPv6 address delegation for SNP ULA/GUA pairs while returning the ULA/GUA prefixes per [RFC8028] as specified in [I-D.templin-6man-omni3]. If the RS message also contains DHCPv6 prefix delegation parameters the FHS Proxy/Server processes the prefix delegations locally as a MAP or forwards a proxyed version of the RS to another candidate MAP Proxy/Server.

When the MAP Proxy/Server processes the RS, it determines the correct MNPs for the Client by processing OMNI DHCPv6 sub-option(s). When the MAP Proxy/Server returns the MNPs, it also creates a forwarding table entry for each MNP resulting in BGP updates (see: Section 4.2.3). The MAP Proxy/Server then returns an RA to the Client via the FHS Proxy/server as specified in Section 15 of [I-D.templin-6man-omni3].

After the initial RS/RA exchange, the MAP 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 MAP Proxy/Server sends an RA response and resets ReachableTime. If the MAP Proxy/Server receives an IPv6 ND message with a prefix release indication it sets the Client's NCE to the DEPARTED state and withdraws the MNP routes 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 MAP Proxy/Server marks the interface as DOWN. If ReachableTime expires before any new RS is received on any individual underlay interface, the MAP Proxy/Server sets the NCE state to STALE and sets a 10 second timer. If the MAP Proxy/Server has not received a new RS or uNA message with a prefix release indication before the 10 second timer expires, it deletes the NCE and withdraws the MNP routes from the routing system.

The MAP 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 MAP 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 MAP Proxy/Server may also receive carrier packets via the secured

spanning tree that contain initial data sent while route optimization is in progress. The MAP 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 MAP Proxy/Server forwards any OAL packets/fragments it receives from the secured spanning tree and destined to the Client to the new MAP Proxy/Server, then deletes the entry after DepartTime expires.

Note: Clients SHOULD arrange to notify former MAP Proxy/Servers of their departures, but MAP Proxy/Servers are responsible for expiring NCEs and withdrawing MNP routes even if no departure notification is received (e.g., if the Client leaves the network unexpectedly). MAP 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.

4.12.3.1. Additional Proxy/Server Considerations

AERO Clients register with FHS Proxy/Servers for each underlay interface. Each of the Client’s FHS Proxy/Servers in turn inform the MAP Proxy/Server of the Client’s underlay interface(s) that it services. For Clients on Direct and VPN/IPsec underlay interfaces, the FHS Proxy/Server for each interface is directly connected, for Clients on (M)ANET underlay interfaces the FHS Proxy/Server is located on the (M)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 MAP role or forward a proxyed registration to another Proxy/Server "A" acting as the MAP. 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 AFV associated with the NCE for this Client 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 and Neighbor Synchronization sub-option parameters as well as the Client’s observed L2ADDR (noting that it may differ

from the Interface Attributes L2ADDR if there were NATs on the path). Proxy/Server "B" then examines the RS OAL header SRH extension. If Segments Left is greater than 0 and the next hop SRH address contains the SNP SRA GUA of a different Proxy/Server "A", Proxy/Server "B" prepares a separate proxyed version of the RS message with Source Address set to the MLA of the Client and with Destination Address set to link-scoped All-Routers multicast. Proxy/Server "B" then sets the OAL header Source Address to its own SNP SRA GUA and Destination Address to Proxy/Server A's SNP SRA GUA. Proxy/Server "B" also writes its own L2 address information over the Interface Attributes sub-option L2 information supplied by the Client, then forwards the message into the OMNI link secured spanning tree.

* when MAP Proxy/Server "A" receives the RS, it assumes the MAP role, delegates MNPs for the Client if necessary, and creates/ updates a NCE indexed by the Client's MLA with FHS Proxy/Server "B"'s Interface Attributes as the link layer address information for this FHS ifIndex. MAP Proxy/Server "A" then prepares an RA message with Source Address set to its own MLA, Destination Address set to the Client's MLA, and with OMNI option DHCPv6 sub-options with the prefix delegation results. MAP Proxy/Server "A" then encapsulates the RA in an OAL header with Source Address set to its own SNP SRA GUA, Destination Address set to the SNP SRA GUA of FHS Proxy/Server "B" and with an SRH extension that includes the Client's MLA. MAP Proxy/Serer "A" then finally performs L2 encapsulation and sends the resulting carrier packet into the secured spanning tree.

* when FHS Proxy/Server "B" receives the RA, it locates the Client NCE based on OAL addressing information. If the RA message includes an OMNI "Proxy/Server Departure" sub-option with non-zero old FHS/MAP Proxy/Server SNP GUAs that do not match its own GUA, FHS Proxy/Server "B" first sends a uNA to the old FHS/MAP Proxy/ Servers named in the sub-option. Proxy/Server "B" then re-inserts the cached Neighbor Synchronization sub-option for this Client while updating the window synchronization parameters. Proxy/ Server "B" then resets the RA Source Address to its own MLA and resets the RA Destination Address to the Client's MLA.

* Proxy/Server "B" then re-encapsulates the message with OAL Source Address set to its own MLA and OAL Destination Address set to the Client's MLA. Proxy/Server "B" also includes an appropriate Identification value and authentication signature if necessary, then includes the Client's Interface Attributes sub-option with the cached observed L2ADDR. Proxy/Server "B" sets the P flag in the RA flags field to indicate that the message has passed through a proxy [RFC4389] then returns the RA 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 MAP "A" and the Client. The Client creates/updates NCEs for each such FHS Proxy/Server as well as the MAP 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 SNP SRA GUA as the Source Address and with Destination Address 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 or link layer information - the process of establishing NCEs is specified in Section 4.13.

While the Client is still associated with FHS Proxy/Servers "B", "C", "D", "E", etc., each FHS Proxy/Server can send NS, RS and/or MI/MR/MC messages to update the NCEs of other AERO nodes on behalf of the Client based on changes in Interface Attributes, Traffic Selectors, Neighbor Synchronization parameters, 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 (M)ANET data links.

If the MAP Proxy/Server "A" ceases to send solicited RAs, FHS Proxy/Servers "B", "C", "D", "E", etc. can send unsolicited RAs over to the Client with Destination Address set to (link-local) All-Nodes multicast and with Router Lifetime set to zero to announce the MAP Proxy/Server failure. Although Proxy/Servers "B", "C", "D", "E", etc. 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.

If the Client becomes unreachable over all underlay interfaces it serves, the MAP Proxy/Server sets the NCE state to DEPARTED and retains the entry for DepartTime seconds. While the state is DEPARTED, the MAP Proxy/Server forwards any OAL packets/fragments destined to the Client to a new MAP Proxy/Server if known; otherwise, it discards the OAL packets/fragments. When DepartTime expires, the MAP Proxy/Server deletes the NCE, withdraws any MNP routes and discards any further carrier packets that contain OAL packets/fragments destined to the former Client.

4.12.3.2. 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 MAP 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 NUD control messaging is carried only over well-connected ground domain networks (i.e., and not low-end aeronautical radio links) and can therefore be tuned for rapid response.

FHS Proxy/Servers can perform continuous NS/NA(NUD) exchange with the MAP Proxy/Server, e.g., one exchange per N seconds. The FHS Proxy/Server sends the NUD message via the spanning tree with its own SNP SRA GUA as the OAL Source Address and the SNP SRA GUA of the MAP Proxy/Server as the OAL Destination Address. The MAP Proxy/Server responds with a NUD reply. When the FHS Proxy/Server also sends RS messages to a MAP 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 NUD message if it has already sent an RS within the same period. If the MAP 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 MAP Proxy/Server’s MLA, 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 MAP Proxy/Server will receive the RA messages and select a different Proxy/Server to assume the MAP role.

4.13. AERO Address Resolution, Multilink Forwarding and Route Optimization

AERO nodes invoke address resolution, multilink forwarding and route optimization when they need to forward the initial original IP packets of flows to new neighbors over (M)ANET/INET interfaces as well as for ongoing multilink forwarding coordination with existing neighbors.

Possible Source and Destination Addresses for original IP packets that traverse a local (M)ANET/INET and/or the rest of the OMNI link include addresses taken from an FNP or MNP, or the SNP GUA assigned to a Client. The flow is then identified by the 3-tuple consisting of the IPv6 Source Address, Destination Address and Flow Label.

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). The ARS engages address resolution by sending NS(AR) messages to determine adaptation and link-layer address mappings for the ART network layer address. The ARS discovers this information in any OMNI Interface Attributes sub-options included in NA(AR) messages returned by the ART. Both the ARS and ART can update their Destination Caches based on any peer IPv6 addresses and/or update their routing tables based on any Route Information Options (RIOs) [RFC4191] included in the NS/NA(AR) exchange.

The original source or its current FHS/MAP Proxy/Server serves as the ARS. Either the ART itself or the current LHS/MAP Proxy/Server (or Relay) for the ART serves as the Address Resolution Responder (ARR), i.e., the NA(AR) source.

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

* For Clients on VPN/IPsec and Direct interfaces, the Client’s FHS Proxy/Server is the ARS.

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

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

* For FNP correspondent nodes on foreign links/networks serviced by a Relay, the Relay is the ARS.

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

* For peer Clients within the same (M)ANET/ENET, address resolution and 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 complete, 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 subsequent multilink forwarding and route optimization exchanges to maintain optimal forwarding profiles for each distinct flow.

During address resolution, multilink forwarding and/or route optimization an IPv6 ND message source may attach a small number of original IP packets associated with the message exchange as composite packet extensions per [I-D.templin-6man-omni3]. The authentication signatures and/or lower-layer security features employed at the OAL source and each OAL intermediate system will provide authorization and integrity services for both the IPv6 ND messages and their IP packet attachments. The final OAL intermediate system in the path will then securely forward the IPv6 ND message IP packet attachments to the target.

The source can attach original IP packets to the subject IPv6 ND message, but this may cause the message size to exceed the IPv6 minimum MTU and/or result in sub-optimal forwarding for the IP packet attachments. In that case, the source can instead create small MC "pilot" messages used to transport the original IP packets as attachments over shortest paths determined by routing. The OAL source can attach as many IP packets as will fit without causing the OAL packet to exceed the adaptation layer path MTU using the composite packet construct discussed in [I-D.templin-6man-omni3].

When the target Proxy/Server or Client receives a pilot MC, it removes all passenger attachment packets then delivers the original IP packet(s) to the destination. This service supports assured (but sub-optimal) short-term delivery of protocol data while neighbor coordination is in progress without creating network state.

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

4.13.1. Multilink Address Resolution

The IP layer engages address resolution over OMNI interfaces the same as specified in Section 7 of [RFC4861] including the sending and receiving of NS/NA(AR) messages as well as their implications for neighbor cache entry creation and state management. The OMNI interface therefore exhibits an IP layer behavior that is indistinguishable from an ordinary Ethernet interface while managing adaptation layer state at a layer below IP as discussed below.

When one or more original IP packets for a flow 3-tuple are forwarded over an OMNI interface, the ARS checks the Destination Cache to determine whether there is a NCE that matches the Destination

Address. If there is a NCE in the REACHABLE state, the ARS invokes the OAL and forwards the resulting carrier packets according to the cached state then returns from processing.

Otherwise, if there is no NCE the ARS creates one in the INCOMPLETE state. The ARS then prepares an Address Resolution NS(AR) message to send toward an ART. The resulting NS(AR) message must be sent securely and includes Source, Destination and Target Addresses as discussed in Section 4.5.1. The NS(AR) message also includes Interface Attributes for any of the source Client's underlay interfaces plus RIOs for any of its MNPs.

The ARS then includes an OMNI option with an Authentication sub-option (if necessary), Interface Attributes and/or Traffic Selectors for all of the source Client's underlay interfaces. The ARS then calculates and includes an authentication signature (if necessary) followed by the checksum, then submits the NS(AR) message for OAL encapsulation.

When the ARS is a FHS Proxy/Server, it sets the OAL Source Address to the Client's SNP GUA and sets the OAL Destination Address to the FNP/MNP SRA GUA or SNP GUA corresponding to the ART. The ARS then performs L2 encapsulation and sends the resulting carrier packet 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 its own MLA as the OAL Source Address and the MLA of the interface-specific FHS Proxy/Server as the OAL Destination Address. If the Client is in a MANET or an open INET, it next calculates and includes an authentication signature then includes an OAL IPv6 Extended Fragment Header with Identification set to an in-window value for this FHS Proxy/Server. The ARS Client then performs L2 encapsulation and forwards the carrier packet to the FHS Proxy/Server.

The FHS Proxy/Server then performs L2 decapsulation, verifies the Identification, verifies the NS(AR) checksum/authentication signature and confirms that the Client's claimed MNP RIO(s) and Source Address are correct. The FHS Proxy/Server then changes the OAL Source Address to the Client's SNP GUA and changes the OAL Destination Address to the FNP/MNP SRA GUA or SNP GUA corresponding to the NS(AR) Target Address. The FHS Proxy/Server next removes the IPv6 Extended Fragment Header, performs L2 encapsulation and sends the resulting carrier packet into the secured spanning tree on behalf of the Client.

Note: both the source and target Client/Relay and their MAP Proxy/Servers include current and accurate information for their multilink Interface Attributes profile. The MAP Proxy/Servers can be trusted to provide an authoritative ARR response and/or mobility update message on behalf of the source/target if necessary.

Note: Address Resolution over OMNI interfaces is driven by network layer NS/NA(AR) messaging the same as for any IP interface. The OMNI interface rewrites the S/TLLAO with a local representation of the address upon message reception while caching any updated information in the adaptation layer view of the neighbor cache.

4.13.1.1. Relaying the NS(AR)

When a Gateway receives carrier packets containing the NS(AR), it performs L2 decapsulation and determines the next hop by consulting its standard IPv6 forwarding table for the OAL header Destination Address. The Gateway next decrements the OAL header Hop Limit, 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 intermediate OMNI link segments interconnected by Gateways. The final Gateway will deliver the carrier packets via the secured spanning tree to the LHS/MAP Proxy/Server (or Relay) that services the ART.

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

When the LHS/MAP Proxy/Server (or Relay) of the ART receives the NS(AR) secured carrier packets with the FNP/MNP SRA GUA or SNP GUA of the ART as the OAL Destination Address, it performs L2 decapsulation then either forwards the NS(AR) to the ART or processes it locally if it is acting as the ART's designated ARR. The LHS/MAP Proxy/Server (or Relay) processes the message as follows:

* if the NS(AR) target matches a Client NCE in the DEPARTED state, the (old) MAP Proxy/Server resets the OAL Destination Address to the SNP SRA GUA of the Client's new MAP Proxy/Server. The old MAP Proxy/Server then decrements the OAL header Hop Limit, performs L2 encapsulation and forwards the resulting carrier packet over the secured spanning tree.

* If the NS(AR) target matches a Client NCE in the REACHABLE state, the LHS/MAP Proxy/Server (or Relay) notes whether the NS(AR) arrived from the secured spanning tree. If the message arrived via the secured spanning tree the LHS/MAP Proxy/Server (or Relay) verifies the NS(AR) checksum only; otherwise, it must also verify the message authentication signature.

* If the LHS/MAP Proxy/Server maintains a Report List for the ART, it next records the NS(AR) Source Address in the Report List for this ART. If the MAP Proxy/Server is the ART's designated ARR, it forwards any original IP packet(s) attached to the NS(AR) composite packet to the ART and prepares to return an NA(AR) as discussed below; otherwise, the LHS/MAP Proxy/Server determines the underlay interface for the ART and proceeds as follows:

  - If the LHS/MAP Proxy/Server is also the LHS Proxy/Server on the underlay interface used to convey the NS(AR) to the ART, it includes an OAL IPv6 Extended Fragment Header with an in-window Identification for the ART Client and authentication signature if necessary then recalculates the NS(AR) checksum. The Proxy/Server then changes the OAL Source Address to its own MLA and OAL Destination Address to the MLA of the ART, decrements the OAL Hop Limit, performs L2 encapsulation and forwards the resulting carrier packet over the underlay interface to the ART.

  - If the MAP Proxy/Server is not the LHS Proxy/Server on the underlay interface used to convey the NS(AR) to the ART, it instead changes the OAL Source Address to its own SNP SRA GUA and changes the OAL Destination Address to the SNP SRA GUA of the LHS Proxy/Server for a selected ART interface. The MAP Proxy/Server next decrements the OAL Hop Limit, performs L2 encapsulation and forwards the resulting carrier packet over the secured spanning tree.

  - When the LHS Proxy/Server receives the carrier packets, it performs L2 decapsulation, verifies the NS(AR) checksum, then forwards to the ART while changing the OAL Source and Destination Addresses to MLAs as above. The LHS Proxy/Server also includes an IPv6 Extended Fragment Header and authentication signature if necessary while recalculating the checksum the same as described above.

* If the NS(AR) target matches one of its FNP routes, the MAP/LHS Proxy/Server serves as both a Relay and an ARR, since the Relay forwards original IP packets toward FNP target nodes at the network layer.

* Note: when the target's MAP Proxy/Server acts as the ARR, it detaches any original IP packets attached to the NS(AR) and attaches them to an MC message addressed to the ART. The MAP Proxy/Server then forwards the resulting composite packet into the secured spanning tree or includes an authentication signature if the MAP is also an LHS Proxy/Server of the ART.

If the ARR is a Relay or the ART itself, it first creates or updates a NCE for the NS(AR) MLA Source Address while caching all Interface Attributes and Traffic Selector information in the NCE and caching any IPv6 addresses for the original source found in the Interface Attributes (plus the MLA) in the Destination Cache. The ARR then installs any RIO MNP prefixes in the routing table with next hop set to the LLA corresponding to the neighbor's MLA via the OMNI interface. Next, the ARR prepares a solicited NA(AR) message to return to the ARS with the IPv6 Source Address set to the ART's MLA, with IPv6 Destination Address set to the NS(AR) Source Address, and with Target Address set to the NS(AR) Target Address.

The ARR then includes RIOs for all of the ART's MNPs plus Interface Attributes and Traffic Selector sub-options for all of the ART's underlay interfaces with current information for each interface including their SNP GUA addresses. 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 and IPv6 Extended Fragment Header and an SRH with addressing information for the ARS MAP/FHS Proxy/Server if necessary. The ARR next calculates the NA(AR) message checksum then submits the NA(AR) for encapsulation with OAL Source Address set to its own MLA and Destination Address set to either the MLA that appeared in the NS(AR) OAL source for (M)ANET traversal or the NS(AR) source itself for INET traversal. The ARR then performs L2 encapsulation and forwards the resulting carrier packet.

When the ART's FHS Proxy/Server receives carrier packets sent by an ART acting as an ARR on its own behalf, it performs L2 decapsulation then verifies the OAL Identification and NA(AR) message checksum/ authentication signature. The Proxy/Server then verifies that any RIO information is acceptable, changes the OAL Source Address to the Client's SNP GUA and changes the OAL Destination Address to the next segment address found in the SRH corresponding to the NS(AR) Source Address. The Proxy/Server next decrements the OAL Hop Limit, removes the OAL Extended Fragment Header, performs L2 encapsulation and finally forwards the resulting carrier packet into the secured spanning tree.

4.13.1.3. Relaying the NA(AR)

When a Gateway receives NA(AR) carrier packets, it performs L2 decapsulation 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, performs L2 encapsulation and forwards the resulting carrier packet via the SRT secured spanning tree where it may traverse multiple intermediate OMNI link segments interconnected by other Gateways. The final-hop Gateway will deliver the carrier packets via the secured spanning tree to a Proxy/Server for the ARS.

4.13.1.4. ARS MAP Proxy/Server NA(AR) Processing

If the ARS Client's MAP Proxy/Server maintains a Report List, the carrier packets containing the NA(AR) will arrive at the MAP due to the OAL Destination Address supplied by the ART (see above). This source MAP then performs L2 decapsulation and records the NA(AR) Target Address in the Report List for this source Client. The MAP then leaves the OAL Source Address unchanged, but changes the OAL Destination Address to the SNP GUA corresponding to the ARS. The MAP then decrements the OAL header Hop Limit, performs L2 encapsulation and sends the resulting carrier packet into the secured spanning tree.

4.13.1.5. Processing the NA(AR) at the ARS

When the ARS receives NA(AR) carrier packets, it performs L2 decapsulation then searches for a NCE that matches the NA(AR) Source Address. The ARS then processes the message the same as for standard IPv6 Address Resolution [RFC4861]. In the process, it caches all OMNI option Interface Attributes and Traffic Selectors in the NCE for the NA(AR) MLA Source Address and caches any IPv6 addresses for the ART found in the Interface Attributes (plus the MLA) in the Destination Cache. The ARS then installs any RIO MNP prefixes in the routing table with next hop set to the LLA corresponding to the NA(AR) MLA Source Address via the OMNI interface. All included Interface Attributes sub-options plus RIOs together provide the address mapping information necessary to satisfy address resolution.

When the ARS is a Client, the SRT secured spanning tree will first deliver the solicited NA(AR) message to the Client's FHS Proxy/ Server, which rewrites the OAL header addresses, includes an OAL Extended Fragment Header with an in-window Identification for this Client, 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 or Identification; if the Client is in a MANET or in the

open INET the Proxy/Server must instead include an Identification and authentication signature (while adjusting the OMNI option size, if necessary). The Proxy/Server changes the OAL Source Address to its own MLA and changes the OAL Destination Address to the MLA of the Client when it forwards the NA(AR). The Proxy/Server then decrements the OAL Hop Limit, performs L2 encapsulation and forwards the resulting carrier packet over the underlay interface to the Client.

4.13.1.6. Reliability

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

4.13.2. Multilink Forwarding

Following address resolution, the ARS and ART (i.e., the end system Clients or their respective Proxy/Servers) can assert per-flow multilink forwarding paths through underlay interface pairs serviced by the same Source/Destination Addresses by sending MI/MR/MC messages with OMNI Neighbor Synchronization sub-options. The MI/MR/MC messages establish per-flow multilink forwarding and header compression state in OAL intermediate systems in the path between the ARS and ART. Note that either the ARS or ART can independently initiate multilink forwarding by sending MI messages on behalf of specific flows over underlay interface pairs.

The source Client or FHS Proxy/Server uses the Source Address of the original IP packet as the MI Target Address, uses its own MLA as the MI Source Address and uses the Destination Address of the original IP packet as the MI Destination Address. The source Client or FHS Proxy/Server then includes the Flow Label of the original IP packet set according to [RFC6437][RFC6438] then finally performs OAL encapsulation while including the MLA of the target neighbor in an SRH extension. The flow 3-tuple is then identified by the MI Target Address, Destination Address and Flow Label.

When the target Client or LHS Proxy/Server returns an MR, it sets the Destination Address to the same address that appeared in the MI Source Address, sets the Target Address to the same address that appeared in the MI Destination Address and sets the Source Address to its own MLA. The target Client or LHS Proxy/Server then sets the Flow Label to the same value that appeared in the MI and finally performs OAL encapsulation while including the MLA of the source neighbor in an SRH extension.

When an OAL source asserts a multilink forwarding path through the transmission of an MI message, it includes an IPv6 Minimum Path MTU Hop-by-Hop Option for the (adaptation layer) IPv6 header per [RFC9268]. Each OAL intermediate node along the path then updates the minimum MTU per the specification. When the OAL destination responds with an MR message, it returns an IPv6 Minimum Path MTU Hop-by-Hop (HBH) option based on the one it received in the MI message per [RFC9268]. This allows the OAL source to discover any OAL Fragment Size (OFS) limitations for this OAL destination (see: [I-D.templin-6man-omni3]). For this reason, OAL nodes that connect SRT segments MUST implement [RFC9268].

The multilink forwarding profile provides support for redundant paths that each OAL node can harness to its best advantage. For example, OAL nodes can use traffic selectors to distribute different traffic types over available multilink paths, while other factors such as metrics, cost, provider, etc. can also provide useful decision points. OAL nodes can also employ multilink forwarding for fault tolerance by sending redundant data over multiple paths simultaneously, or for load balancing where the individual packets of a single traffic flow are spread across multiple independent paths. OAL nodes that engage in multilink forwarding therefore must incorporate a policy engine that selects both inbound and outbound multilink paths for a given traffic profile at a given point in time. This specification therefore provides multilink forwarding mechanisms without mandating any specific multilink policy.

All Client, Proxy/Server and Gateway nodes that configure OMNI interfaces and engage in multilink coordination include an additional forwarding table termed the AERO Flow Information Base (AFIB) that supports OAL packet/fragment forwarding based on original IP packet flows over specific OMNI neighbor interface pairs. The AFIB contains per-flow AERO Flow Vectors (AFVs) identified by the L2 address of the previous OAL hop plus a value known as the AFV Index (AFVI). The AFVs cache uncompressed OAL header information to support forwarding of packets with compressed headers as well as previous/next-hop addressing and AFVI information. The AFVs also cache window synchronization state (i.e., the starting sequence number and window size) for each specific flow. Using the window synchronization state, simple Identification-based data origin authentication is enabled at each OAL source, intermediate system and target node.

Client and Proxy/Server OMNI interfaces manage end system AFIB entries in conjunction with their internal neighbor cache, where the NCEs link to (possibly) multiple AFVs with one per flow over a specific FHS/LHS interface ifIndex pair. When OMNI interface peers need to coordinate, they locate a NCE for the peer (established through address resolution) then use the NCE as a nexus that

aggregates potentially many AVFs which cache AFVIs to support multilink forwarding on a per-flow basis. Gateway OMNI interfaces and the OMNI interfaces of Clients or Proxy/Servers acting as OAL intermediate nodes manage transit AFIB entries independently of their internal neighbor caches. These transit AFVs are indexed by the L2 address and AFVI supplied by the previous hop.

OAL source, intermediate system and target nodes create or update AFVs/AFVIs when they process an MI/MR/MC initiation or response message with an OMNI Neighbor Synchronization sub-option with the SYN flag set (see: [I-D.templin-6man-omni3]). The Target Address of the initiating MI (which is also the Destination Address of the responsive MR) is considered to reside in the "First Hop Segment (FHS)", while the Destination Address of the MI (which is also the Target Address of the responsive MR) is considered to reside in the "Last Hop Segment (LHS)".

The FHS and LHS roles are determined on a per-flow and per-interface-pair basis. After address resolution, either peer is equally capable of initiating multilink forwarding on behalf of a specific flow. The peer that sends the initiating MI message with Neighbor Synchronization for a specific pair becomes the FHS peer while the one that returns the responsive MR becomes the LHS peer for that (flow, interface pair) only. It is therefore commonplace that peers may assume the FHS role for some flows while assuming the LHS role for others, i.e., even though each peer maintains only a single NCE.

When an OAL node sends/forwards an initiating MI or responsive MR with a Neighbor Synchronization sub-option with the SYN flag set, it creates or updates an AFV, caches the Identification window information, caches the MI/MR and OAL IPv6 headers, records the previous hop L2 address and AFVI, then generates a new next hop AFVI or updates the lifetime of an already-established AFVI. The next hop AFVI should be selected within the range [1 - (2**16-1)] unless all values within that range are already in active use. Otherwise, the AFVI must be selected within the range [2**16 - (2**32-1)] while the value 0 indicates "AFVI unspecified". When the OAL node forwards future OAL packets/fragments that include the previous hop L2 address and AFVI, it can unambiguously locate the correct AFV and use the cached information to forward to the next OAL hop.

OAL nodes cache AFVs for up to ReachableTime seconds following their initial creation. If the node processes another MI/MR message specific to an AFV, it updates ReachableTime to REACHABLE_TIME seconds, i.e., the same as for NCEs. If ReachableTime expires, the node deletes the AFV.

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

4.13.2.1. FHS Client-Proxy/Server MI Forwarding

When an FHS OAL source has an original IP packet to send toward an LHS OAL target, it first performs address resolution resulting in the creation of a NCE for the SNP GUA of the target then selects a source and target underlay interface pair. The FHS source then uses its cached information for the target interface as LHS information then prepares an MI message with a Neighbor Synchronization sub-option while setting the MI Source, Target and Destination Addresses as specified above.

The FHS source next creates an AFV then generates and assigns an AFVI for the flow over this interface pair; the AFVI must be unique for its communications to this next OAL hop. The FHS source then includes an SRH with segment addressing information for routing purposes. The FHS source finally includes an OMNI Neighbor Synchronization sub-option with window synchronization parameters and prepares the MI message for transmission while also caching the window synchronization parameters in the local AFV.

If the FHS source is the FHS Proxy/Server, it performs OAL encapsulation while setting the OAL Source Address to the Client’s SNP GUA. The FHS Proxy/Server then performs L2 encapsulation and forwards the resulting carrier packet into the secured spanning tree which will deliver it to an FHS Gateway.

If the FHS source is the FHS Client, it instead includes an authentication signature and OAL Extended Fragment Header with an in-window Identification for its FHS Proxy/Server if necessary. If FMT-Forward and FMT-Mode are both set, the Client sets the Neighbor Synchronization LHS ifIndex to the ifIndex of the target; otherwise, it sets the ifIndex to 0 to allow the FHS Proxy/Server to select the target ifIndex. The FHS Client then calculates the IPv6 ND message checksum, performs OAL encapsulation, sets the OAL Source Address to its own MLA and sets the OAL Destination Address to the MLA of the FHS Proxy/Server. The FHS Client finally performs L2 encapsulation and forwards the resulting carrier packet to the FHS Proxy/Server.

If there are multiple OAL hops between the Client and FHS Proxy/Server, the first OAL intermediate node receives the carrier packets containing the MI then also verifies the checksum and authentication signature. The OAL intermediate node then caches the FHS/LHS Client addressing, AFVI and window synchronization information as previous hop information in a new or existing AFV. The OAL intermediate hop then creates a new unique AFVI to forward to the next OAL hop, then

both caches the AFVI and writes it into the IPv6 ND message AVI field, i.e., while over-writing the value supplied by the previous hop. The OAL intermediate node then forwards to the next OAL hop toward the FHS Proxy/Server which then performs the same functions as the previous OAL hop.

When the FHS Proxy/Server receives the carrier packets, it performs L2 decapsulation, verifies the Identification, and verifies the MI checksum and authentication signature. The FHS Proxy/Server then creates an AFV (i.e., the same as the FHS Client had done) while caching the FHS/LHS Client addressing, AFVI and window synchronization information as previous hop information for this AFV. The FHS Proxy/Server next generates a new unique AFVI to forward to the next OAL hop, then both caches the AFVI in the AFV and writes it into the MI AFVI field. The FHS Proxy/Server next calculates the MI checksum then sets the OAL Source Address to the Client's SNP GUA and OAL Destination Address to the SRH next hop segment address. The FHS Proxy/Server finally decrements the OAL Hop Limit, removes the OAL Extended Fragment Header, performs L2 encapsulation and forwards the resulting carrier packet into the secured spanning tree.

4.13.2.2. FHS/intermediate/LHS Gateway MI 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 MI packet over the secured spanning tree, it performs L2 decapsulation then verifies the MI checksum. The FHS Gateway next creates an AFV based on the previous hop Neighbor Synchronization information, i.e., the same as the FHS Proxy/Server had done. The FHS Gateway then generates a locally-unique AFVI for the next hop and both caches the value in the AFV and copies it into the MI AFVI.

The FHS Gateway then examines the SRT prefixes corresponding to both the 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 MI Neighbor Synchronization information in the AFV, and writes a new locally-unique AFVI for the next hop into the AFV and MI AFVI. The FHS Gateway then decrements the OAL Hop Limit, performs L2 encapsulation and forwards the resulting carrier packet 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 update their transit AFVs in the same

fashion while selecting new locally-unique AFVIs for the next hop based on Neighbor Synchronization and SRH information. The LHS Gateway then performs L2 decapsulation, verifies the Identification, verifies the MI checksum then creates an AFV (i.e., the same as all previous hop Gateways had done) while caching the Neighbor Synchronization information from the previous hop and creating a new AFVI for the next hop. The LHS Gateway then decrements the OAL Hop Limit, performs L2 encapsulation and forwards the resulting carrier packet into the secured spanning tree.

4.13.2.3. LHS Proxy/Server-Client MI/MR Processing

When the LHS Proxy/Server receives the carrier packets from the secured spanning tree, it performs L2 decapsulation, verifies the MI checksum then creates an AFV and caches the previous hop Neighbor Synchronization and addressing information.

If the MI Destination Address matches the SNP GUA of the target and the LHS Proxy/Server is configured to respond on the target's behalf, (i.e., if FMT-Forward is set) it next prepares to return a responsive MR. The LHS Proxy/Server next creates or updates an NCE for the MI Source Address (if necessary) with state set to STALE.

The LHS Proxy/Server then creates an MR while copying the Neighbor Synchronization sub-option from the MI and including responsive window synchronization information. The LHS Proxy/Server sets the MR Source Address to its own MLA, sets the MR Destination Address to the MI Target Address and sets the MR Target Address to the MI Destination Address. The LHS Proxy/Server then encapsulates the MR with OAL Source Address set to the MI OAL Destination Address, with OAL Destination Address set to the MI OAL Source Address and with an SRH extension with segment addressing information for the FHS Proxy/Server. If the LHS Proxy/Server will set the MR message SYN flag, it also writes a non-zero value in the MR AFVI field; otherwise, it writes the value 0. The LHS Proxy/Server then calculates the MR checksum, performs L2 encapsulation and forwards the resulting carrier packet into the secured spanning tree.

The LHS Proxy/Server then creates a locally-unique AFVI for the Client and both caches it in the newly-created AFV and writes it into the MR AFVI field. If FMT-Forward is clear and FMT-Mode is set, the LHS Proxy/Server next resets the Neighbor Synchronization FHS ifIndex to 0. The LHS Proxy/Server next includes an authentication signature in the MI if necessary, changes the OAL Source Address to its own MLA and changes the OAL Destination Address to the MLA of the LHS Client. The LHS Proxy/Server then decrements the OAL Hop Limit, includes an OAL Extended Fragment Header with an appropriate Identification value if necessary, performs L2 encapsulation and forwards the resulting carrier packet to the LHS Client.

If there are multiple OAL hops between the LHS Proxy/Server and LHS Client, the first OAL intermediate node receives the carrier packet containing the MI then also verifies the checksum and authentication signature. The OAL intermediate node then caches the FHS/LHS Client addressing, AFVI and window synchronization information as previous hop information in a new or existing AFV. The OAL intermediate hop then creates a new unique AFVI to forward to the next OAL hop, then both caches the AFVI and writes it into the MI, i.e., while over-writing the value supplied by the previous hop. The OAL intermediate node then forwards to the next OAL hop toward the LHS Client which then performs the same functions as the previous OAL hop.

When the LHS Client receives the carrier packet, it performs L2 decapsulation, verifies the Identification, then verifies the MI checksum/authentication signature. The LHS Client then creates a NCE for the MI Source Address (if necessary) in the STALE state and caches the MI Neighbor Synchronization information in a new AFV associated with the NCE corresponding to the MI Source Address. If the LHS Client will request reverse path state establishment, it finally generates and assigns a locally-unique AFVI for a flow to be forwarded to the previous hop, which it caches in the new AFV. Otherwise, the LHS Client sets the reverse path AFVI to 0.

The LHS Client then prepares an MR using the same procedures as for the LHS Proxy/Server above while including responsive window synchronization information, the new AFVI in the MR and with SRH addressing information necessary for OAL routing in the reverse path. The LHS Client includes an authentication signature if necessary, calculates the MR message checksum, then encapsulates the MR with OAL Source Address set to its own MLA and OAL Destination Address set to the MLA of the LHS Proxy/Server and with an SRH extension with addressing information for the LHS/FHS Proxy/Servers. The LHS Client finally includes an OAL Extended Fragment Header with an appropriate Identification if necessary, performs L2 encapsulation and forwards the resulting carrier packet to the LHS Proxy/Server.

If there are multiple OAL hops between the LHS Client and LHS Proxy/Server, the first OAL intermediate node receives the carrier packet containing the MR then also verifies the checksum and authentication signature. If the MR Neighbor Synchronization sub-option SYN flag is set, the OAL intermediate node then caches the LHS/FHS Client addressing, AFVI and window synchronization information as previous hop information in a new or existing AFV. The OAL intermediate hop then creates a new unique AFVI to forward to the next OAL hop, then both caches the AFVI and writes it into the MR, i.e., while over-writing the value supplied by the previous hop. The OAL intermediate node then forwards to the next OAL hop toward the LHS Proxy/Server which then performs the same functions as the previous OAL hop.

4.13.2.4. Reverse Path Forwarding of the MR

When the LHS Proxy/Server receives the carrier packet, it performs L2 decapsulation, verifies the Identification and verifies the MR checksum/authentication signature. The LHS Proxy/Server then changes the OAL Source Address to the LHS Client's SNP GUA, changes the OAL Destination Address to the FHS Client's SNP GUA (i.e., based on SRH information provided by the LHS Client), removes the OAL Extended Fragment Header and forwards the MR over the reverse path toward the initiating FHS node, where it may traverse many intermediate Gateways.

Each Gateway along the reverse path processes the Neighbor Synchronization information in the MR message in the same way that the Gateways in the forward path had processed the MI. If the FHS Proxy/Server forwards the message, it changes the OAL Destination Address to the MLA of the FHS Client and changes the OAL Source Address to its own MLA. The MR will eventually arrive at the initiating FHS node as confirmation that AFV state is established in all end and intermediate systems in the forward path.

Flow state is normally unidirectional from the source to the destination, however the LHS node can elect bidirectional state by setting the SYN flag in the MR Neighbor Synchronization sub-option. In that case, the forward and reverse paths between the FHS initiator and LHS responder may traverse different sets of intermediate nodes but the source and destination would need some way to coordinate the value used in the Flow Label. The initiator should then complete the three-way handshake by returning an MC or a data packet with an in-window Identification value to confirm that the SYN was received.

Under nominal conditions when the source and destination do not coordinate their flow 3-tuples, if AFV state is needed in the reverse path a separate MI/MR exchange between the LHS initiator and FHS responder based on asynchronous packet arrivals may be needed. In that case, the forward and reverse path flows bear no relation to one another and will often include different flow 3-tuple information.

4.13.2.5. OAL End System Exchanges Following Synchronization

Following the initial MI/MR/MC exchange OAL end systems can begin exchanging ordinary carrier packets for synchronized flows that include AFVIs and with Identification values within their respective send windows without requiring security signatures and/or secured spanning tree traversal. OAL end and intermediate systems can also consult their AFIBs when they receive carrier packets that contain OAL packets/fragments with AFVIs to unambiguously locate the correct AFV and can use the AFV state to forward OAL packets/fragments to the next hop. OAL end systems must then perform continuous MI/MR/MC exchanges to update window state, register new flows 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-flow 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 NCE must be determined individually using additional MI/MR/MC messages that include Neighbor Synchronization information.

OAL sources can then begin including OCHs in OAL packets/fragments with an AFVI that OAL intermediate systems can use for shortest-path forwarding based on AFVIs instead of spanning tree OAL IPv6 addresses. Forwarding based on the limited OCH information is supported since all OAL nodes in the path up to (and sometimes including) the OAL destination have already established AFVs.

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 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 MI/MR/MC 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 AFVs established through an MI/MR/MC exchange with a remote or local peer. 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 to obtain the OAL packet then decapsulates and delivers the original IP packet to the network layer.

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 4.13.6).

When the FHS Client or FHS Proxy/Server sends an MI for the purpose of establishing multilink forwarding state, it should wait up to RETRANS_TIMER seconds to receive a responsive MR. The FHS node can then retransmit the MI up to MAX_UNICAST_SOLICIT times before giving up.

4.13.2.6. Dynamic Multilink Flow State Management

Following address resolution, either the ARS or ART acting as an OAL source may immediately begin forwarding original IP packets for a flow as composite packet attachments to "pilot" MC messages without waiting for an initial MI/MR exchange. The messages will include an authentication signature if necessary that covers the entire composite packet including the MC pilot and all original IP packet attachments.

Each MC message OMNI option contains a Neighbor Synchronization sub-option with the SYN flag set to cause OAL intermediate systems in the forward path to create or update AFIB state. The MC sets its Source/Destination/Target Addresses and Neighbor Synchronization flags the same as described for MI messages above. All OAL intermediate systems in the forward path will process the MC message the same as specified for MI messages above. If the MC includes an IPv6 Minimum Path MTU HBH option, the OAL destination returns an MC message with addresses the same as described for MR messages above and with a responsive IPv6 Minimum Path MTU HBH option. Otherwise, the OAL destination silently discards the MC without returning an acknowledgement regardless of the Neighbor Synchronization flags.

The OAL source should continue to forward additional original IP packets for the flow that arrive during a brief convergence interval as secured MC composite packets, e.g., up to N composite packets within a 1 second window. The OAL source can then begin forwarding new IP packet arrivals for the flow via the unsecured spanning tree while applying OAL encapsulation, fragmentation and header compression. The OAL source can continue forwarding under the optimistic expectation that AFIB state is securely established in the forward path. As AFIB state becomes stale, the OAL source can resume forwarding secured MC composite packets during a new convergence interval to refresh/renew state as above before resuming via the unsecured spanning tree.

The OAL source can also forward OAL-fragmented packets in this manner by including the fragment body preceded by an OMNI Fragment Header (OMNI-FH) immediately following the OAL-encapsulated MC message. Each fragment MUST be no larger than the minimum OAL Fragment Size (OFS) of 1024 octets to ensure they will transit the secured spanning tree without encountering a size restriction. The fragment size is determined by the MC message payload length minus the initial payload length up to the beginning of the OMNI-FH, i.e., the fragment must appear as the final trailing component of the composite packet.

Fragmentation proceeds according to standard IPv6 fragmentation as specified in [RFC8200] using a monotonically incrementing Identification value. The OAL destination will reassemble these fragments after authenticating the message, then remove the OAL header and forward the resulting original IP packet to the final destination. Note that these same fragmentation and reassembly procedures can be applied during address resolution convergence - see: Section 4.13.1. Further information on OMNI fragmentation and reassembly can be found in [I-D.templin-6man-omni3].

4.13.3. Mobile Ad-hoc Network (MANET) Forwarding

Clients with OMNI interfaces configured over underlay interfaces with indeterminant neighborhood properties may be connected to a Mobile Ad-hoc NETwork (MANET). Each MANET may be either completely outside of the range of any OMNI link Proxy/Servers or may require multihop traversal between Clients acting as MANET routers to reach Proxy/Servers that connect to the rest of the OMNI link. The former class of MANETs must operate in isolation solely based on the unique IPv6 MLAs they configure locally. The latter class allows MANET routers to extend infrastructure-based addressing information including MNPs over multiple OMNI link hops as discussed in the OMNI specification.

MANET Clients configure their OMNI interfaces over one or more MANET interfaces where multihop forwarding may be necessary. Routing protocols suitable for use over MANET interfaces include OSPFv3 [RFC5340] with MANET Designated Router (OSPF-MDR) extensions [RFC5614], OLSRv2 [RFC7181], AODVv2 [I-D.perkins-manet-aodvv2] and others. Other services specific to MANET link-local and/or site-local operations (including SMF [RFC6621], DLEP [RFC8175] and others) are also considered in-scope. These services strive for optimal use of available radio bandwidth and power consumption in their control message transmissions, but efficient data plane operation is also essential.

Clients must therefore reduce overhead through minimal encapsulation and effective header compression whenever possible. For this reason, when the MANET routing protocol discovers a new route the Client configures a lesser-preferred forwarding table entry over the corresponding MANET interface and a more-preferred forwarding table entry over the OMNI interface. This will cause the network layer to direct outbound packets to the OMNI interface, which can apply header compression and underlay MANET interface selection.

Multilink Clients that connect a MANET to the rest of the OMNI link act as regular Clients for exchanges with external INETs, but act as Proxy/Servers over their MANET interfaces. Each such Client therefore has at least two underlay interfaces, including both INET and MANET interfaces. The Client therefore services the MANET as if it were a Proxy/server but presents itself as a Client to external facing INETs. This class of Clients are also known as "Proxy/Clients".

The process for a multihop Client to establish multilink forwarding and header compression AFV state in the MANET is conducted in the same fashion as described above and using the same MI/MR/MC or unidirectional MC message exchanges. Each intermediate MANET node in the path creates or updates AFV state in the same fashion as for

intermediate Gateways in the secured spanning tree except that the MI/MR/MC messages require authentication signatures (unless neighboring MANET nodes configure IPsec tunnels) and an Identification that is within the window for its serving Proxy/Server if the destination is outside of the local MANET. The MI/MR/MC messages extend from the initiating FHS MANET Client, then across any intermediate FHS MANET hops to the FHS Proxy/Client, then to the FHS Proxy/Server, then across the secured SRT spanning tree to the LHS Proxy/Server, then finally across any intermediate LHS MANET hops to the responding LHS Client. In all other ways, the MI/MR/MC or unidirectional MC exchanges are the same as discussed in Section 4.13.2.

Following the MI/MR/MC exchanges, each MANET router in the forward (and optionally also reverse) path in both the FHS and LHS MANETs will have established AFVs containing multilink forwarding and header compression state for the flow. The AFVs determine AFVI-based forwarding based on the OCH header contents, and each MANET router only forwards packets with in-window Identification values for the flow. MANET routers maintain AFVs for up to ReachableTime seconds unless they are refreshed by a new MI/MR/MC message. New window synchronization exchanges must also be performed periodically to avoid window exhaustion and/or spoofing based on predictable Identifications.

Note: while the MANET routing protocol runs directly over the node’s MANET interfaces to discover routing information, the node configures lesser-preferred forwarding table entries over the MANET interface and corresponding more-preferred forwarding table entries over the OMNI interface. This causes the network layer to forward outbound packets via the OMNI interface which applies encapsulation, fragmentation and/or header compression as necessary before forwarding over the underlying MANET interface. The OMNI protocol designator in the UDP port, IP protocol or Ethernet EtherType field will then cause the packets to visit the OMNI interface of each successive next-hop MANET node.

4.13.4. Proxy/Server-to-Proxy/Server Route Optimization

When the FHS and LHS Proxy/Servers are both connected to an IPv6 underlay for the same SRT segment, they can forward MI/MR/MC exchanges directly over the underlay without engaging SRT spanning tree hops. This is made possible when the FHS and LHS Proxy/Servers inject their SNPs into the underlay network routing system as well as the SRT overlay routing system.

If the underlay is not secured, the FHS and LHS Proxy/Servers must include an authentication signature with their MI/MR/MC messages, which could either be the original authentication signature included by their respective Clients or a new signature included by the Proxy/Server itself. If the Proxy/Server that processes the MI/MR/MC message determines that the message is authentic, it creates or updates an AFV entry according to the multilink forwarding parameters. This establishes both AFVI and Identification window state to be used for future data traffic forwarding.

4.13.5. Gateway-to-Proxy/Server Route Optimization

When the LHS gateway and FHS Proxy/Server are both connected to an IPv6 underlay for the same SRT segment, they can forward MI/MR/MC exchanges directly over the underlay without engaging additional SRT spanning tree hops. In this arrangement, the LHS Gateway acts the same as the FHS Proxy/Server as discussed in Section 4.13.4 and observes the requirement for including authentication signatures.

4.13.6. Client-to-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 any shared segment Gateways) 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 initial MI/MR/MC exchanges 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 when FMT-Forward and FMT-Mode are both set). The MI/MR/MC messages then include an Interface Attributes sub-option (i.e., in addition to a Neighbor Synchronization sub-option) with the mapped L2ADDR information 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 the local chain of NATs for reception of future carrier packets from that L2 address (see: [RFC4380] and [I-D.templin-6man-omni3]). The source Client then prepares an MI message with its own MNP SRA GUA or SNP GUA as the Source Address,

with the MNP SRA GUA or SNP GUA of the target as the Destination Address and with an OMNI option with an Interface Attributes sub-option. The source Client then encapsulates the MI in an OAL header with its own MLA as the Source Address, with the MLA of the Proxy/Server as the Destination Address and with an in-window Identification for the target. The source Client then performs L2 encapsulation and 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. The target Client then prepares an MR message with its own MNP SRA GUA or SNP GUA as the Source Address, with the MNP SRA GUA or SNP GUA of the source Client as the Destination Address and with an OMNI option with an Interface Attributes sub-option. The target Client then encapsulates the MR in an OAL header with its own MLA as the Source Address, with the MLA of the source Client as the Destination Address and with an in-window Identification for the source Client. The target Client then performs L2 encapsulation then forwards the resulting carrier packet directly to the source Client.

Following the initial MI/MR exchange, both Clients mark their respective (source, target) underlay interface pairs as "trusted" for no more than ReachableTime seconds. The Clients can then begin exchanging ordinary data packets as OCH encapsulated carrier packets. While the Clients continue to exchange packets via the direct path avoiding all Proxy/Servers and Gateways, they should perform additional MI/MR exchanges via their local Proxy/Servers to refresh NCE state as well as send additional bubbles to the peer’s L2ADDR if necessary to refresh NAT state.

Note: 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: each communicating pair of Clients may need to maintain NAT state for peer to peer communications via multiple underlay interface pairs and/or multiple flows. It is therefore important that L2ADDR information is maintained with the correct peer interface and that the NCE may cache information for multiple peer interfaces.

Note: the source and target Client exchange L2ADDR information during the secured MI/MR/MC multilink route optimization exchange. This allows for subsequent MI/MR/MC 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.

4.13.7. Intra-(M)ANET/ENET Route Optimization

When a Client forwards an OAL packet (or an original IP packet) from 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 any RIOs with MNP information corresponding to the target. These Redirect messages update the Destination and Neighbor Caches the same as specified in [RFC4861].

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

All other route optimization functions are conducted per the MI/MR/MC messaging discussed in the previous sections.

4.14. Neighbor Unreachability Detection (NUD)

AERO nodes perform Neighbor Unreachability Detection (NUD) per [RFC4861] either reactively in response to persistent link layer errors (see Section 4.11) 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 carrier packets should be ignored.

AERO nodes can perform NS/NA(NUD) exchanges over the OMNI link secured spanning tree (i.e. the same as described above) to test reachability without risk of DoS attacks from nodes pretending to be a neighbor. These NS/NA(NUD) messages use the MLAs of the parties involved in the NUD test as Source and Destination Addresses. When only reachability information is required without updating any other NCE state, AERO nodes can instead perform NS/NA(NUD) exchanges directly between neighbors without employing the secured spanning tree as long as they include in-window Identifications and 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 system 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(NUD) messages over the SRT secured or unsecured spanning tree, or through NS(NUD) messages sent directly to an underlay interface of the target itself. While testing a target underlay interface for a given flow, the source can optionally continue to forward OAL packets/fragments via alternate interfaces or maintain a small queue of carrier packets until target reachability is confirmed.

NS(NUD) messages are encapsulated, fragmented and transmitted as carrier packets the same as for ordinary original IP data packets. The source encapsulates the NS(NUD) message the same as described in Section 4.13.2 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, performs L2 encapsulation then forwards the resulting carrier packet into the unsecured spanning tree directly to the target if it is in the local segment.

When the target receives the NS(NUD) carrier packets, it performs L2 decapsulation, verifies that it has a NCE for this source and that the Identification is in-window then performs OAL reassembly. The target next verifies the NS(NUD) checksum/authentication signature, then searches for Interface Attributes in its NCE for the source that match the NS for the NA(NUD) reply. The target then prepares the NA(NUD) with the Source and Destination Addresses reversed, encapsulates and sets the OAL Source and Destination Addresses, includes an Interface Attributes sub-option in the NA(NUD) to identify the ifIndex of the underlay interface the NS(NUD) arrived on and sets the Target Address to the same value included in the NS(NUD). The target next sets the R flag to 1, the S flag to 1 and the O flag to 1, then includes an in-window Identification for the source. The node then performs L2 encapsulation and forwards the resulting carrier packet into the unsecured spanning tree directly to the source if it is in the local segment.

When the source receives the NA(NUD), 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.

4.15. 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 MAP Proxy/Servers via RS/RA exchanges to maintain the DMM profile, and the AERO routing system tracks all current Client/Proxy/Server peering relationships.

MAP 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 MAP 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 MAP 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 MI/MR/MC exchange to re-seed new AFVs in the path.

Mobility management considerations are specified in the following sections.

4.15.1. Mobility Update Messaging

Mobile Clients (and/or their MAP 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 MNP SRA GUA or SNP ULA/GUA, sets the Destination and Target Address to the neighbor's SNP ULA/GUA or one of the mobile Client's MNP SRA GUAs. 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 next sets the uNA R flag to 1, S flag to 0 and O flag to 1, then encapsulates the message in an OAL header. Following OAL and L2 encapsulation, the carrier packet containing 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 send an MI(MM) message to request an MR(MM) response (see: Section 4.5.1).

When the FHS/LHS Proxy/Server receives a secured uNA message prepared as above, if the uNA Destination Address was its own SNP SRA 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 Address was the SNP GUA of the FHS/LHS Client, the Proxy/Server instead changes the OAL Source Address to its own SNP SRA ULA/GUA, includes an authentication signature if necessary, and includes an in-window Identification for this Client.

4.15.2. 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 MAP 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 4.15.5); otherwise, any stale state in the former FHS Proxy/Server will simply expire after ReachableTime expires with no effect on the MAP 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.

4.15.3. 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 MAP 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 interface parameters 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.

4.15.4. Deactivating Existing Links

When a Client needs to deactivate an existing underlay interface, it sends an uNA message toward the MAP Proxy/Server via an FHS Proxy/Server with an OMNI option with appropriate Interface Attributes values for the deactivated link.

If the Client needs to send uNA messages over an underlay interface other than the one being deactivated, it MUST include Interface Attributes 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 NCEs. 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. The manner in which the Client dynamically manages its local ifIndex to interface mappings is a local decision, but should not be done in a manner that could cause state inconsistencies in the network.

4.15.5. Moving Between Proxy/Servers

The Client performs the procedures specified in Section 4.12.2 when it first associates with a new MAP Proxy/Server or renews its association with an existing MAP Proxy/Server.

When a Client associates with a new MAP Proxy/Server, it sends RS messages to register its underlay interfaces with the new MAP while including the old MAP's GUA in the "Old MAP Proxy/Server GUA" field of a Proxy/Server Departure OMNI sub-option. When the new MAP Proxy/Server returns the RA message via the FHS Proxy/Server (acting as a proxy), the FHS Proxy/Server sends an uNA to the old MAP Proxy/Server (i.e., if the GUA is non-zero and different from its own). The uNA has the MNP SRA GUA of the Client as the Target Address and the SNP SRA GUA of the old MAP as the Destination Address and with an OMNI Proxy/Server Departure sub-option as above. The FHS Proxy/Server encapsulates the uNA in an OAL header with the SNP SRA GUA of the new MAP as the Source Address and the SNP SRA GUA of the old MAP as the Destination Address, then performs L2 encapsulation and forwards the resulting carrier packet via the secured spanning tree.

When the old MAP 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 MAP Proxy/Server GUA. After a short delay (e.g., 2 seconds) the old MAP Proxy/Server withdraws the Client's MNP(s) from the routing system. While in the DEPARTED state, the old MAP Proxy/Server forwards any carrier packets received via the secured spanning tree destined to the Client's MNP GUAs or SNP GUA to the new MAP Proxy/Server's SNP GUA. When DepartTime expires, the old MAP 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 for the "Old FHS Proxy/Server

L3ADDR", and the new FHS Proxy/Server will issue a uNA using the same procedures as outlined for the MAP above while using its own SNP SRA GUA 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 MAP 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 MAP Proxy/Server include a MAP 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.

4.15.6. Accommodating Path Changes

After AFV state has been established for a flow, all OAL intermediate systems in the forward path will have AFVs with header compression state and (AFVI, L2ADDR) information for the next hop. However, paths can fluctuate due to factors such as node mobility, routing changes, network membership, etc. If an OAL intermediate system forwarding OAL packets with OCH headers detects that the next hop in the path has changed, it immediately reverts to sending the packets with header compression disabled by including full OAL IPv6 and Extended Fragment Headers (plus full original IP headers) in future packets.

If the OAL intermediate system receives an OCH1 packet with the Q bit set and M bit clear during a path change event, it first decompresses the original IP headers of each payload packet in the (packed) OAL packet while retaining the packets as attachments to the (full) OAL header. The OAL intermediate system then processes the OAL packet further.

If an OAL packet is larger than the minimum OFS, the OAL intermediate system applies OAL fragmentation to produce (sub-)fragments no larger than the minimum OFS. If the original OAL packet/fragment had a fragment ordinal value N, the OAL intermediate node writes the same value N into each of the (sub-)fragments produced.

The OAL intermediate node then encapsulates the OAL packet or fragments as attachments to OAL-encapsulated MC messages (i.e., the same as for initial packets during a multilink forwarding exchange) but also includes an OMNI Neighbor Synchronization sub-option with the "Path Change (PCH)" bit set (see: [I-D.templin-6man-omni3]. The OAL intermediate node then applies an authentication signature and includes an IPv6 Extended Fragment Header if necessary or admits the MC-encapsulated packets into the secured spanning tree. These (sub-)fragments (along with any other OAL fragments) will not be further fragmented by other OAL intermediate nodes on the path and will be reassembled by the OAL destination.

When the OAL destination begins to receive MC messages with full headers and with the PCH bit set, it assumes that the network path for this flow has changed and begins sending MC messages to the OAL source. The OAL destination sends the MC messages subject to rate limiting, and includes a Neighbor Synchronization OMNI sub-option with both the ACK and RST flags set and with the most recent OAL packet Identification recorded in the Acknowledgment field.

When the OAL source receives the MC messages with ACK and RST set, it re-initiates multilink forwarding for this flow by issuing a new MI/MR/MC or unidirectional MC exchange the same as for a new flow as specified in Section 4.13.2. The AFV state in the former path then simply becomes stale and is soon purged by the former OAL intermediate nodes.

4.16. Multicast

Each Client provides an IGMP (IPv4) [RFC2236] or MLD (IPv6) [RFC3810] proxy service for its ENETs and/or hosted applications [RFC4605] and acts 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, VPN/IPsec or Direct interfaces, and Relays also act as OMNI link PIM routers on behalf of nodes on other links/networks.

Clients on VPN/IPsec, Direct or (M)ANET underlay interfaces for which the *NET has deployed native multicast services forward IGMP/MLD messages into the *NET. The IGMP/MLD messages may be further forwarded by a first-hop *NET access router acting as an IGMP/MLD-snooping switch [RFC4541], then ultimately delivered to a *NET (FHS) 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 *NET 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.

4.16.1. 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]. The same as for unicast destinations, the 3-tuple of Source Address, Destination Address and Flow Label identifies a flow for multicast group G.

For each S belonging to a prefix reachable via X's OMNI interface, X sends an NS(AR) message (see: Section 4.13) 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 and RIOs for S.

When X processes the NA(AR) it selects one or more underlay interfaces for S and performs an MI/MR/MC 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 MNP SRA GUA or SNP GUA 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 MI/MR/MC 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 MI/MR/MC 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 an MC message to X the same as specified for unicast mobility in Section 4.15. When X receives the MC message, it updates its NCE for the MLA for source S and sends new Join messages in MI/MR/MC 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.

4.16.2. 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 MI/MR/MC message, then encapsulates the MI/MR/MC message in an OAL header and sends the message into the secured spanning tree.

For each source "S" that sends multicast traffic to group G via R, Client S* that aggregates S (or its Proxy/Server) encapsulates the original IP packets in PIM Register messages, includes the PIM Register messages in the OMNI options of MC messages, performs OAL encapsulation and fragmentation with Identification values within the receive window for Client R* that aggregates R, then performs L2 encapsulation and forwards the resulting carrier packets. Client R* may then elect to send a PIM Join to S* in the OMNI option of a MC over the secured spanning tree. This will result in an (S,G) tree rooted at S* with R as the next hop so that R will begin to receive two copies of the original IP packet; one native copy from the (S, G) tree and a second copy from the pre-existing (*, G) tree that still uses MC PIM Register encapsulation. R can then issue a MC PIM Register-stop message over the secured spanning tree to suppress the Register-encapsulated stream. At some later time, if Client S* moves to a new Proxy/Server, it resumes sending original IP packets via MC 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 4.16.1. Once the (S,G) tree is established, the listeners can send (S, G) Prune messages to R so that multicast original IP packets 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.

4.16.3. 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.

4.17. 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/INET link layer 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 4.2.4).

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

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

4.18. 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 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.

4.19. 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 FNPs 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. Alternatively, if an INET partition is transitioned to a native IP protocol version and addressing scheme 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].

4.20. Proxy/Server-Gateway Bidirectional Forwarding Detection

In environments where rapid failure recovery is essential, 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 can 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 operational practice.

4.21. 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-omni3]) that can serve as a Client ID seed for MNP prefix delegation. The Client would then be obligated to renumber its internal networks whenever its MNP changes. This should not present problems for Clients with automated network renumbering services, however it can limit the durations of ongoing sessions that would prefer to use a constant address.

5. Implementation Status

AERO/OMNI Release-3.2 was tagged on March 30, 2021, and was subject to internal testing. The implementation is not planned for public release.

A write-from-scratch reference implementation is under active internal development, with release version v0.1 tagged on December 9, 2024 and version v0.2 tagged on January 22, 2025. Future versions will be made available for public release.

6. IANA Considerations

The IANA is instructed to assign three new Codes in the "ICMPv6 "Code" Fields - Type 136 - Neighbor Advertisement" registry of the https://www.iana.org/assignments/icmpv6-parameters registry group (registration procedure is Standards Action or IESG Approval). The registry entries should appear as follows:

Code	Message Name	Reference
TBD1	Multilink Initiate (MI)	[RFCXXXX]
TBD2	Multilink Respond (MR)	[RFCXXXX]
TBD3	Multilink Control (MC)	[RFCXXXX]

Figure 5: IPv6 ND Neighbor Solicitation Codes:

[RFCXXXX] above refers to this document, while [TBD1/TBD2/TBD3] may be any available values. The values [250/251/252] are suggested to support maximum forward compatibility for future specifications that might define earlier values in this and/or other IPv6 ND message tables.

The IANA assigned UDP port number "8060" for an experimental first edition of AERO [RFC6706]. The Overlay Multilink Network Interface (OMNI) specification [I-D.templin-6man-omni3] reclaims "8060" as the service port for AERO/OMNI UDP/IP encapsulation, therefore this document makes no IANA request. (Note: although [RFC6706] was not widely implemented or deployed, it need not be obsoleted since it uses ICMPv6 message type '0' (Reserved) which implementations of this specification ignore.)

7. Security Considerations

AERO Gateways establish security associations with AERO Proxy/Servers and Relays within their local OMNI link segments using secured tunnels over underlay interfaces. The AERO Gateways of all OMNI link segments in turn configure secured tunnels with neighboring AERO Gateways for other OMNI link segments in a secured spanning tree topology. Applicable security services include IPsec [RFC4301] with IKEv2 [RFC7296], etc. (Note that secured direct point-to-point links can also be used instead of or in addition to network layer security.) Together, these services are responsible for assuring connectionless integrity and data origin authentication with optional

protection against replays for control messages that traverse the secured spanning tree.

To prevent unauthorized local applications from congesting the secured spanning tree, Proxy/Servers and Gateways configure local access controls 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 over the secured spanning tree. 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 messages they process for the same (source, target) pair, e.g., by applying IPv6 ND MAX_UNICAST_SOLICIT; MAX_NEIGHBOR_ADVERTISEMENT limits.

To prevent spoofing, Proxy/Servers MUST silently discard without responding to any unsecured IPv6 ND messages with OMNI sub-options that would otherwise affect state. Also, Proxy/Servers MUST silently discard without forwarding any original IP packets received from one of their own Clients (whether directly or following OAL reassembly) with a Source Address that does not match the Client’s MNP and/or a Destination Address that does match the Client’s MNP. Finally, Proxy/Servers MUST silently discard without forwarding any carrier packets that include an OAL packet/fragment with Source and Destination Addresses that both match the same MNP or SNP ULA/GUA.

AERO Clients that connect to secured ANETs need not apply additional security to their IPv6 ND messages, since the messages will be accepted and forwarded by a perimeter Proxy/Server that applies security over its INET-facing interface to the secured spanning tree (see above). AERO Clients that connect to MANETs or open INETs can use network and/or transport layer security services such as VPNs (e.g., IPsec tunnels) or can by some other means establish a secured 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 IPv6 ND messages as specified in [I-D.templin-6man-omni3].

End systems SHOULD apply transport or higher layer security services such as QUIC-TLS [RFC9000], TLS/SSL [RFC8446], DTLS [RFC6347], etc. to provide a level of protection comparable to critical secured Internet services. End systems that require host-based VPN services SHOULD use network and/or transport layer security services such as IPsec, TLS/SSL, DTLS, etc. AERO Proxy/Servers and Clients can also provide a network-based VPN service on behalf of end systems, e.g., if the end system 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 AERO service for MANET and 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 Proxy/Servers and potential correspondent peer nodes. Similarly, each Client must be able to determine the public key of each Proxy/Server, e.g. by consulting an online database.

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

Security considerations for IPv6 fragmentation and reassembly are discussed in [I-D.templin-6man-omni3]. 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.

Security considerations for accepting link layer ICMP messages and reflected carrier packets are discussed throughout the document.

8. 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, Felipe Magno de Almeida, Mark Andrews, Fred Baker, Amanda Baber, 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, Ron Sackman, 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:

* Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [RFC5214]

* The Subnetwork Encapsulation and Adaptation Layer (SEAL) [RFC5320]

* Virtual Enterprise Traversal (VET) [RFC5558]

* Routing and Addressing in Networks with Global Enterprise Recursion (RANGER) [RFC5720][RFC6139]

* The Internet Routing Overlay Network (IRON) [RFC6179]

* AERO, First Edition [RFC6706]

Note that these works cite numerous earlier efforts that are not included 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) Airplane Internet of Things (AIoT) and autonomy programs.

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

This work is aligned with the Boeing/Virginia Tech Network Security Institute (VTNSI) 5G MANET research program.

Honoring life, liberty and the pursuit of happiness.

9. References

9.1. Normative References

[I-D.ietf-dhc-rfc8415bis]
Mrugalski, T., Volz, B., Richardson, M., Jiang, S., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", Work in Progress, Internet-Draft, draft-ietf-dhc-rfc8415bis-12, 4 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dhc-rfc8415bis-12>.

[I-D.templin-6man-omni3]
Templin, F., "Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces", Work in Progress, Internet-Draft, draft-templin-6man-omni3-59, 23 May 2025, <https://datatracker.ietf.org/doc/html/draft-templin-6man-omni3-59>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[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>.

[RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, DOI 10.17487/RFC6890, April 2013, <https://www.rfc-editor.org/info/rfc6890>.

[RFC8028] Baker, F. and B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <https://www.rfc-editor.org/info/rfc8028>.

[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>.

[RFC9268] Hinden, R. and G. Fairhurst, "IPv6 Minimum Path MTU Hop-by-Hop Option", RFC 9268, DOI 10.17487/RFC9268, August 2022, <https://www.rfc-editor.org/info/rfc9268>.

9.2. Informative References

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

[CERF] Cerf, V., "The Catenet Model For Internetworking, IETF IEN48, http://www.postel.org/ien/pdf/ien048.pdf", July 1978.

[EUI] "IEEE 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.ietf-6man-rfc6724-update] Buraglio, N., Chown, T., and J. Duncan, "Prioritizing known-local IPv6 ULAs through address selection policy", Work in Progress, Internet-Draft, draft-ietf-6man-rfc6724-update-20, 13 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-rfc6724-update-20>.

[I-D.ietf-intarea-tunnels] Touch, J. D. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-15, 9 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-intarea-tunnels-15>.

[I-D.ietf-rtgwg-atn-bgp]
Templin, F., 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-28, 17 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-atn-bgp-28>.

[I-D.perkins-manet-aodvv2]
Perkins, C. E., Dowdell, J., Steenbrink, L., and V. Pritchard, "Ad Hoc On-demand Distance Vector Version 2 (AODVv2) Routing", Work in Progress, Internet-Draft, draft-perkins-manet-aodvv2-05, 22 November 2024, <https://datatracker.ietf.org/doc/html/draft-perkins-manet-aodvv2-05>.

[I-D.templin-6man-mla]
Templin, F., "IPv6 Addresses for Ad Hoc Networks", Work in Progress, Internet-Draft, draft-templin-6man-mla-27, 3 April 2025, <https://datatracker.ietf.org/doc/html/draft-templin-6man-mla-27>.

[I-D.templin-6man-parcels2]
Templin, F., "IPv6 Parcels and Advanced Jumbos (AJs)", Work in Progress, Internet-Draft, draft-templin-6man-parcels2-27, 21 May 2025, <https://datatracker.ietf.org/doc/html/draft-templin-6man-parcels2-27>.

[I-D.templin-intarea-parcels2]
Templin, F., "IPv4 Parcels and Advanced Jumbos (AJs)", Work in Progress, Internet-Draft, draft-templin-intarea-parcels2-18, 21 May 2025, <https://datatracker.ietf.org/doc/html/draft-templin-intarea-parcels2-18>.

[KAHN] Perry, T., "The Great Interconnector, IEEE Spectrum, https://spectrum.ieee.org/bob-kahn-2667754905", May 2024.

[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>.

[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>.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <https://www.rfc-editor.org/info/rfc3056>.

[RFC3724] Kempf, J., Ed., Austein, R., Ed., and IAB, "The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture", RFC 3724, DOI 10.17487/RFC3724, March 2004, <https://www.rfc-editor.org/info/rfc3724>.

[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>.

[RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, DOI 10.17487/RFC4007, March 2005, <https://www.rfc-editor.org/info/rfc4007>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[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>.

[RFC5614] Ogier, R. and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding", RFC 5614, DOI 10.17487/RFC5614, August 2009, <https://www.rfc-editor.org/info/rfc5614>.

[RFC5720] Templin, F., "Routing and Addressing in Networks with Global Enterprise Recursion (RANGER)", RFC 5720, DOI 10.17487/RFC5720, February 2010, <https://www.rfc-editor.org/info/rfc5720>.

[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>.

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

[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>.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, DOI 10.17487/RFC6145, April 2011, <https://www.rfc-editor.org/info/rfc6145>.

[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>.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[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>.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <https://www.rfc-editor.org/info/rfc6296>.

[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>.

[RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding", RFC 6621, DOI 10.17487/RFC6621, May 2012, <https://www.rfc-editor.org/info/rfc6621>.

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

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <https://www.rfc-editor.org/info/rfc6724>.

[RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol Version 2", RFC 7181, DOI 10.17487/RFC7181, April 2014, <https://www.rfc-editor.org/info/rfc7181>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[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>.

[RFC8175] Ratliff, S., Jury, S., Satterwhite, D., Taylor, R., and B. Berry, "Dynamic Link Exchange Protocol (DLEP)", RFC 8175, DOI 10.17487/RFC8175, June 2017, <https://www.rfc-editor.org/info/rfc8175>.

[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>.

[RFC9365] Jeong, J., Ed., "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases", RFC 9365, DOI 10.17487/RFC9365, March 2023, <https://www.rfc-editor.org/info/rfc9365>.

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 4.13 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 may be delivered over longer paths 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 without any encapsulation. In that case, the Client sends packets over the Direct link according to traffic selectors. If the Direct interface is selected, then the Client’s packets are transmitted directly to the peer without traversing an ANET/INET. If other interfaces are selected, then the Client’s packets 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 do not present a single point of failure in the architecture 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 can be 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 NCEs 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 NCEs 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 NCEs 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 communications can continue .

Therefore, links that provide many Proxy/Servers with high availability profiles are responsive to loss of individual infrastructure elements, since Clients can quickly establish new Proxy/Server relationships in event of failures.

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 >>

Differences from earlier versions:

Draft -44 to -45

* Updated SRH contents specification.

* Removed obsolete "OFH" references.

Draft -42 to -44

* Globally changed "AERO Forwarding" to "AERO Flow".

* Globally replaced "packet/parcel" with simply "packet".

Draft -39 to -42

* Removed all references to L2 fragmentation/reassembly.

* Removed unused RFC citations.

Draft -37 to -39

* Replaced Origin with Interface Attributes.

* Removed references to specific MLA candidate types.

Draft -36 to -37
* Segment Routing for the adaptation layer per [RFC8754].

Draft -35 to -36
* Discussion of DHCPv6 service model for OMNI links.

Draft -34 to -35
* Further clarification on unidirectional nature of flows.

* Introduced "Proxy/Client" archetype.

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