Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 2 September 2026

F. L. Templin, Ed.
The Boeing Company

1 March 2026

Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces
draft-templin-6man-omni3-80

Abstract

Mobile nodes that operate in air/land/sea/space domains (e.g., aircraft of various configurations, terrestrial vehicles, seagoing vessels, space systems, enterprise wireless devices, pedestrians with cell phones, etc.) configure mobile routers to connect end user network peers with Internetwork correspondents over wireless and/or wired-line data links. This document presents a multilink virtual interface specification that supports mobile node coordination with a network-based mobility service, fixed node correspondents and/or other mobile node peers. The virtual interface provides an adaptation layer service that supports secure global mobile Internetworking. This document specifies the transmission of IP packets over Overlay Multilink Network (OMNI) Interfaces.

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 2 September 2026.

Copyright Notice

Table of Contents

1. Introduction

Mobile nodes that operate in air/land/sea/space domains (e.g., aircraft of various configurations, terrestrial vehicles, seagoing vessels, space systems, enterprise wireless devices, pedestrians with cellphones, etc.) configure mobile routers that connect end user network peers with Internetwork correspondents over multiple interface connections to wireless and/or wired-line data links. These data links often have diverse performance, cost and availability properties that can change dynamically due to mobility patterns, flight phases, proximity to infrastructure, etc. The mobile router acts as a Client of a network-based Mobility Service (MS) by configuring a virtual interface over its underlay interface data link connections to support secure global mobile Internetworking.

Each Client configures a virtual network interface (termed the "Overlay Multilink Network (OMNI) Interface") as a thin layer over its underlay interfaces which may themselves connect to virtual or physical links. The OMNI interface therefore exposes a single interface abstraction to the IP layer which behaves according to the Non-Broadcast, Multiple Access (NBMA) interface principle, while each underlay interface appears as a link layer communication channel in the architecture. The OMNI interface appears as an ordinary network interface and internally employs the "OMNI Adaptation Layer (OAL)" to efficiently accommodate original IP packets of all sizes over diverse underlay interfaces with heterogeneous properties.

The OMNI interface connects to a virtual overlay termed the "OMNI link" which spans one or more concatenated Internetwork underlays such as private-use infrastructures (e.g., enterprise networks, operator networks, etc.) and/or the global public Internet itself. Together, OMNI and the OAL provide foundational elements necessary to support the "6 M's of Modern Internetworking", including:

1. Multilink - a Client's ability to coordinate multiple diverse underlay interfaces 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 an end to end topology connecting multiple diverse administrative domain network segments while maintaining seamless communications between mobile Clients and correspondents such as air traffic controllers, fleet administrators, other mobile Clients, etc.

3. Mobility - a Client'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 Clients belonging to the same interest group, but without disturbing other Clients 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 packets of all sizes 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.

Client OMNI interfaces coordinate with the MS and/or OMNI link peers through secure IPv6 Neighbor Discovery (ND) control message exchanges [RFC4861]. The MS consists of a distributed set of service elements (including fixed or mobile Proxy/Servers and other supporting infrastructure) that also configure OMNI interfaces. Automatic Extended Route Optimization (AERO) in particular provides a candidate MS compatible with the OMNI architecture [I-D.templin-6man-aero3]. AERO discusses details of ND messaging for address resolution, multilink forwarding, route optimization, mobility management, and multinet traversal, while fundamental aspects of OMNI link operation and router discovery are discussed in this document.

Clients that connect to multiple distinct OMNI links configure a corresponding OMNI interface for each link, e.g., omni0, omni1, omni2, etc. Each OMNI interface is configured over a distinct set of underlay interfaces and provides a nexus for Safety-Based Multilink (SBM) operation within an OMNI domain. The IP layer applies SBM routing to select a specific OMNI interface which then applies Performance-Based Multilink (PBM) to select appropriate underlay interfaces. Applications select SBM topologies based on IP layer Segment Routing [RFC8402], while each OMNI interface applies PBM internally based on OAL Multinet traversal.

Each OMNI interface assigns an external Link-Local Address (LLA) for use by the network layer the same as for any IPv6 interface and also assigns a different internal LLA for use by the adaptation layer. The adaptation layer then presents the appearance of a virtual router

on the same link as the network layer in the role of a virtual host allowing for the exchange of ICMPv6 control messages per [RFC4443][RFC4861]. The OMNI interface also assigns a unique Multilink Local Address (MLA) to support multilink operations.

Each OMNI link assigns one or more IP Global Unicast Address (GUA) Mobility Service Prefixes (MSPs). The MS then delegates network layer Mobile Network Prefixes (MNPs) taken from an MSP to Client end systems according to the "prefix-per-Client" model [RFC9663] [RFC9762].

Clients receive MNP delegations from Proxy/Servers through OMNI-encapsulated ICMPv6 control message exchanges over Access Networks (ANETs), Mobile Ad-hoc Networks (MANETs) and/or open Internetworks (INETs). Clients sub-delegate MNPs to downstream-attached End-user Networks (EUNs) independently of the underlay interfaces selected for upstream data transport. Each Client acts as a fixed or mobile router on behalf of EUN peers, and uses OMNI interface control messaging to coordinate with Proxy/Servers and/or other Clients. The Client registers with Proxy/Servers over each of the OMNI interface's (M)ANET/INET underlay interfaces to enroll each interface with the MS (see: Section 14). The Client can also provide (proxyed) multihop forwarding services for a recursively extended chain of other Clients and end systems connected via downstream-attached *NETs.

Proxy/Servers on the link delegate MNPs to Clients via the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) service. DHCPv6 messaging is carried as OMNI extensions to encapsulated ICMPv6 router discovery messages since each Proxy/Server supports both DHCPv6 Server and IPv6 router functions.

Peer-to-peer neighbor coordination over the OMNI link proceeds according to either the "on-link" or "off-link" IPv6 Neighbor Discovery model. For IP destinations that match on-link prefixes, the initiating peer's network layer invokes address resolution to create a network layer neighbor cache entry and also cause the interface to create a corresponding adaptation layer cache entry. The network layer then engages the OMNI interface as a "virtual Ethernet" [VETH] or "TAP" [TUNTAP] interface including the mapping of network layer addresses to link-layer addresses. The on-link model may be more suitable for general purpose systems and/or when multiple OMNI interfaces are necessary, but requires careful coordination between the network and adaptation layers.

For IP destinations that do not match on-link prefixes the network layer forwards all original IP packets to a virtual router function within the OMNI interface without disturbing the network layer neighbor cache. The OMNI interface then invokes address resolution

as an adaptation layer function. This off-link model can be coordinated over virtual Ethernet and TAP interfaces the same as for the on-link model, or can instead use a "TUN" interface [TUNTAP] without mapping network layer addresses to link-layer addresses. The off-link model may be more applicable for end user equipment such as cellphones where administrative privileges may not support creation of virtual Ethernet interfaces, or when operation may be simplified by avoiding coordination between the network and adaptation layer. Implementations are free to select the on- or off-link model on a per-prefix basis independently of other nodes on the link, as the external appearance is identical in both cases.

The OMNI link model is suitable for a wide range of use cases. For example, the International Civil Aviation Organization (ICAO) Working Group-I Mobility Subgroup is developing a future Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS) and has issued a liaison statement requesting IETF adoption [ATN] in support of ICAO Document 9896 [ATN-IPS]. The IETF IP Wireless Access in Vehicular Environments (ipwave) working group has further included problem statement and use case analysis for OMNI in [RFC9365]. Still other communities of interest include AEEC, RTCA Special Committee 228 (SC-228) and NASA programs that examine commercial aviation, Urban Air Mobility (UAM) and Unmanned Air Systems (UAS). Pedestrians with handheld mobile devices using 5G/6G wireless services, home and small office networks, enterprise networks and many others represent additional large classes of potential OMNI users.

This document specifies the transmission of original IP packets and control messages over OMNI interfaces. The operation of both IP protocol versions (i.e., IPv4 [RFC0791] and IPv6 [RFC8200]) is specified as the network layer data plane, while OMNI interfaces use ICMPv6 messaging in the control plane independently of the data plane protocol(s). OMNI interfaces also provide an adaptation layer based on encapsulation and fragmentation for transmission over heterogeneous underlay interfaces as an OAL sublayer between L3 and an underlay configured over any L2 media. OMNI and the OAL are specified in detail throughout the remainder of this document.

2. Terminology

The terminology in the normative references applies; especially, the terms "link" and "interface" are the same as defined in the IPv6 [RFC8200] and IPv6 Neighbor Discovery (ND) [RFC4861][RFC9762] specifications. This document assumes the following IPv6 ND control plane message types: Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS), Neighbor Advertisement (NA), unsolicited NA (uNA) and Redirect.

The terms "All-Routers multicast", "All-Nodes multicast" and "Subnet-Router anycast" are the same as defined in [RFC4291]. Also, IPv6 ND state names, variables and constants including REACHABLE, ReachableTime and REACHABLE_TIME are the same as defined in [RFC4861].

The term "IP" is used to refer collectively to either Internet Protocol version (i.e., IPv4 [RFC0791] or IPv6 [RFC8200]) when a specification at the layer in question applies equally to either version.

The terms (Proxy/)Client and Proxy/Server are intentionally capitalized to denote an instance of a particular node type that also configures an OMNI interface and engages the OMNI Adaptation Layer.

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 defines an "adaptation layer" positioned below the network layer and 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 following terms are defined within the scope of this document:

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 reference model, also known as "layer 3" or the "IP layer". The Network layer engages the Adaptation layer via OMNI interfaces.

Adaptation layer
: An IPv6 encapsulation and fragmentation mid-layer that adapts L3 to a diverse collection of underlay interfaces. The adaptation layer then engages an underlay network that performs UDP/IP, IP, or NULL encapsulation for transmission over underlay interface attachments to L2 media.

L2
: The Data Link layer in the OSI reference model, also known as "layer 2" or "link layer".

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 rest of the OMNI link. Physical and/or data link level security is assumed (sometimes referred to as "protected spectrum" for wireless domains). ANETs such as private enterprise networks and ground domain aviation service networks often provide multiple secured IP hops between the Client's physical 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. Clients nested deeply within a MANET often require adaptation layer proxy services from "upstream" peer Proxy/Clients located progressively nearer the MANET border. The OMNI link model naturally supports MANET Internetworking [I-D.templin-manet-inet].

Internetwork (INET)
: a connected network region with a coherent IP addressing plan that provides transit forwarding services between (M)ANETs and/or OMNI nodes that coordinate with the Mobility Service over unprotected media. Since physical and/or data link level security cannot always be assumed, security must be applied by higher layers in the architecture as necessary. The global public Internet itself is an example.

End-user Network (EUN)
: a simple or complex "downstream" network tethered to a Client as a single logical unit that travels together. The EUN could be as simple as a single link connecting a small number of hosts, or as complex as a large network with many links, routers, bridges and end user devices. The EUN 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.

*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 EUN interfaces are "downstream" interfaces that the Client uses to connect downstream EUNs and/or other Clients. Local communications between correspondents within the same *NET can often be conducted based on IPv6 MLAs [I-D.templin-6man-mla].

underlay network/interface
: a *NET interface over which an OMNI interface is configured. The network layer engages the OMNI interface as an ordinary network interface, and the adaptation layer engages each underlay interface as a link layer conduit. The underlay includes UDP/IP, IP or NULL encapsulations for data units transferred between the adaptation and link layers.

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
: a Non-Broadcast, Multiple Access (NBMA) virtual overlay configured over one or more INETs and their connected (M)ANETs/EUNs. An OMNI link may comprise multiple distinct "segments" joined by "bridges" the same as for any link; the addressing plans in each segment may be mutually exclusive and managed by different administrative entities. Proxy/Servers and other infrastructure elements extend the link to support communications between Clients as single-hop neighbors.

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 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 "external" LLA and Ethernet link-layer address the same as for any IPv6 interface, assigns a different "internal" LLA and link-layer address to support a virtual internal router entity, and assigns an MLA for adaptation layer addressing over underlay interfaces. Since OMNI interface MLAs are managed for uniqueness and LLAs are used for node-local operations only, OMNI interfaces assume Optimistic Duplicate Address Detection (DAD) per [RFC4429].

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 OMNI link SRT traversal. The OAL presents a new layer in the Internet architecture known simply as the "adaptation layer". The OMNI link is an example of a limited domain [RFC8799] at the adaptation layer although its segments may be joined over open Internetworks in the underlay.

OMNI Option
a pseudo IPv6 ND option providing multilink parameters for the OMNI interface as specified in Section 10. The OMNI option is appended to the end of a control message during OAL encapsulation such that it appears immediately following the final message option or composite packet extension.

(OMNI) Client
a network platform/device mobile router or end system that configures one or more OMNI interfaces over distinct sets of underlay interfaces grouped as logical OMNI link units. The Client coordinates with the Mobility Service via upstream networks over *NET interfaces, and may also serve as a Proxy for other Clients on downstream *NETs. The Client’s upstream network interface addresses and performance characteristics may change over time (e.g., due to node mobility, link quality, etc.) while other Clients on downstream networks can engage the (upstream) Client as a Proxy.

(OMNI) Proxy/Server
a segment management topology edge node that configures an OMNI interface and connects Clients to the Mobility Service. As a server, the Proxy/Server responds directly to some Client IPv6 ND messages. As a proxy, the Proxy/Server forwards other Client IPv6 ND messages to other Proxy/Servers and Clients. As a router, the

Proxy/Server provides a forwarding service for ordinary data messages that may be essential in some environments and a last resort in others. Proxy/Servers at (M)ANET boundaries configure both an (M)ANET downstream interface and *NET upstream interface, while INET-based Proxy/Servers configure only an INET interface.

First-Hop Segment (FHS) Proxy/Server
a Proxy/Server connected to the source Client's *NET that forwards OAL packets sent by the source into the segment management topology. FHS Proxy/Servers act as intermediate forwarding systems to facilitate RS/RA-based Prefix Delegation exchanges between Clients and Mobility Anchor Point (MAP) Proxy/Servers.

Last-Hop Segment (LHS) Proxy/Server
a Proxy/Server connected to the target Client's *NET that forwards OAL packets received from the segment management topology to the target.

Mobility Anchor Point (MAP) Proxy/Server
a Proxy/Server selected by the Client that provides a designated router service for any *NET underlay networks that register the Client's Mobile Network Prefix (MNP). Since all Proxy/Servers provide equivalent services, Clients normally select the first FHS Proxy/Server they coordinate with to serve as the MAP. However, the MAP can instead be any available Proxy/Server for the OMNI link, i.e., and not necessarily one of the Client's FHS Proxy/Servers. This flexible arrangement supports a fully distributed mobility management service.

Segment Routing Topology (SRT)
a multinet forwarding region configured over one or more INETs between the FHS Proxy/Server and LHS Proxy/Server, where each INET appears as a segment in a virtual overlay link. The SRT spans the OMNI link on behalf of communicating peer nodes in a manner outside the scope of this document (see: [I-D.templin-6man-aero3]).

Mobility Service (MS)
a mobile routing service that tracks Client movements and ensures that Clients remain continuously reachable even across mobility events. The MS consists of the set of all Proxy/Servers plus all other OMNI link supporting infrastructure nodes. Specific MS details are out of scope for this document, with an example found in [I-D.templin-6man-aero3].

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 Network Prefixes (MNPs) 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 also be used subject to certain limitations (see: Section 8). OMNI links that connect to the global Internet advertise their MSPs to their interdomain routing peers.

Mobile Network Prefix (MNP)
: a longer IP prefix delegated from an MSP (e.g., 2001:db8:1000:2000::/56, 2002:192.0.2.8::/46, etc.) and assigned to a Client. Clients receive MNPs from MAP Proxy/Servers and sub-delegate them to routers in downstream EUNs.

Foreign Network Prefix (FNP)
: a global IP prefix not covered by a MSP and assigned to a link or network outside of the OMNI domain.

Subnet Router Anycast (SRA) Address
: An IPv6 address taken from an FNP/MNP in which the remainder of the address beyond the prefix is set to the value "all-zeros". For example, the SRA for 2001:db8:1::/64 is simply 2001:db8:1:: (i.e., with the 64 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].

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 underlay encapsulation, or following underlay 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.

carrier packet
an OAL packet or fragment submitted for underlay interface encapsulation. OAL nodes exchange carrier packets over underlay interfaces in a hop-by-hop fashion beginning with the OAL source, then continuing over any OAL intermediate nodes and ending with the OAL destination. Each intermediate hop removes the underlay encapsulations of the previous hop and inserts underlay encapsulations appropriate for the next hop. Carrier packets may themselves be subject to fragmentation and reassembly in some IPv4 underlay networks at a layer below the OAL. Carrier packets sent over unsecured paths use OMNI protocol underlay encapsulations, while those sent over secured paths use security encapsulations such as IPsec [RFC4301]. (The term "carrier" honors agents of the service postulated by [RFC1149] and [RFC6214].)

OAL source
a node that configures an OMNI interface acts as an OAL source when it encapsulates original IP packets to form OAL packets, then performs OAL fragmentation and encapsulation to create carrier packets. Every OAL source is also an OMNI link ingress.

OAL destination
a node that configures an OMNI interface acts as an OAL destination when it decapsulates carrier packets (following reassembly if necessary), then performs OAL reassembly/ decapsulation to derive the original IP packet. Every OAL destination is also an OMNI link egress.

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 original OAL packet/fragment, then re-encapsulates in new underlay 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.

Multilink
: a Client OMNI interface's manner of managing multiple diverse *NET underlay interfaces as a single logical unit. The OMNI interface provides a single unified interface to the network layer, while underlay interface selections are performed on a per-flow basis considering traffic selectors such as Traffic Class, Flow Label, application policy, signal quality, cost, etc. Multilink selections are coordinated in both the outbound and inbound directions based on source/target underlay interface pairs.

Multinet
: an intermediate system's manner of spanning multiple diverse IP Internetwork and/or private enterprise network "segments" through OAL encapsulation. Multiple diverse Internetworks (such as the global public IPv4 and IPv6 Internets) can serve as transit segments in an end-to-end OAL forwarding path through intermediate system concatenation of SRT network segments. This OAL concatenation capability provides benefits such as supporting IPv4/IPv6 transition and coexistence, joining multiple diverse operator networks into a cooperative single service network, etc. See: [I-D.templin-6man-aero3] for further information.

Multihop
: an iterative relaying of carrier packets between Clients over an OMNI underlay interface technology (such as omnidirectional wireless) without support of fixed infrastructure. Multihop services entail Client-to-Client relaying within a Mobile/ Vehicular Ad-hoc Network (MANET/VANET) for Vehicle-to-Vehicle (V2V) communications and/or for Vehicle-to-Infrastructure (V2I) "range extension" where Clients within range of communications infrastructure elements provide forwarding services for other Clients.

Mobility
any action that results in a change to a Client underlay interface address. The change could be due to, e.g., a handover to a new wireless base station, loss of link due to signal fading, an actual physical node movement, etc.

Safety-Based Multilink (SBM)
A means for ensuring fault tolerance through redundancy by connecting multiple OMNI interfaces within the same domain to independent routing topologies (i.e., multiple independent OMNI links). SBM can also include non-terrestrial physical link types such as low earth orbit satellite services in a hyperconnected service model.

Performance Based Multilink (PBM)
A means for selecting one or more underlay interface(s) for carrier packet transmission and reception within a single OMNI interface.

OMNI Domain
The set of all SBM/PBM OMNI links that collectively provides services for a common set of MSPs. All OMNI links within the same domain configure, advertise and respond to the SRA address(es) corresponding to the MSP(s) assigned to the domain.

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 (Source Address, Destination Address, Flow Label) enables efficient IPv6 flow classification. All packets of the same flow will receive identical forwarding services over the OMNI link and must therefore have compatible traffic classifications. 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. See: [I-D.templin-6man-aero3] for further discussion.

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 underlay 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. See: [I-D.templin-6man-aero3] for further discussion.

AERO Flow Vector Index (AFVI)
A 2 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 underlay 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.

underlay 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 underlay network partition. Common underlay encapsulation combinations include UDP, IP and/or Ethernet using a port/protocol/type number for OMNI.

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

OAL Fragment Size (OFS)
the current OAL source fragmentation size for a given flow which must be no smaller than 1024 octets and should be no larger than 65279 octets to allow sufficient space for OAL and underlay encapsulations. (OFS pertains to the fragment payload immediately following the fragment header; if OAL extension headers are included following the first fragment header a slightly larger minimum OFS may be necessary to accommodate maximum-sized packets.) Each OAL source maintains OFS in an AERO Flow Vector (AFV) for each independent flow. The OAL source discovers larger OFS sizes through dynamic probing the same as defined for Maximum Packet Size (MPS) probing per Section 4.4 of [RFC8899] and should adaptively maintain the best possible OFS for each flow according to current network conditions.

3. Requirements

OMNI interfaces maintain the same Conceptual Data Structures as for any IPv6 interface, including the Neighbor Cache, Destination Cache, Prefix List and Default Router List [RFC4861]. These data structures are affected by the exchange of IPv6 ND messages in the same manner as for any IPv6 interface. The OMNI interface also internally configures an extension to the Neighbor Cache that includes adaptation layer information managed in parallel with network layer information. This document refers to the distinct neighbor cache views as the Adaptation Layer Neighbor Cache (ALNC) and Network Layer Neighbor Cache (NLNC).

Client and Proxy/Server OMNI interfaces maintain per-destination state in Destination Cache Entries (DCEs) as a level of indirection into per-neighbor state in Neighbor Cache Entries (NCEs) which include both network layer (NLNCE) and adaptation layer (ALNCE) information. The IPv6 ND Protocol Constants associated with these caches defined in Section 10 of [RFC4861] apply also to this document.

OMNI interfaces should limit the size of their 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 cause a control message to exceed this size, the OMNI interface should forward the information as multiple smaller 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.

L3, the adaptation layer and the underlay each include distinct packet Identification numbering spaces. The adaptation layer employs an extended Identification numbering space that is distinct from the L3 and underlay spaces, with an Identification value appearing in an IPv6 Extended Fragment Header [I-D.templin-6man-ipid-ext2] or an OMNI Compressed Header (OCH) (see: Section 6.5) in each adaptation layer encapsulation.

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. Overlay Multilink Network (OMNI) Interface Model

The IP layer engages the OMNI interface as a virtual interface configured over one or more underlay interfaces, which may be physical (e.g., an aeronautical radio link, a cellular wireless link, etc.) or virtual (e.g., an internet-layer or higher-layer "tunnel"). The OMNI interface architectural layering model is the same as in [RFC5558][RFC7847], and augmented as shown in Figure 1. The network layer therefore engages the OMNI interface as a single L3 interface nexus for multiple underlay interfaces that appear as L2 communication channels in the architecture.

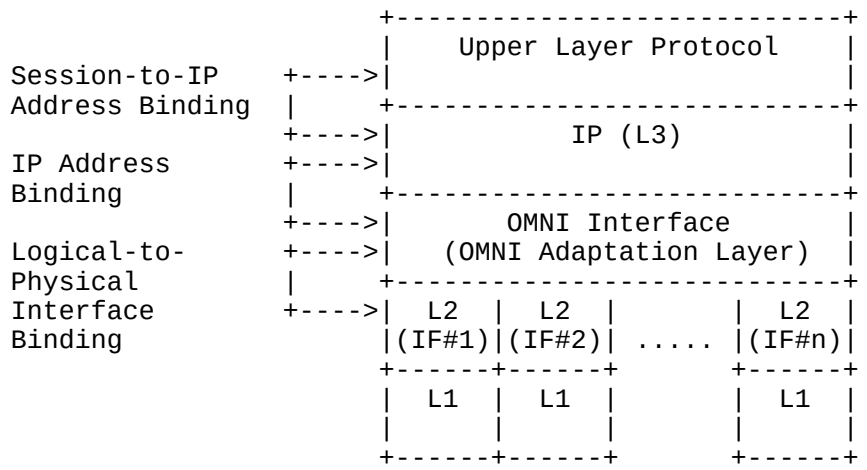

Figure 1: OMNI Interface Architectural Layering Model

Each underlay interface provides an L2/L1 abstraction according to one of the following models:

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

* (M)ANET interfaces connect to a (M)ANET that is connected to the open INET by Proxy/Servers. The (M)ANET interface may be either on the same link segment as a Proxy/Server, or separated from a Proxy/Server by multiple adaptation layer and/or underlay/L2 hops. (Note that NATs may appear internally within a (M)ANET or on the Proxy/Server itself and may require NAT traversal the same as for the INET case.)

* EUN interfaces connect a Client's downstream-attached networks, where the Client provides forwarding services for EUN end system communications to remote peers. An EUN may 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 *NET.

* VPN interfaces use security encapsulations (e.g. IPsec tunnels) over underlay networks to connect Client, Proxy/Server or other critical infrastructure nodes. VPN interfaces provide security services at sub-network layers of the architecture with securing properties similar to Direct point-to-point interfaces.

* Direct point-to-point interfaces securely connect Clients, Proxy/Servers and/or other critical infrastructure nodes 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.

The OMNI interface forwards original IP packets from the network layer using the OMNI Adaptation Layer (OAL) (see: Section 5) as an encapsulation and fragmentation sublayer service. This "OAL source" then further encapsulates the resulting OAL packets/fragments in underlay network headers (e.g., UDP/IP, IP-only, Ethernet-only, etc.) to create encapsulated "carrier packets" for transmission over underlay interfaces. The target OMNI interface then receives the carrier packets from underlay interfaces and performs decapsulation to obtain OAL packets/fragments.

If the resulting OAL packets/fragments are addressed to itself, the OMNI interface performs reassembly/decapsulation as an "OAL destination" and delivers the original IP packet to the network layer. If the OAL packets/fragments are addressed to another node, the OMNI interface instead re-encapsulates them in new underlay network headers as an "OAL intermediate system" then forwards the resulting carrier packets over an underlay interface. The OAL source and OAL destination appear as "neighbors" on the OMNI link, while OAL intermediate systems provide a virtual bridging service that joins the segments of a (multinet) Segment Routing Topology (SRT).

The OMNI interface transports carrier packets over either secured or unsecured underlay interfaces to access the secured/unsecured OMNI link spanning trees as discussed further throughout the document. Carrier packets that carry control plane messages over secured underlay interfaces use sub-network securing services such as IPsec, direct encapsulation over secured point-to-point links, etc. Carrier packets that carry data plane messages over unsecured underlay interfaces instead use encapsulations appropriate for public or private Internetworks.

The OMNI interface and its OAL can forward original IP packets over underlay interfaces while including/omitting various lower layer encapsulations including OAL, UDP, IP and (ETH)ernet or other link layer header. The network layer can also engage underlay interfaces directly while bypassing the OMNI interface entirely when necessary. This architectural flexibility may be beneficial for underlay interfaces (e.g., some aviation data links) for which encapsulation overhead is a primary consideration. OMNI interfaces that send original IP packets directly over underlay interfaces without invoking the OAL can only reach peers located on the same OMNI link segment. Source Clients can instead use the OAL to coordinate with target Clients in the same or different OMNI link segments by sending initial carrier packets to a First-Hop Segment (FHS) Proxy/Server. The FHS Proxy/Sever then sends the carrier packets into the SRT spanning tree, which transports them to a Last-Hop Segment (LHS) Proxy/Server for the target Client.

The OMNI interface encapsulation/decapsulation layering possibilities are shown in Figure 2 below. An imaginary vertical lines drawn between the Network Layer at the top of the figure and Underlay Interfaces at the bottom of the figure then allowed to slide horizontally either to the right or left illustrates the various encapsulation/decapsulation layering combination possibilities. Common combinations include IP-only (i.e., direct access to underlay interfaces with or without using the OMNI interface), IP/IP, IP/UDP/IP, IP/UDP/IP/ETH, IP/OAL/UDP/IP, IP/OAL/UDP/ETH, etc.

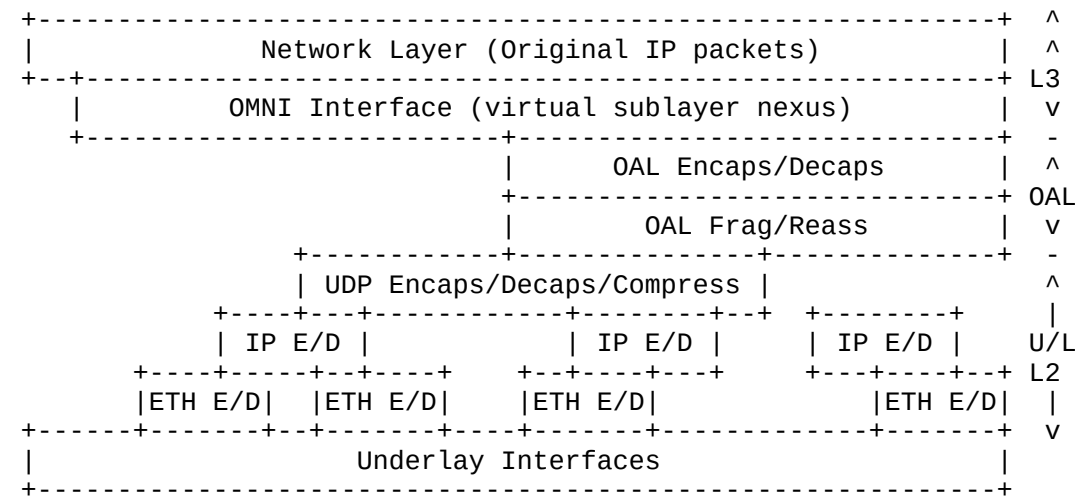

Figure 2: OMNI Interface Layering

The OMNI/OAL model gives rise to a number of opportunities:

* Clients coordinate with the MS and receive MNP delegations through ICMPv6 control plane message exchanges with Proxy/Servers and hence assured unique. Since the MLAs assigned to the OMNI interface are managed for uniqueness, Duplicate Address Detection (DAD) and Multicast Listener Discovery (MLD) messaging is serviced locally within the OMNI interface without disturbing other nodes on the link.

* underlay interfaces on the same L2 link segment as a Proxy/Server can omit UDP/IP underlay encapsulations for communications coordinated entirely over the OMNI interface.

* as underlay interface properties change (e.g., link quality, cost, availability, etc.), any active interface can be used to update the profiles of multiple additional interfaces in a single message. This allows for timely adaptation and service continuity under dynamically changing conditions.

* coordinating underlay interfaces in this way allows them to be represented in a unified MS profile with provisions to support the "6 M’s of Modern Internetworking".

* header compression and path MTU determination is conducted on a per-flow basis, with each flow adapting to the best performance profiles and path selections.

* exposing a single virtual interface abstraction to the network layer allows for multilink operation (including QoS based link selection, carrier packet replication, load balancing, etc.) in the underlay while still permitting network layer traffic shaping based on, e.g., Traffic Class, IP address range, transport protocol port number, Flow Label, etc.

* the OMNI interface supports multinet traversal over the SRT when communications across different administrative domain network segments are necessary. This mode of operation would not be possible via direct communications over the underlay interfaces themselves.

* the OAL supports lossless and adaptive path MTU mitigations not available for communications directly over the underlay interfaces themselves. The OAL supports "packing" of multiple original IP payload packets within a single OAL "composite packet" and also supports transmission of IP packets of all sizes up to and including (advanced) jumbograms.

* the OAL assigns per-packet Identification values that allow for adaptation/link layer reliability and data origin authentication.

* The network layer engages the OMNI interface as a point of connection to the OMNI link; if there are multiple OMNI links, the network layer will engage multiple OMNI interfaces.

* Multiple independent OMNI interfaces can be used for increased fault tolerance through Safety-Based Multilink (SBM), with Performance-Based Multilink (PBM) applied within each interface.

* Multiple independent OMNI links can be joined together into a single link without requiring renumbering of infrastructure elements, since the MNPs assigned by Proxy/Servers of the different links will be mutually exclusive.

* The concept of OMNI endpoint intermediate systems supports logical partitioning (or "clustering") within MANETs without requiring address aggregation. Instead, MANET routing within each partition/cluster is based on MLA host routes (i.e., MLA::/128) that are not redistributed into other partitions/clusters. A hierarchy of partitions/clusters then connects MANET Clients to an FHS Proxy/Server. Packet forwarding between distinct partitions/clusters is accomplished using the Segment Routing Header (SRH).

* Clients can configure OMNI interfaces in parallel with physical link types such as non-terrestrial and/or low-earth orbit satellite services in a hyperconnected architecture. Each interface configures a distinct set of IP addresses so that upper layers experience a multi-addressed network layer.

Figure 3 depicts the architectural model for a source Client with an attached EUN connecting to the OMNI link via multiple independent *NETs. The Client's OMNI interface forwards adaptation layer encapsulated ICMPv6 solicitation messages over available *NET underlay interfaces using any necessary underlay encapsulations. The ICMPv6 messages traverse the *NETs until they reach an FHS Proxy/Server (FHS#1, FHS#2, ..., FHS#n), which returns an ICMPv6 advertisement message and/or forwards a proxyed version of the message over the SRT to an LHS Proxy/Server near the target Client (LHS#1, LHS#2, ..., LHS#m). The Hop Limit in ICMPv6 messages is not decremented due to encapsulation; hence, the source and target Client OMNI interfaces appear to be attached to a shared NBMA link.

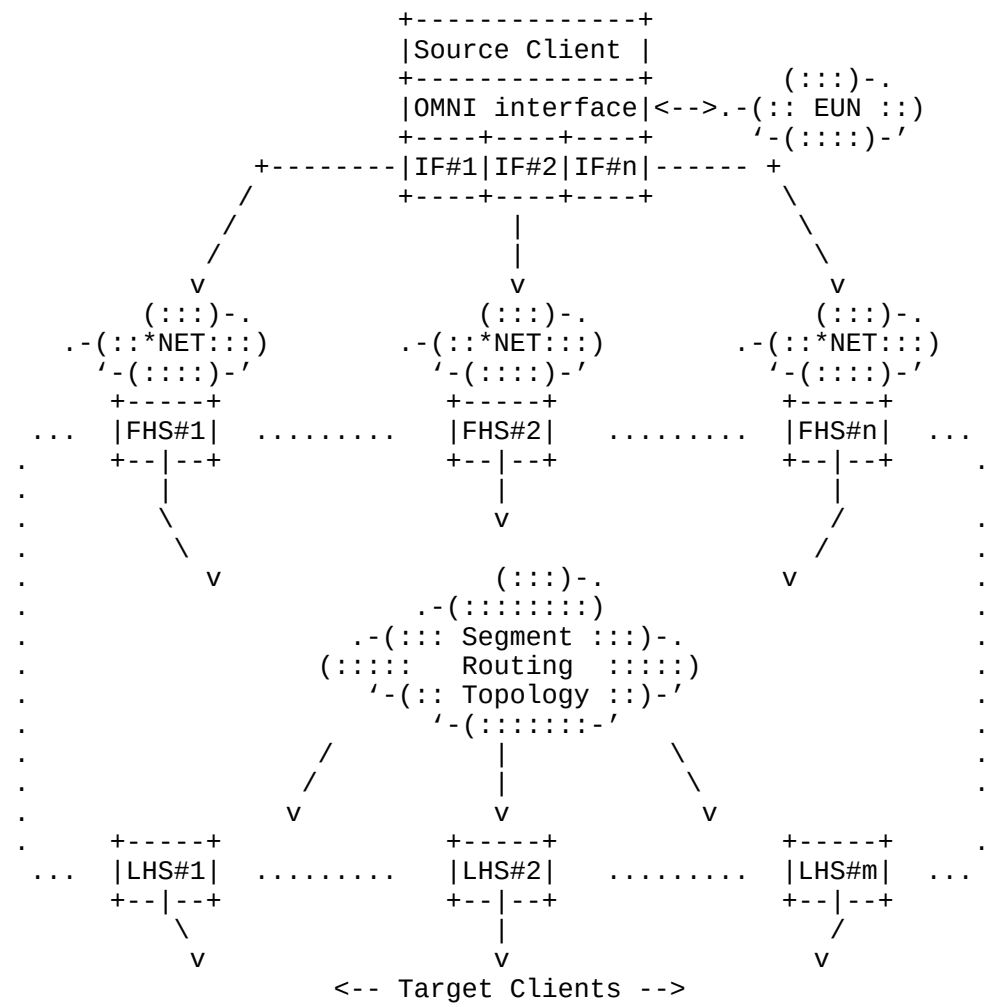

Figure 3: Source/Target Client Coordination over the OMNI Link

After the initial control message exchange, the source Client (as well as any nodes on its attached EUNs) can send carrier packets to the target Client via the OMNI interface. OMNI interface multilink services will forward the carrier packets via FHS Proxy/Servers for the correct underlay *NETs. The FHS Proxy/Server then re-encapsulates the carrier packets and forwards them over the SRT which delivers them to an LHS Proxy/Server, and the LHS Proxy/Server in turn re-encapsulates and forwards them to the target Client. (Note that when the source and target Client are on the same SRT segment, the FHS and LHS Proxy/Servers may be one and the same.)

Mobile Clients select a MAP Proxy/Server (not shown in the figure), which will often be one of their FHS Proxy/Servers but may instead be any Proxy/Server on the OMNI link. Clients then register all of their *NET underlay interfaces with the MAP Proxy/Server via per interface FHS Proxy/Servers in a pure proxy role. The MAP Proxy/ Server then advertises the Client's MLA and MNPs into the OMNI link routing system, and the Client can quickly migrate to a new MAP Proxy/Server if the former becomes unresponsive.

Clients therefore use Proxy/Servers as bridges into the SRT to reach OMNI link correspondents via a spanning tree established in a manner outside the scope of this document. Proxy/Servers forward critical MS control messages via the secured spanning tree and forward other messages via the unsecured spanning tree (see: Security Considerations). When AERO route optimization is applied, Clients can instead forward directly to correspondents in the same SRT segment to reduce Proxy/Server and/or Gateway load.

Note: Original IP packets sent into an OMNI interface will receive consistent consideration according to their size as discussed in the following sections, while those sent directly over underlay interfaces that exceed the underlay network path MTU are dropped with an ordinary ICMP Packet Too Big (PTB) message returned. These PTB messages are subject to loss the same as for any non-OMNI IP interface [RFC2923].

5. OMNI Interface Maximum Transmission Unit (MTU)

The OMNI interface observes the link nature of tunnels, including the Maximum Transmission Unit (MTU), Effective MTU to Send (EMTU_S), Effective MTU to Receive (EMTU_R) and the role of fragmentation and reassembly [I-D.ietf-intarea-tunnels]. The OMNI interface is configured over one or more underlay interfaces as discussed in Section 4, where underlay links and network paths may have diverse MTUs. OMNI interface considerations for accommodating original IP packets of various sizes are discussed in the following sections.

IPv6 underlay interfaces are REQUIRED to configure a minimum MTU of 1280 octets and a minimum EMTU_R of 1500 octets [RFC8200]. Therefore, the minimum IPv6 path MTU is 1280 octets since routers on the path are not permitted to perform network fragmentation even though the destination is required to reassemble more. The network therefore MUST forward original IP packets as large as 1280 octets without generating an IPv6 Path MTU Discovery (PMTUD) Packet Too Big (PTB) message [RFC8201].

IPv4 underlay interfaces are REQUIRED to configure a minimum MTU of 68 octets [RFC0791] and a minimum EMTU_R of 576 octets [RFC0791][RFC1122]. However, links that configure small MTUs are likely to have low-end performance and occur only at the extreme network edges while higher-performance interior network links commonly configure MTUs no smaller than 1280 octets and EMTU_Rs no smaller than 1500 octets [RFC3819].

The OMNI interface itself sets an "unlimited" MTU of (2**32 - 1) octets. The network layer therefore unconditionally admits all original IP packets into the OMNI interface, where the adaptation layer accommodates them if possible according to their size. The OAL source then invokes adaptation layer encapsulation/fragmentation services to transform all original IP packets no larger than 65535 octets into OAL packets/fragments. The OAL source then applies underlay encapsulation to form carrier packets and finally forwards the carrier packets via underlay interfaces.

When the OAL source performs IPv6 encapsulation and fragmentation (see: Section 6), the Payload Length field limits the maximum-sized original IP packet that the OAL can accommodate while applying IPv6 fragmentation to (2**16 - 1) = 65535 octets (i.e., not including the OAL encapsulation header lengths). The OAL source is also permitted to forward packets larger than this size as a best-effort delivery service if the path can accommodate them through "jumbo-in-jumbo" encapsulation; otherwise, the OAL source discards the packet and arranges to return a PTB "hard error" to the original source (see: Section 6.9).

Each OMNI interface therefore sets a minimum EMTU_R of 65535 octets (plus the length of the OAL encapsulation headers), and each OAL destination must consistently either accept or reject still larger whole packets that arrive over any of its underlay interfaces according to their size. If an underlay interface presents a whole packet larger than the OAL destination is prepared to accept (e.g., due to a buffer size restriction), the OAL destination discards the packet and arranges to return a PTB "hard error" to the OAL source which in turn forwards a translated PTB to the original source (see: Section 6.9).

6. The OMNI Adaptation Layer (OAL)

The OMNI interface forwards original IP packets from the network layer for transmission over underlay interfaces. The OMNI Adaptation Layer (OAL) acting as the OAL source then replaces the virtual Ethernet header with an IPv6 encapsulation header to form OAL packets. The OMNI interface then applies source fragmentation to break these OAL packets into IPv6 fragments suitable for underlay

encapsulation and transmission as carrier packets.

The carrier packets then traverse one or more underlay networks spanned by OAL intermediate systems in the SRT. Each successive OAL intermediate system then re-encapsulates by removing the first underlay network encapsulations and appending encapsulations appropriate for the next underlay network. (This process supports the multinet concatenation capability needed for joining multiple diverse networks.) Following any forwarding by OAL intermediate systems, the carrier packets eventually arrive at the OAL destination.

When the OAL destination receives the carrier packets, it discards the underlay encapsulations and reassembles the resulting OAL fragments into an OAL packet as described in Section 6.3. The OAL destination next replaces the OAL packet IPv6 encapsulation header with a virtual Ethernet header to obtain the original IP packet for delivery to the network layer via the OMNI interface. The OAL source may be either the source Client or its FHS Proxy/Server, while the OAL destination may be either the LHS Proxy/Server or the target Client. Proxy/Servers (and SRT Gateways per [I-D.templin-6man-aero3]) may also serve as OAL intermediate systems.

The OAL presents an OMNI sublayer abstraction similar to ATM Adaptation Layer 5 (AAL5). Unlike AAL5 which performs segmentation and reassembly with fixed-length 53 octet cells over ATM networks, however, the OAL uses IPv6 encapsulation, fragmentation and reassembly with larger variable-length cells over heterogeneous networks. (The OAL also does not include a trailing CRC since each IPv6 fragment is covered by hop-by-hop link layer integrity checks.) Detailed operations of the OAL are specified in the following sections.

6.1. OAL Source Encapsulation and Fragmentation

When the network layer forwards an original IP packet into the OMNI interface, it either sets the TTL/Hop Limit for locally-generated packets or decrements the TTL/Hop Limit according to standard IP forwarding rules. The OAL source next creates an "OAL packet" by replacing the virtual Ethernet header with an IPv6 encapsulation header per [RFC2473]. The OAL source sets the IPv6 encapsulation header Version to "OMNI-IP6" (see: Section 6.2) and Next Header to TBD1 (see: IANA Considerations).

When the OAL source performs IPv6 encapsulation, it sets the IPv6 header "Flow Label" as specified in [RFC6438]. The OAL source next copies the "Type of Service/Traffic Class Differentiated Service Code Point (DSCP)" [RFC2474][RFC2983] and "Explicit Congestion Notification (ECN)" [RFC3168] values in the original packet's IP header into the corresponding fields of the OAL IPv6 header.

For original IP packets with DSCP '111111' (including ordinary network control/data plane messages), the OAL source resets the OAL IPv6 encapsulation header DSCP to '110111'. The OAL source instead sets the IPv6 encapsulation header DSCP to '111111' for adaptation layer control plane messages that must be processed by all OAL intermediate systems on the path including both endpoints and transits. These DSCP values belong to the "Pool 2 Experimental and Local Use (EXP/LU)" range reserved in [RFC2474] and correspond to Network/Internetwork Control precedence in [RFC0791].

The OAL source next sets the IPv6 header Payload Length to the length of the original IP packet and sets Hop Limit to a value that is sufficiently large to support loop-free forwarding over multiple concatenated OAL intermediate hops. The OAL source next selects OAL IPv6 Source and Destination Addresses associated with its own OMNI interface and the OMNI interface of the target. (See: Section 8 for Source and Destination Address selection requirements.)

The OAL source next inserts any necessary extension headers following the IPv6 header as specified in Section 6.4. For OAL data plane packets, the source first inserts any per-fragment extension headers (e.g., Hop-by-Hop, Routing, etc.) then inserts an IPv6 Extended Fragment Header (see: [I-D.templin-6man-ipid-ext2]) with an extended OAL packet Identification. Note that the extension header insertions could cause the IPv6 Payload Length to exceed 65535 octets by a small amount when the original IP packet is (nearly) the maximum length.

The OAL source then source-fragments the OAL packet if necessary according to an OAL Fragment Size (OFS) maintained in AERO Flow Vectors (AVFs) for each independent flow. (The OAL source encapsulates payloads that are no larger than the OFS and original IP packets larger than 65535 octets as "atomic fragments".) OAL fragments prepared by the source must not be fragmented further by OAL intermediate systems on the path to the OAL destination.

OAL packets that contain original IP packets no larger than 65535 octets are subject to OAL source fragmentation using the IPv6 Extended Fragment Header (EFH) fragmentation specification [I-D.templin-6man-ipid-ext2] with the exception that the IPv6 Payload Length may exceed 65535 by at most the length of the extension headers. For each independent flow, the OAL source MUST set OFS to a

size no smaller than 1024 octets and thereafter adjust OFS according to dynamic network control message feedback. The OAL source SHOULD limit OFS to a size no larger than 65279 octets (i.e., 256 octets less than the maximum length IPv6 payload packet to allow room for encapsulation) unless it has assurance that the path can accommodate a larger size. (Note: the minimum size ensures that OAL fragments can be accommodated over any potential combination of IPv4/IPv6 underlay network paths where transit for smaller sizes is assured without probing, while the maximum size observes the 65535 octet limitation for conventional IP packets.)

When the OAL source performs fragmentation, it SHOULD produce the minimum number of fragments under current OFS constraints. The fragments produced MUST be non-overlapping and the portion of each non-final fragment following the IPv6 Extended Fragment Header MUST be equal in length while that of the final fragment MAY be smaller and MUST NOT be larger.

For each consecutive OAL fragment beginning with the first, the OAL source then writes a monotonically-increasing "ordinal" value between 0 and 63 in the Extended Fragment Header Index field. Specifically, the OAL source writes the ordinal value ’0’ for the first fragment, ’1’ for the first non-first fragment, ’2’ for the next, ’3’ for the next, etc. up to the final fragment. The final fragment may assign an ordinal as large as ’63’ such that at most 64 fragments are possible.

The OAL source finally encapsulates the fragments in any underlay headers necessary to form carrier packets for transmission over underlay interfaces, while retaining the fragments and their ordinal numbers (i.e., #0, #1, #2, etc.) for a brief period to support adaptation layer retransmissions (see: Section 6.8). OAL fragment and carrier packet formats are shown in Figure 4.

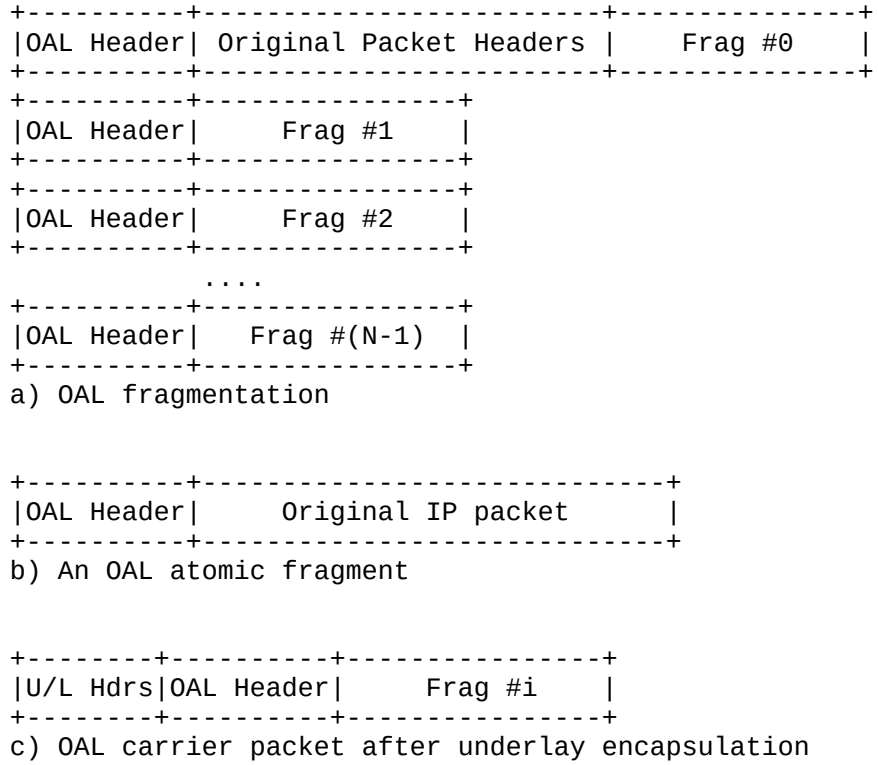

Figure 4: OAL Fragments and Carrier Packets

After establishing AFV state in the forward path for a given flow, the OAL source dynamically manages the per-flow OFS by continually probing the forward path to the OAL destination beginning with a size no smaller than 1024 octets and increasing to progressively larger sizes per [RFC8899]. In this process, the OAL source acts as a datagram packetization layer for the flow when it applies OAL encapsulation, fragmentation and header compression.

The OAL source creates a probe by setting the P flag in the Type 1 OMNI Compressed Header (OCH1) (see: Section 6.5) of a probe packet for the flow. For probes that advance the OFS to a larger size, the probe packet can include discard data (e.g., an IP packet with Next Header/Protocol set to 59 ("No Next Header"), a UDP packet with service port number set to 9 ("discard"), etc.) or live protocol data with null padding. For probes that confirm the current OFS, the probe packet can instead entirely include live protocol data. The OAL source then admits the probe for underlay encapsulation and transmission.

When the OAL destination receives the probe, it returns an OAL-encapsulated secured control message to the OAL source with an OMNI option that includes a FRAGREP sub-option (see: Section 10.2.15). The OAL destination then returns the secured control message to the OAL source without marking it for examination by OAL intermediate systems.

When the OAL source receives the secured control message, it first determines that the message is authentic. The OAL source can then tentatively advance OFS for this flow to the probe size but should maintain an ongoing stream of additional probes for the flow to confirm the current OFS and/or to advance to still larger OFS values. The OAL source may additionally receive MTU soft error feedback from an OMNI destination or intermediate system and should compensate accordingly as discussed in Section 6.9.

6.2. Underlay Encapsulation and Re-Encapsulation

The OAL source or intermediate system next encapsulates each OAL fragment (with either full or compressed headers) in underlay encapsulation headers to create a carrier packet. The OAL source or intermediate system (i.e., the underlay source) includes a full/compressed IP header if there may be routers on the path to the underlay destination. The source appends any additional underlay IP encapsulation sublayers (e.g., IPsec AH/ESP, an IP Hop-by-Hop option for jumbo-in-jumbo encapsulation, etc.) and also includes a UDP header if NATs and/or filtering middleboxes might occur on the path. The underlay source finally includes an actual L2 header such as an Ethernet header for Ethernet-compatible links.

The underlay source encapsulates the OAL information immediately following the innermost underlay header. The underlay source next interprets the first 4 bits following the underlay headers as a Type field that determines the type of OAL header that follows. The underlay source sets Type to (OMNI-IP6) for an uncompressed IPv6 OMNI Full Header or (OMNI-OCH1/2) for an OMNI Compressed Header (OCH1/2) as specified in Section 6.5. Other Type values may also appear as specified in Section 6.5.

The underlay source prepares the underlay encapsulation headers for OAL packets/fragments as follows:

* For UDP/IP encapsulation, the underlay source sets the UDP source port to 8060 (i.e., the port number reserved for AERO/OMNI). When the underlay destination is a Proxy/Server or Gateway, the underlay source sets the UDP Destination Port to 8060; otherwise, the underlay source sets the UDP Destination Port to its cached port number value for the peer. The underlay source next sets the UDP Length the same as specified in [I-D.ietf-tsvwg-udp-options].

* The underlay source then sets the IP {Protocol, Next Header} to '17' (the UDP protocol number) and sets the {Total, Payload} Length the same as specified in the base IP protocol specifications for ordinary IP packets (see: [RFC0791], [RFC8200] and [I-D.ietf-tsvwg-udp-options]). The underlay source then continues to set the remaining IP header fields as discussed below.

* For raw IP encapsulation, the underlay source sets the IP {Protocol, Next Header} to TBD1 (see: IANA Considerations) and sets the {Total, Payload} Length the same as specified in [RFC0791] or [RFC8200]. The underlay source then continues to set the remaining IP header fields as discussed below.

* For IPsec AH/ESP encapsulation, the underlay source sets the appropriate IP or UDP header to indicate AH/ESP then sets the AH/ESP Next Header field to TBD1 the same as for raw IP encapsulation.

* For direct encapsulations over Ethernet-compatible links, the underlay source prepares an Ethernet Header with EtherType set to TBD2 (see: Section 20.2) (see: Section 7).

* For OAL packet/fragment encapsulations over secured underlay interface connections to the secured spanning tree, the underlay source applies any underlay security encapsulations according to the protocol (e.g., IPsec). These secured carrier packets are then subject to lower layer security services.

When an underlay source includes a UDP header, it SHOULD calculate and include a UDP checksum in carrier packets with full OAL headers to prevent mis-delivery and/or detect IPv4 reassembly corruption; the underlay source MAY set UDP checksum to 0 (disabled) in carrier packets with compressed OAL headers (see: Section 6.5) or when reassembly corruption is not a concern. If the underlay source discovers that a path is dropping carrier packets with UDP checksums disabled, it should supply UDP checksums in future carrier packets sent to the same underlay destination. If the underlay source discovers that a path is dropping carrier packets that do not include a UDP header, it should include a UDP header in future carrier packets.

When an underlay source sends carrier packets with compressed OAL headers and with UDP checksums disabled, mis-delivery due to corruption of the AFVI is possible but unlikely since the corrupted index would somehow have to match valid state in the (sparsely-populated) AERO Flow Information Base (AFIB). In the unlikely event that a match occurs, an OAL intermediate system or destination may receive carrier packets that contain a mis-delivered OAL fragment but can immediately reject any with incorrect Identifications. If the Identification value is somehow accepted, the OAL destination may submit the mis-delivered OAL fragment to the reassembly cache where it will most likely be rejected due to incorrect reassembly parameters. If a reassembly that includes the mis-delivered OAL fragment somehow succeeds (or, for atomic fragments) the OAL destination will verify any included checksums to detect corruption. Finally, any spurious data that somehow eludes all prior checks will be detected and rejected by end-to-end upper layer integrity checks. See: [RFC6935] [RFC6936] for further discussion.

For UDP/IP or raw IP encapsulations, when the underlay source is also the OAL source it next copies the DSCP, ECN and Flow Label values from the OAL header into the underlay IP header. The underlay source then sets the IP TTL/Hop Limit the same as for any host (i.e., it does not copy the Hop Limit value from the OAL header) and finally sets the IP Source and Destination Addresses to direct the carrier packet to the next OAL hop. For carrier packets subject to re-encapsulation, the OAL intermediate system removes the underlay header(s) then prepares to act as the underlay source for the next hop.

The underlay source first decrements the OAL header Hop Limit and discards the OAL packet/fragment if the value reaches 0. Otherwise, the underlay source copies the DSCP value from OAL IPv6 header into the next segment underlay IP header while setting the next segment underlay IP Source and Destination Addresses the same as above. The underlay source then copies the ECN value from the previous segment underlay IP header into both the OAL full/compressed header and the next segment underlay IP header.

The underlay source then prepares to forward the carrier packets to the next OAL intermediate system or destination. For underlay encapsulations over IPv4, if the carrier packet is no larger than 1280 octets the underlay source sets the IPv4 Don’t Fragment (DF) bit to 0 and includes a suitable IPv4 Identification value; otherwise, the underlay source sets DF to 1. This ensures that all IPv4 carrier packets no larger than 1280 octets will be delivered to the underlay destination even if a small amount of fragmentation occurs in the path (see: [RFC3819] for IPv4 link MTU expectations according to their performance characteristics).

For IPv4 carrier packets that set DF to 1 and for all IPv6 carrier packets, delivery is best-effort according to the available path MTU in the spirit of [RFC2473] and [RFC4213]. Since carrier packet transmissions are not within the scope of an explicit tunnel required to pass the IPv6 minimum MTU, however, there is no need for the underlay source to apply source fragmentation since the 1024 octet minimum OFS is operationally assured over all IPv4 and IPv6 paths. The underlay source should therefore ignore any ICMPv6 Packet Too Big or IPv4 Fragmentation Needed messages returned from the network in response to any of its large carrier packet transmissions since the OAL source engages in active probing per [RFC8899].

The underlay source then sends the resulting carrier packets over one or more underlay interfaces. Underlay interfaces often connect directly to physical media on the local platform (e.g., an aircraft with a radio frequency link, a laptop computer with WiFi, etc.), but in some configurations the physical media may be hosted on a separate Local Area Network (LAN) node. In that case, the OMNI interface can establish a Layer-2 VLAN or a point-to-point tunnel (at a layer below the underlay interface) to the node hosting the physical media. The OMNI interface may also apply encapsulation at the underlay interface layer (e.g., as for a tunnel virtual interface) such that carrier packets would appear "double-encapsulated" on the LAN; the node hosting the physical media in turn removes the LAN encapsulation prior to transmission or inserts it following reception. Finally, the underlay interface must monitor the node hosting the physical media (e.g., through periodic keepalives) so that it can convey up-to-date Interface Attribute information to the OMNI interface.

6.3. Reassembly and Decapsulation

For both IPv4 and IPv6, OAL intermediate systems and destinations MUST configure a minimum EMTU_R of 1500 octets on all unsecured underlay interfaces. (Secured underlay interfaces instead use an EMTU_R specific to the underlay security service such as IPsec.) OAL intermediate systems and destinations are permitted to configure a larger underlay interface EMTU_R in order to pass still larger carrier packets.

OAL destinations MUST configure an adaptation layer EMTU_R of 65535 octets to support reassembly of fragmented OAL packets of all sizes. OAL nodes must further recognize and honor the extended Identifications included in the IPv6 Extended Fragment Header [I-D.templin-6man-ipid-ext2].

When an OMNI interface processes a carrier packet received on an underlay interface, it copies the ECN value from the underlay IP encapsulation header into the OAL header but does not copy the DSCP value from the underlay IP header into the OAL header according to the differentiated services pipe model for tunnels [RFC2983]. The OMNI interface next discards the underlay headers and examines the OAL header of the enclosed OAL packet/fragment according to the value in the Type field as discussed in Section 6.2

If the OAL packet/fragment is addressed to a different node, the OMNI interface (acting as an OAL intermediate system) decrements the OAL Hop Limit as discussed in Section 6.2 then performs underlay encapsulation and forwards the resulting carrier packet. If the OAL packet/fragment is addressed to itself, the OMNI interface (acting as an OAL destination) accepts or drops based on the (Source, Destination, Flow Label, Identification)-tuple.

The OAL destination next drops all ordinal OAL non-first fragments that would overlap or leave "holes" with respect to other ordinal fragments already received. The OAL destination updates a checklist of accepted ordinal fragments of the same OAL packet but admits all accepted fragments into the reassembly cache.

The OAL destination then reassembles the original OAL packet after all fragments have arrived. The reassembled OAL packet may exceed 65535 by as much as the size of the OAL encapsulation extension headers. The OAL destination does not write this (too-large) value into the OAL header Payload Length field, but instead retains the value during reassembly. When reassembly is complete, the OAL destination finally replaces the OAL IPv6 encapsulation header with a virtual Ethernet header. The OAL destination's OMNI interface then delivers the original IP packet to the network layer. The original IP packet may therefore be as large as 65535 octets.

When an OAL path traverses an IPv6 network with routers that perform adaptation layer forwarding based on full IPv6 headers with OAL addresses, the OAL intermediate system at the head of the IPv6 path forwards the OAL packet/fragment the same as an ordinary IPv6 packet without decapsulating and delivering to the network layer. Once within the IPv6 network, these OAL packets/fragments may traverse arbitrarily-many IPv6 hops before arriving at an OAL intermediate system which may again encapsulate the OAL packets/fragments as carrier packets for transmission over underlay interfaces.

Note: carrier packets often traverse paths with underlying links that use integrity checks such as CRC-32 which provide adequate hop-by-hop integrity assurance for payloads up to ~9K octets [CRC]. However, other paths may traverse links (such as fragmenting tunnels over IPv4 - see: [RFC4963]) that do not include adequate checks.

6.4. OMNI-Encoded IPv6 Extension Headers

The IPv6 specification [RFC8200] defines extension headers that follow the base IPv6 header, while Upper Layer Protocols (ULPs) are specified in other documents. Each extension header present is identified by a "Next Header" octet in the previous (extension) header and encodes a "Next Header" field in the first octet that identifies the next extension header or ULP instance. The OMNI specification supports encoding of IPv6 extension header chains immediately following the underlay UDP, IP or Ethernet header even if the underlay IP protocol version is IPv4.

The OAL source prepares an OMNI extension header chain by setting the first 4 bits of the first IPv6 extension header in the chain to a Type value for the extension header itself immediately following the underlay protocol header. The source then sets the next 4 bits to a Next value that identifies either a terminating ULP or the next extension header in the chain. The source then sets the first 8 bits of each subsequent IPv6 extension header in the chain to the standard Next Header encoding as shown in Figure 5:

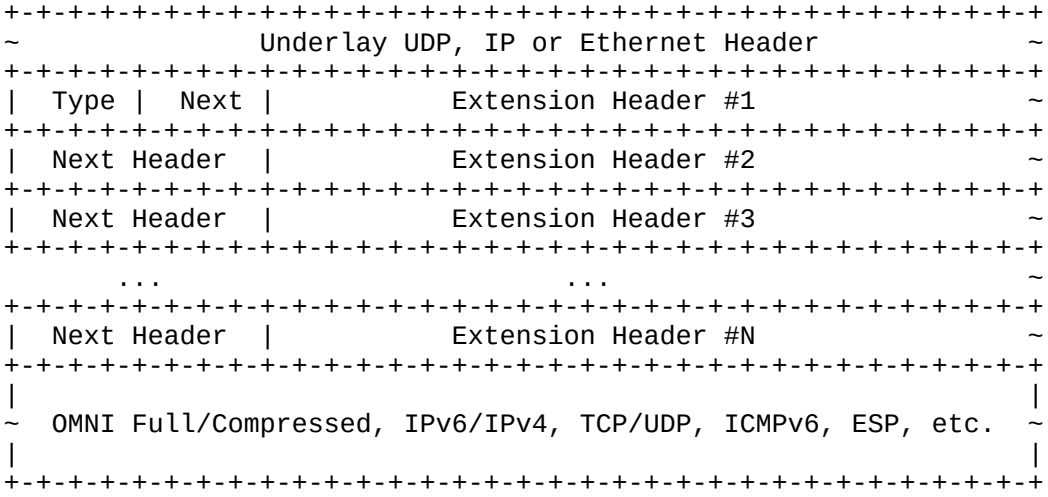

Figure 5: OMNI Extension Header Chains

The following Type/Next values are currently defined:

0 (OMNI-RES1) - Reserved for experimentation.

1 (OMNI-OCH1) - OMNI Compressed Header, Type 1 per Section 6.5.

2 (OMNI-OCH2) - OMNI Compressed Header, Type 2 per Section 6.5.

3 (OMNI-RES2) - Reserved for experimentation.

4 (OMNI-IP4) - IPv4 header per [RFC0791].

5 (OMNI-HBH) - Hop-by-Hop Options per Section 4.3 of [RFC8200].

6 (OMNI-IP6) - IPv6 header per [RFC8200].

7 (OMNI-RH) - Routing Header per Section 4.4 of [RFC8200].

8 (OMNI-FH) - Fragment Header per Section 4.5 of [RFC8200].

9 (OMNI-DO) - Destination Options per Section 4.6 of [RFC8200].

10 (OMNI-AH) - Authentication Header per [RFC4302].

11 (OMNI-ESP) - Encapsulating Security Payload per [RFC4303].

12 (OMNI-NNH) - No Next Header per Section 4.7 of [RFC8200].

13 (OMNI-TCP) - TCP Header per [RFC9293].

14 (OMNI-UDP) - UDP Header per [RFC0768].

15 (OMNI-ULP) - Upper Layer Protocol shim (see below).

Entries OMNI-OCH1 through OMNI-AH in the above list follow the convention that the OMNI Type/Version appears in the first 4 bits of the extension header (or IP header) itself. Conversely, entries OMNI-ESP through OMNI-UDP represent commonly-used ULPs which do not encode a Type/Version in the first 4 bits.

Entries OMNI-HBH, OMNI-RH, OMNI-FH, OMNI-DO and OMNI-AH represent true IPv6 extension headers encoded for OMNI, which may be chained. Source and destination processing of OMNI extension headers follows exactly per their definitions in the normative references, with the exception of the special (Type, Next) coding in the first 8 bits of the first extension header.

When a ULP not found in the above table immediately follows the underlay UDP, IP or Ethernet header, the source includes a 2 octet "Type 1 ULP Shim" before the ULP where both the first 4-bit (Type) and next 4-bit (Next) fields encode the special value 15 (OMNI-ULP). The source then includes a Next Header field that encodes the IP protocol number of the ULP. The source then includes the ULP data immediately after the shim as shown in Figure 6.

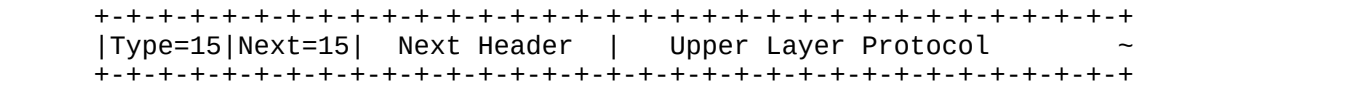

Figure 6: OMNI Upper Layer Protocol (ULP) Shim (Type 1)

When a ULP "OMNI-(N)" found in the above table immediately follows the underlay UDP, IP or Ethernet header, the source includes a 1 octet "Type 2 ULP Shim" before the ULP where the first 4 bits encode the special Type value 15 (OMNI-ULP) and the next 4 bits encode the Next ULP type "N" taken from the table above. The source then includes the ULP data immediately after the shim as shown in Figure 7.

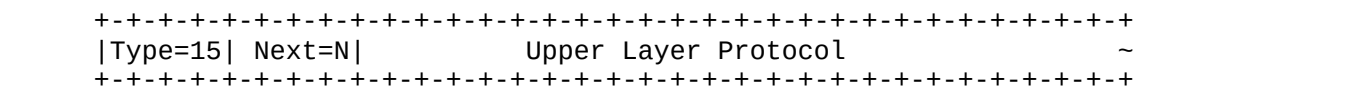

Figure 7: OMNI Upper Layer Protocol (ULP) Shim (Type 2)

When a ULP not found in the above table follows a first OMNI extension header, the source sets the extension header Next field to OMNI-ULP (15) and includes a 1 octet "Type 3 ULP Shim" that encodes the IP protocol number for the Next Header of the ULP data that follows as shown in Figure 8.

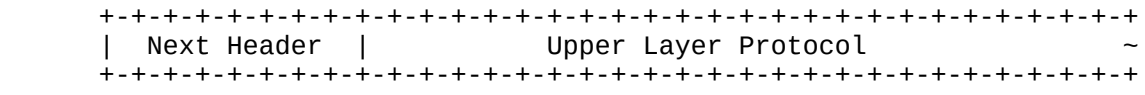

Figure 8: OMNI Upper Layer Protocol (ULP) Shim (Type 3)

When a ULP "OMNI-(N)" found in the above table follows a first OMNI extension header, the source sets the extension header Next field to the ULP Type "N" and does not include a shim. The ULP then begins immediately after the first OMNI extension header.

When a ULP of any kind follows a non-first OMNI extension header, the source sets the extension header Next Header field to the IP protocol number for the ULP and does not include a shim. The ULP then begins immediately after the non-first OMNI extension header.

Note: The underlay UDP header (when present) is logically considered as the first extension header in the chain. If an Advanced Jumbo extension header is also present, its Jumbo Payload length includes the length of the UDP header.

Note: After a node parses the extension header chain, it changes the "Type/Next" field in the first extension header back to the correct "Next Header" value before processing the first extension header.

6.5. OMNI Full and Compressed Headers

OAL sources that send OAL packets with full OMNI IPv6 Headers include a Segment Routing Header (SRH) as an extension per [RFC8754]. The Segment List elements include the adaptation layer addresses of the Client itself and of any OAL intermediate systems on the path. Clients discover their local Segment List elements in their RS/RA exchanges with FHS Proxy/Servers, where each partition border OAL intermediate system in the RS message forwarding path records its address before forwarding to the next partition border OAL intermediate system. See: Section 13 for further discussion.

The SRH is followed by an IPv6 Extended Fragment Header to support segment-by-segment forwarding based on an AERO Flow Information Base (AFIB) in each OAL node in the path. OAL sources, intermediate systems and destinations establish AFIB header compression state based on secured "pilot" control messages. OAL nodes should apply

OMNI Header Compression for subsequent data plane messages to significantly reduce header overhead and suppress advisory ICMPv6 Parameter Problem messages (see: [I-D.templin-6man-aero3]).

OAL sources apply header compression in order to avoid transmission of redundant data found in the original IP packet and OAL encapsulation headers; the resulting compressed headers are often significantly smaller than the original IP packet header itself even when OAL encapsulation is applied. Header compression is limited to the OAL IPv6 encapsulation header plus extensions along with the base original IP packet header; it does not extend to include any extension headers of the original IP packet which appear as upper layer payload immediately following the compressed headers.

Each OAL node establishes AFIB soft state entries known as AERO Flow Vectors (AFVs) which support both OAL packet/fragment forwarding and OAL/IPv6 header compression/decompression. OAL nodes locate each AFV by an AERO Flow Vector Index (AFVI) which in conjunction with the previous hop underlay address provides compression/decompression and next hop forwarding context.

When an OAL source sends carrier packets that contain OAL packets/ fragments to a next hop, it includes a full IPv6 header with an SRH containing segment addressing information followed by an Extended Fragment Header. The first 4 bits following the underlay headers must encode the Type OMNI-IP6 to signify that an uncompressed IPv6 header (plus any extensions) is present.

When AFV state is available, the OAL source should omit significant portions of the OAL header (plus extensions) and original IP packet header by applying OMNI header compression. For OAL first fragments (including atomic fragments), the OAL source uses OMNI Compressed Header, Type 1 (OCH1) Format (a) as shown in Figure 9:

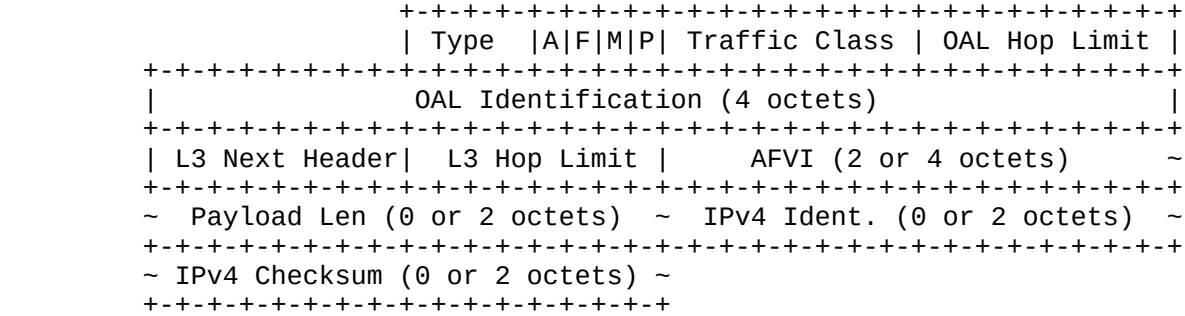

Figure 9: OMNI Compressed Header (OCH1) Format (a)

The format begins with a 4-bit Type followed by 4 flag bits followed by an 1 octet Traffic Class (copied into the OAL header from the original IP packet header) followed by an 1 octet OAL Hop Limit. The OAL source sets Type to OMNI-OCH1, sets Hop Limit to the uncompressed OAL header Hop Limit and sets the ECN bits in the Traffic Class field the same as for an uncompressed IP header. The OAL source next sets (F)ormat to 0 then sets (M)ore Fragments the same as for an uncompressed Extended Fragment Header.

The header next includes the 4 least significant octets of the OAL Identification followed by the Next Header (or Protocol) and Hop Limit (or TTL) values from the original (L3) IP packet header. These values are followed by 2 octet AFVI if the (A) flag is set to 0; otherwise a 4 octet AFVI. The format then includes a 2 octet Payload Length only if the underlay headers do not include a length field. The format finally includes 2 octet Identification and Header Checksum values for IPv4 original packets only. (Compression therefore applies to the original IP packet header plus the OAL IPv6 header along with its SRH and Extended Fragment Header in a unified concatenation.)

The OAL source finally includes the payload of the OAL first fragment (i.e., beginning after the original IP header) immediately following the OCH1 header, and the underlay header length field (if present) is reduced by the difference in length between the compressed and full-length headers. The OAL destination can determine the payload length by examining the underlay header length field if present; otherwise, the OCH1 header itself includes a 2 octet Payload Length field that encodes the length of the packet payload that follows the OCH1. (Note that OAL first fragments and atomic packets are logically considered ordinal fragment 0 even though the format does not include an Index field.)

When the OAL source needs to probe the OAL Fragment Size (OFS) for a given flow, it sets the (P)robe flag and includes a probe message of the desired size following the OCH1 header. Upon receipt, the OAL destination returns a secured control message reply to the OAL source. When the OAL source receives the control message, it can either maintain its current OFS for this flow or advance to a larger OFS according to the probe size.

For OAL non-first fragments (i.e., those with non-zero Index), the OAL uses OMNI Compressed Header, Type 1 (OCH1) Format (b) as shown in Figure 10:

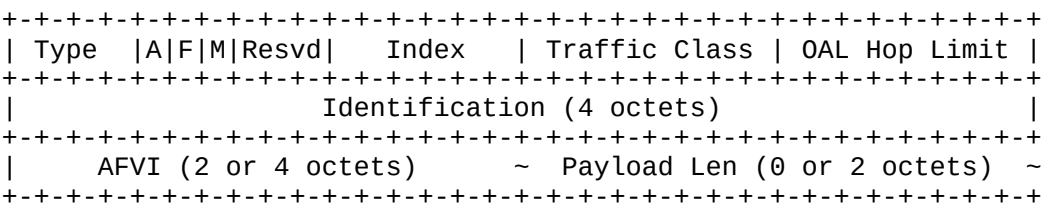

Figure 10: OMNI Compressed Header (OCH1) Format (b)

The format begins with a 4-bit Type followed by 3 flags followed by a 3-bit Reserved field (set to 0) followed by a 6-bit ordinal fragment Index. All other fields up to and including the Payload Length (if present) are included the same as for an OCH1 first fragment.

The OAL source sets Type to OMNI-OCH1, sets Hop Limit to the uncompressed OAL header Hop Limit value, sets (Index, (A)FVI, (M)ore Fragments, Identification) to their appropriate values as a non-first fragment and sets (F)ormat to 1. The OAL source also sets Index to a monotonically increasing ordinal value beginning with 1 for the first non-first fragment, 2 for the second non-first fragment, 3 for the third non-first fragment, etc., up to at most 63 for the final fragment.

The OAL source then includes a non-first fragment body immediately following the OCH1 header, and reduces the underlay header length field (if present) by the difference in length between the compressed headers and full-length original IP header with OAL IPv6 header plus extensions. The OAL destination will then be able to determine the Payload Length by examining the underlay header length field if present; otherwise by examining the 2 octet OCH1 Payload Length the same as for first fragments.

The OCH1 Format (a) is used for all original IPv6 packets that do not include a Fragment Header as well as for original IPv4 packets that set IHL to 5, DF to 1 and (MF; Fragment Offset) to 0 (the OCH1 Format (b) is used for non-first fragments in all IP protocol cases).

For other "non-atomic" original IP packets and first fragments, the OAL source uses the "Type 2" OMNI Compressed Header (OCH2) formats shown in Figure 11 and Figure 12:

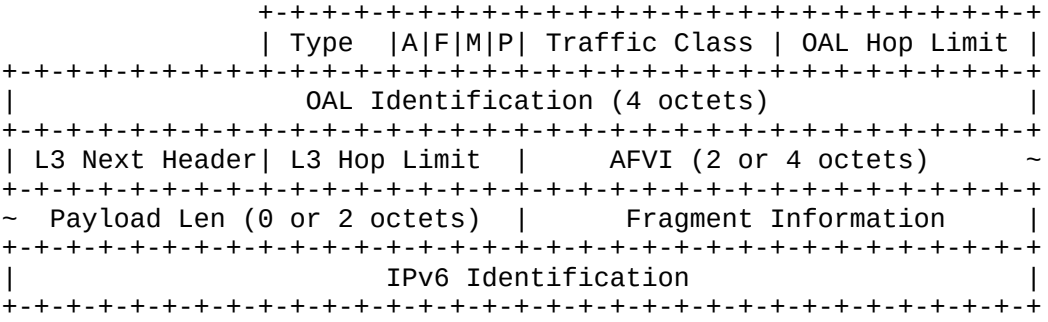

Figure 11: OMNI Compressed Header, Type 2 (OCH2) Format (a)

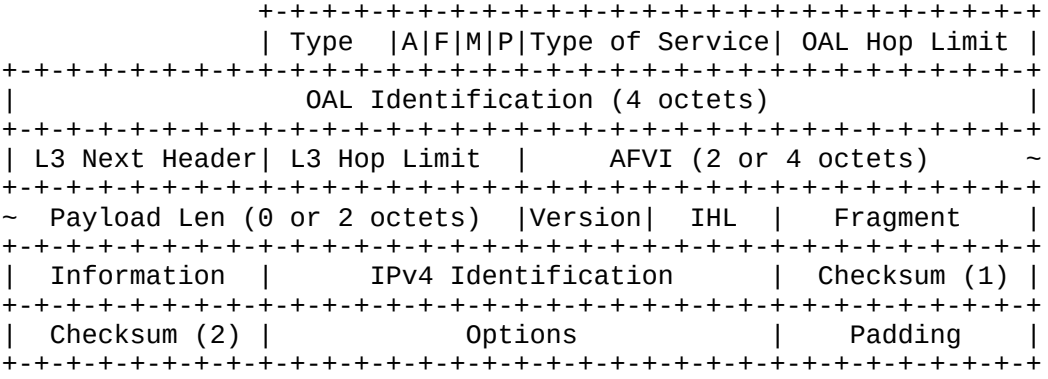

Figure 12: OMNI Compressed Header, Type 2 (OCH2) Format (b)

In both of the above OCH2 formats, the leading octets include the same information that would appear in a corresponding OCH1 format (a) header. The (F) flag is set to 0 for OCH2 format (a) or 1 for OCH2 format (b), while all other flags are processed the same as for OCH1 format (a).

The remainder of the OCH2 format (a) includes fields that would appear in an uncompressed IPv6 header per [RFC8200] plus Fragmentation Information. For the standard IPv6 Fragment Header, Fragment Information consists of the 13-bit Fragment Offset followed by the 3 IPv6 Fragment Header flag bits. For the EFH, Fragmentation Information consists of the NH-Cache followed by the 2 EFH flag bits followed by the 6-bit Index.

The remainder of OCH2 format (b) includes fields that would appear in an uncompressed IPv4 header per [RFC0791] with the Options and Padding lengths calculated based on IHL. In both cases, the Source and Destination Addresses are not transmitted.

When an OAL destination or intermediate system receives a carrier packet, it determines the length of the encapsulated OAL information and verifies that the innermost underlay next header field indicates OMNI (see: Section 6.2), then processes any included OMNI underlay extension headers as specified in Section 6.4. The OAL destination then examines the Next Header field of the final underlay extension header. If the Next Header field contains the value TBD1, and the 4-bit Type that follows encodes a value OMNI-IP6, OMNI-OCH1 or OMNI-OCH2 the OAL node processes the remainder of the OAL header as a full or compressed header as specified above.

When an OAL node forwards an OAL packet, it determines the AFVI for the next OAL hop by using the AFVI included in the OCH to search for a matching AFV. The OAL node then writes the next hop AFVI into the OCH (while adjusting the AFVI length if necessary) and forwards the OAL packet to the next hop. This same AFVI re-writing progression begins with the OAL source then continues over all OAL intermediate systems in the path and finally ends at the OAL destination.

When the OAL destination receives the OAL packet, it reconstructs the OAL and original IP headers based on the information cached in the AFV combined with the received information in the OCH1/2. For non-atomic fragments, the OAL destination then adds the resulting OAL fragment to the reassembly cache if the Identification is acceptable. Following OAL reassembly if necessary, the OAL destination delivers the original IP packet to the network layer.

For all OCH1/2 types, the OAL source sets all Reserved fields and bits to 0 on transmission and the destination node ignores the values on reception. For both OCH1/2, ECN information is compiled for first fragments, and not for non-first fragments.

Finally, if an IPv6 Hop-by-Hop (HBH) and/or Routing Header extension header is required to appear as per-fragment extensions with each OAL fragment that uses OCH1 format (b) or OCH2 compression the OAL source inserts an OMNI-HBH and/or OMNI-RH header as the first extension(s) following the underlay header and before the OMNI-OCH1/2 as discussed in Section 6.4.

6.6. UDP/IP Encapsulation Avoidance

When an OAL node is unable to determine whether the next OAL hop is connected to the same underlay link, it should perform carrier packet underlay encapsulation for initial packets sent via the next hop over a specific underlay interface by including full UDP/IP headers and with the UDP port numbers set as discussed in Section 6.2. The node can thereafter attempt to send an IPv6 ND solicitation message to the next OAL hop in carrier packet(s) that omit the UDP header and set

the IP protocol number to TBD1. If the OAL node receives an IPv6 ND advertisement, it can omit the UDP header in subsequent packets. The node can further attempt to send an IPv6 ND solicitation in carrier packet(s) that omit both the UDP and IP headers and set EtherType to TBD2. If the source receives an IPv6 ND advertisement, it can begin omitting both the UDP and IP headers in subsequent packets.

Note: in the above, "next OAL hop" refers to the first OAL node encountered on the optimized path to the destination over a specific underlay interface as determined through route optimization (e.g., see: [I-D.templin-6man-aero3]). The next OAL hop could be a Proxy/ Server, Gateway or the OAL destination itself.

6.7. OAL Identification Window Maintenance

The OAL encapsulates each original IP packet as an OAL packet then performs fragmentation to produce one or more carrier packets with the same 8 octet Identification value. In environments where spoofing is not considered a threat, OMNI interfaces send OAL packets with Identifications beginning with an unpredictable Initial Send Sequence (ISS) value [RFC7739] monotonically incremented (modulo 2**64) for each successive OAL packet sent to either a specific neighbor or to any neighbor. (The OMNI interface may later change to a new unpredictable ISS value as long as the Identifications are assured unique within a timeframe that would prevent the fragments of a first OAL packet from becoming associated with the reassembly of a second OAL packet.) In other environments, OMNI interfaces should maintain explicit per-flow send and receive windows to detect and exclude spurious carrier packets that might clutter the reassembly cache as discussed below.

OMNI interface neighbors use a window synchronization service similar to TCP [RFC9293] to maintain unpredictable ISS values incremented (modulo 2**64) for each successive OAL packet and re-negotiate windows often enough to maintain an unpredictable profile. OMNI interface neighbors exchange control messages that include OMNI Neighbor Synchronization sub-options that include TCP-like information fields and flags to manage streams of OAL packets instead of streams of octets. As a link layer service, the OAL provides low-persistence best-effort retransmission with no mitigations for duplication, reordering or deterministic delivery. Since the service model is best-effort and only control message sequence numbers are acknowledged, OAL nodes can select unpredictable new initial sequence numbers outside of the current window without delaying for the Maximum Segment Lifetime (MSL).

OMNI interface end systems and intermediate systems maintain current and previous per-flow window state in IPv6 ND NCEs and/or AFVs to support dynamic rollover to a new window while still sending OAL packets and accepting carrier packets from the previous windows. OMNI interface neighbors synchronize windows through asymmetric and/or symmetric control message exchanges. When OMNI end and intermediate systems receive a control message with new per-flow window information, it resets the previous window state based on the current window then resets the current window based on new and/or pending information.

The control message OMNI option Neighbor Synchronization sub-option includes TCP-like information fields including Sequence Number, Acknowledgement Number, Scale, Window and flags (see: Section 10). Window Scaling is applied the same as specified in [RFC7323]. OMNI interface neighbors and intermediate systems maintain the following TCP-like state variables on a per-interface-pair basis (i.e., through a combination of NCE and/or AFV state):

Send Sequence Variables (current, previous and pending)

```
SND.NXT - send next
SND.WND - send window
ISS     - initial send sequence number
```

Receive Sequence Variables (current and previous)

```
RCV.NXT - receive next
RCV.WND - receive window
IRS     - initial receive sequence number
```

OMNI interface neighbors "OAL A" and "OAL B" exchange control messages per [RFC4861] with OMNI options that include TCP-like information fields in a Neighbor Synchronization. When OAL A synchronizes with OAL B, it maintains both a current and previous SND.WND beginning with a new unpredictable ISS and monotonically increments SND.NXT for each successive OAL packet transmission. OAL A initiates synchronization by including the new ISS in the Sequence Number of an authentic control message with the SYN flag set and with Scale set to S (up to 14) and Window set to W (up to 2**16) while creating a NCE in the INCOMPLETE state if necessary. OAL A caches the new ISS as pending, uses the new ISS as the Identification for OAL encapsulation, then sends the resulting OAL packet to OAL B and waits up to RetransTimer milliseconds to receive a control message response with the ACK flag set (retransmitting up to MAX_UNICAST_SOLICIT times if necessary).

When OAL B receives the SYN, it creates a NCE in the STALE state and also an AFV if necessary, resets its RCV variables and caches the source's send window size as its receive window size. OAL B then prepares a control message with the ACK flag set, with the Acknowledgement Number set to OAL A's next sequence number, and with Scale set to S and Window set to W. Since OAL B does not assert an ISS of its own, it uses the IRS it has cached for OAL A as the Identification for OAL encapsulation then sends the ACK to OAL A.

When OAL A receives the ACK, it notes that the Identification in the OAL header matches its pending ISS. OAL A then sets the NCE state to REACHABLE and resets its SND variables based on the Scale, Window and Acknowledgement Number (which must include the sequence number following the pending ISS). OAL A can then begin sending OAL packets to OAL B with Identification values within the (new) current SND.WND for this interface pair for up to ReachableTime milliseconds or until the NCE is updated by a new control message exchange. This implies that OAL A must send a new SYN before sending more than N OAL packets within the current SND.WND, i.e., even if ReachableTime is not nearing expiration. After OAL B returns the ACK, it accepts carrier packets received from OAL A via this interface pair within either the current or previous RCV.WND as well as any new authentic control messages with the SYN flag set received from OAL A even if outside the windows.

OMNI interface neighbors can employ asymmetric window synchronization as described above using 2 independent (SYN -> ACK) exchanges (i.e., a 4-message exchange), or they can employ symmetric window synchronization using a modified version of the TCP "3-way handshake" as follows:

* OAL A prepares a SYN with an unpredictable ISS not within the current SND.WND and with Scale set to S and Window set to W. OAL A caches the new ISS and window size as pending information, uses the pending ISS as the Identification for OAL encapsulation, then sends the resulting OAL packet to OAL B and waits up to RetransTimer milliseconds to receive an ACK response (retransmitting up to MAX_UNICAST_SOLICIT times if necessary).

* OAL B receives the SYN, then resets its RCV variables based on the Sequence Number while caching OAL A's send window size as its receive window size. OAL B then selects a new unpredictable ISS outside of its current window, then prepares a response with Sequence Number set to the pending ISS and Acknowledgement Number set to OAL A's next sequence number. OAL B then sets both the SYN and ACK flags, sets Scale and Window to chosen values S' and W' and sets the OPT flag according to whether an explicit concluding ACK is optional or mandatory. OAL B then uses the pending ISS as

the Identification for OAL encapsulation, sends the resulting OAL packet to OAL A and waits up to RetransTimer milliseconds to receive an acknowledgement (retransmitting up to MAX_UNICAST_SOLICIT times if necessary).

* OAL A receives the SYN/ACK, then resets its SND variables based on the Acknowledgement Number (which must include the sequence number following the pending ISS). OAL A then resets its RCV variables based on the Sequence Number and OAL B's advertised send window S'/W' and marks the NCE as REACHABLE. If the OPT flag is clear, OAL A next prepares an immediate unsolicited control message with the ACK flag set, the Acknowledgement Number set to OAL B's next sequence number, with Scale set to S' and Window set W', and with the OAL encapsulation Identification to SND.NXT, then sends the resulting OAL packet to OAL B. If the OPT flag is set and OAL A has OAL packets queued to send to OAL B, it can optionally begin sending their carrier packets under the current SND.WND as implicit acknowledgements instead of returning an explicit ACK.

* OAL B receives the implicit/explicit acknowledgement(s) then resets its SND state based on the pending/advertised values and marks the NCE as REACHABLE. Note that OAL B sets the OPT flag in the SYN/ACK to assert that it will interpret timely receipt of carrier packets within the (new) current window as an implicit acknowledgement. Potential benefits include reduced delays and control message overhead, but use case analysis is outside the scope of this specification.)

Following synchronization, OAL A and OAL B hold updated NCEs and AFVs, and can exchange OAL packets with Identifications set to SND.NXT for each flow while the state remains REACHABLE and there is available window capacity. (Intermediate systems that establish AFVs for the per-flow window synchronization exchanges can also use the Identification window for source validation.) Either neighbor may at any time send a new SYN to assert a new ISS. For example, if OAL A's current SND.WND for OAL B is nearing exhaustion and/or ReachableTime is nearing expiration, OAL A can continue sending OAL packets under the current SND.WND while also sending a SYN with a new unpredictable ISS. When OAL B receives the SYN, it resets its RCV variables and may optionally return either an asymmetric ACK or a symmetric SYN/ACK to also assert a new ISS. While sending SYNs, both neighbors continue to send OAL packets with Identifications set to the current SND.NXT for each interface pair then reset the SND variables after an acknowledgement is received.

While the optimal symmetric exchange is efficient, anomalous conditions such as receipt of old duplicate SYNs can cause confusion for the algorithm as discussed in Section 3.5 of [RFC9293]. For this

reason, the OMNI Neighbor Synchronization sub-option includes an RST flag which OAL nodes set in solicited control message responses to ACKs received with incorrect acknowledgement numbers. The RST procedures (and subsequent synchronization recovery) are conducted exactly as specified in [RFC9293].

OMNI interfaces that employ the window synchronization procedures described above observe the following requirements:

* OMNI interfaces MUST select new unpredictable ISS values that are at least a full window outside of the current SND.WND.

* OMNI interfaces MUST set the Scale and Window fields in SYN messages as a non-negotiable advertised send window size.

* OMNI interfaces MUST send control messages used for window synchronization securely while using unpredictable initial Identification values until synchronization is complete.

It is essential to understand that the above window synchronization operations between nodes OAL(A) and OAL(B) are conducted in control message exchanges over multihop paths with potentially many OAL(i) intermediate hops in the forward and reverse paths (which may be disjoint). Each such forward path OAL(i) caches the Sequence Number, Scale and Window values advertised from OAL(A) to OAL(B) in its AFV entry indexed by the previous hop underlay address and AFVI, while each such reverse path OAL(i) caches the Sequence Number, Scale, Window and AFVI advertised from OAL(B) to OAL(A). (The forward/ reverse path OAL(i) nodes then select new unique next-hop AFVIs before forwarding.)

While multiple independent paths may exist between nodes OAL(A) and OAL(B), the synchronized Sequence Numbers between the two nodes apply collectively to all paths. Nodes OAL(A) and OAL(B) therefore perform initial synchronization through control message exchanges with the SYN flag set over a first path for which intermediate nodes cache the Sequence Number, Scale and Window values in their AFVs. However, control message exchanges that establish and maintain alternate paths include the current Sequence Number and residual window size but with the SYN flag clear.

Each neighbor pair can therefore dynamically coordinate multiple independent paths from a single Sequence Number space in this way. When nodes OAL(A) and OAL(B) need to re-synchronize they again advertise new Sequence Number, Scale and Window size values with the SYN flag set. The nodes must then exchange additional control messages using the new values and with the SYN flag clear to establish or maintain alternate paths.

Note: Although OMNI interfaces employ TCP-like window synchronization and support ACK responses to SYNs, all other aspects of the IPv6 ND protocol (e.g., control message exchanges, NCE state management, timers, retransmission limits, etc.) are honored exactly per [RFC4861]. OMNI interfaces further manage per-interface-pair window synchronization parameters in one or more AFVs for each neighbor pair.

Note: Recipients of OAL-encapsulated control messages index the NCE based on the message Source Address, which also determines the carrier packet Identification window. However, control messages may contain a message Source Address that does not match the OMNI encapsulation Source Address when the recipient acts as a proxy.

Note: OMNI interface neighbors apply separate send and receive windows for all of their (multilink) underlay interface pairs that exchange carrier packets. Each interface pair represents a distinct underlay network path, and the set of paths traversed may be highly diverse when multiple interface pairs are used. OMNI intermediate systems therefore become aware of each distinct set of interface pair window synchronization parameters based on periodic control message updates to their respective AFVs.

6.8. OAL Fragmentation Reports and Retransmissions

When the OAL destination experiences reassembly congestion for a specific flow (e.g., when excessive numbers of reassembly failures are occurring), it can send an OAL Fragmentation Report (FRAGREP) message to the OAL source to recommend a reduced Maximum Receive Unit (MRU) for the flow (see: Section 10.2.15). When the OAL source receives the FRAGREP, it caches the new MRU for the flow and returns "soft errors" to original sources that send larger packets (see: Section 6.9). When the OAL destination experiences reassembly congestion for all flows from the same OAL source, it can return FRAGREP messages with Flow Label set to 0 as indication that all flows are affected.

When the round-trip delay from the original source to the final destination is long while the round-trip time from the OAL source to the OAL destination is significantly shorter, the OAL source can maintain a short-term cache of the OAL fragments it sends to OAL destinations for each flow in case timely best-effort selective retransmission is requested. The OAL destination in turn maintains a checklist for (Source, Destination, Flow Label, Identification)-tuples of recently received OAL fragments and notes the ordinal numbers of OAL fragments already received (i.e., as ordinals #0, #1, #2, #3, etc.). The timeframe for maintaining the OAL source and destination caches determines the link persistence (see: [RFC3366]).

If the OAL destination notices some fragments missing after most other fragments within the same link persistence timeframe have already arrived, it may issue an Automatic Repeat Request (ARQ) with Selective Repeat (SR) by sending an unsolicited control message to the OAL source. The OAL destination creates a control message with an OMNI option with one or more FRAGREP sub-options that include Bitmaps for fragments received and missing from this OAL source (see: Section 10.2.15). The OAL destination includes an authentication signature if necessary, performs OAL encapsulation (with its own address as the OAL Source Address and the Source Address of the message that prompted the unsolicited control message as the OAL Destination Address) and sends the message to the OAL source.

If an OAL intermediate system or OAL destination processes an OAL fragment for which corruption is detected, it may similarly issue an immediate ARQ/SR the same as described above. The FRAGREP provides an immediate (rather than time-bounded) indication to the OAL source that a fragment has been lost.

When the OAL source receives the control message, it authenticates the message then examines any enclosed FRAGREPs. For each (Source, Destination, Flow Label, Identification)-tuple, the OAL source determines whether it still holds the corresponding OAL fragments in its cache and retransmits any for which the Bitmap indicates a loss event. For example, if the Bitmap indicates that ordinal fragments #3, #7, #10 and #13 from the OAL packet with Identification 0x0123456789abcdef are missing the OAL source only retransmits those fragments. When the OAL destination receives the retransmitted OAL fragments, it admits them into the reassembly cache and updates its checklist. If some fragments are still missing, the OAL destination may send a small number of additional ARQ/SR control messages within the link persistence timeframe.

The OAL therefore provides a link layer low-to-medium persistence ARQ/SR service consistent with [RFC3366] and Section 8.1 of [RFC3819]. The service provides the benefit of timely best-effort link layer retransmissions which may reduce OAL fragment loss and avoid some unnecessary end-to-end delays. This best-effort network-based service therefore complements transport and higher layer end-to-end protocols responsible for true reliability.

6.9. OMNI Interface MTU Feedback Messaging

When the OMNI interface forwards original IP packets from the network layer, it invokes the OAL and returns internally-generated Path MTU Discovery (PMTUD) ICMPv4 "Fragmentation Needed and Don’t Fragment Set" [RFC1191] or ICMPv6 "Packet Too Big (PTB)" [RFC8201] messages as necessary. This document refers to both message types as "PTBs" and introduces a distinction between PTB "hard" and "soft" errors as discussed below.

Ordinary PTB messages are hard errors that always indicate loss due to a real MTU restriction has occurred. However, the OMNI interface can also forward original IP packets via OAL encapsulation and fragmentation while at the same time returning PTB soft error messages (subject to rate limiting) to the original source to suggest smaller sizes due to factors such as link performance characteristics, excessive numbers of fragments needed, reassembly congestion, etc.

This ensures that the path MTU is adaptive and reflects the current path used for a given data flow. The OMNI interface can therefore continuously forward original IP packets without loss while returning PTB soft error messages that recommend smaller sizes. Original sources that receive the soft errors in turn reduce the size of the original IP packets they send the same as for hard errors, but not necessarily due to a loss event. The original source can then resume sending larger packets if the soft errors subside.

OAL intermediate systems that experience fragment loss and OAL end systems that experience reassembly cache congestion can return unsolicited control messages that include OMNI encapsulated PTB soft error messages to OAL sources that originate fragments (subject to rate limiting). The OAL node creates a secured ICMPv6 PTB control message with MTU set to a reduced value and with the leading portion an OAL first fragment containing the header of an original IP packet for which the source must be notified (see: Section 10).

The OAL node that sends the control message encapsulates the leading portion of the OAL first fragment (beginning with the OAL header) in the PTB "packet in error" field and signs the message if an authentication signature is necessary. The OAL node then performs OAL encapsulation (with its own address as the Source Address and the Source Address of the message that prompted the control message response as the Destination Address) and sends the message to the OAL source. (Note that OAL intermediate systems forward control messages via the secured spanning tree while OAL source and destination end systems include an authentication signature when necessary.)

The OAL source prepares the PTB soft error by first setting the Type field to 2 for IPv6 [RFC4443] or "Packet Too Big" for IPv4 (see: [I-D.templin-6man-ipid-ext2]). The OAL source then sets the Code field to "PTB Soft Error (no loss)" if the OAL destination forwarded the original IP packet successfully or "PTB Soft Error (loss)" if it was dropped (see: [I-D.templin-6man-ipid-ext2]). The OAL source next sets the PTB Destination Address to the original IP packet Source Address, and sets the PTB Source Address to one of its OMNI interface addresses that is reachable from the perspective of the original source.

The OAL source then sets the MTU field to a value smaller than the original IP packet size but no smaller than 1280, writes as much of the original IP packet first fragment as possible into the "packet in error" field such that the entire PTB including the IP header is no larger than 1280 octets for IPv6 or 576 octets for IPv4. The OAL source then calculates and sets the ICMP Checksum and returns the PTB to the original source.

An original sources that receives these PTB soft errors first verifies that the ICMP Checksum is correct and the packet-in-error contains the leading portion of one of its recent packet transmissions. The original source can then adaptively tune the size of the original IP packets it sends to produce the best possible throughput and latency, with the understanding that these parameters may fluctuate over time due to factors such as congestion, mobility, network path changes, etc. Original sources should therefore consider receipt or absence of soft errors as hints of when decreasing or increasing packet sizes may provide better performance.

The OMNI interface supports continuous transmission and reception of packets of various sizes in the face of dynamically changing network conditions. Moreover, since PTB soft errors do not indicate a hard limit, original sources that receive soft errors can resume sending larger packets without waiting for the recommended 10 minutes specified for PTB hard errors [RFC1191][RFC8201]. The OMNI interface therefore provides an adaptive service that accommodates MTU diversity especially well-suited for air/land/sea/space mobile Internetworking.

Note: when the OAL source receives persistent Fragmentation Reports for a given flow (see: Section 6.8), it should return PTB soft errors to the original source (subject to rate limiting) the same as if it had received PTB soft errors from the OAL destination. When the original source is likely to retransmit an entire original IP packet on its own behalf in case of loss, the OAL destination can elect to return only PTB soft errors and refrain from returning Fragmentation Reports.

Note: the OAL source may receive control messages that include both a PTB soft error and Fragmentation Report(s). If so, the OAL source both returns PTB soft errors to the original source (subject to rate limiting) and retransmits any missing fragments if it is configured to do so.

6.10. OAL Composite Packets

The OAL source ordinarily includes a 40 octet IPv6 encapsulation header for each original IP packet during OAL encapsulation. The OAL source then performs fragmentation such that a copy of the 40 octet IPv6 header plus a 16 octet IPv6 Extended Fragment Header is included in each OAL fragment (when a Routing Header is added, the OAL encapsulation headers become larger still). However, these encapsulations may represent excessive overhead in some environments.

OAL header compression as discussed in Section 6.5 can significantly reduce encapsulation overhead, however a complementary technique known as "packing" (see: [I-D.ietf-intarea-tunnels]) supports encapsulation of multiple original IP packets and/or control messages within a single OAL "composite packet".

When the OAL source has multiple original IP packets to send to the same OAL destination with total length no larger than the OAL destination EMTU_R, it can concatenate them into a composite packet encapsulated in a single OAL header. Within the OAL composite packet, the IP header of the first original IP packet (iHa) followed by its data (iDa) is concatenated immediately following the OAL header. The IP header of the next original packet (iHb) followed by its data (iDb) is then concatenated immediately following the first, with each remaining original IP packet concatenated in succession. The OAL composite packet format is transposed from [I-D.ietf-intarea-tunnels] and shown in Figure 13:

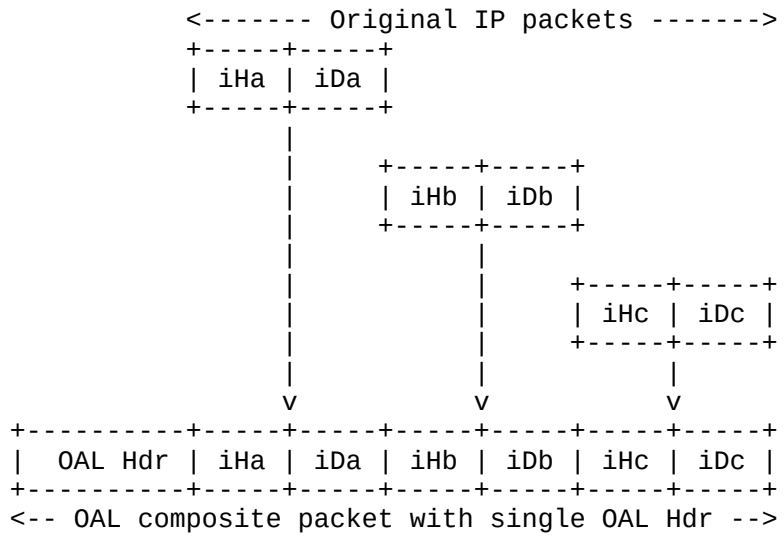

Figure 13: OAL Composite Packet Format

When the OAL source prepares a composite packet, it applies OAL fragmentation then applies underlay encapsulation and sends the resulting carrier packets to the OAL destination. When the OAL destination receives the composite packet it first reassembles if necessary. The OAL destination then selectively extracts each original IP packet (e.g., by setting pointers into the composite packet buffer and maintaining a reference count, by copying each packet into a separate buffer, etc.) and forwards each one to the network layer. During extraction, the OAL determines the IP protocol version of each successive original IP packet 'j' by examining the 4 most-significant bits of iH(j), and determines the length of each one by examining the rest of iH(j) according to the IP protocol version.

When an OAL source prepares a composite packet that includes a control message as the first original IP packet (i.e., iHa/iDa) it includes any additional original IP packets in concatenated succession then includes a trailing OMNI option. If the OMNI option contains an authentication sub-option, the OAL source calculates the authentication signature over the entire length of the composite packet. (A second common use case entails a path MTU probe beginning with an unsigned control message followed by a suitably large NULL packet (e.g., an IP packet with padding octets added beyond the IP header and with {Protocol, Next Header} set to 59 ("No Next Header"), a UDP/IP packet with port number set to 9 ("discard") [RFC0863], etc.)

The OAL source can also apply this composite packet packing technique at the same time it performs OCH1 header compression as discussed in Section 6.5. Note that this technique can only be applied for original IP packets of a single flow, such as for a stream of packets for the flow that are queued for transmission service at roughly the same time.

6.11. OAL Bubbles

OAL sources may send NULL OAL packets known as "bubbles" for example to establish Network Address Translator (NAT) state on the path to the OAL destination. The OAL source prepares a bubble by crafting an OAL header with appropriate IPv6 Source and Destination ULAs, with the IPv6 Next Header field set to the value 59 ("No Next Header" - see: [RFC8200]) and with 0 or more octets of NULL protocol data immediately following the IPv6 header.

The OAL source includes a randomly-chosen Identification value then encapsulates the OAL packet in underlay headers destined to either the mapped address of the OAL destination's first-hop ingress NAT or the underlay address of the OAL destination itself. When the OAL source sends the resulting carrier packet, any egress NATs in the path toward the underlay destination will establish state based on the activity. At the same time, the bubbles themselves will be harmlessly discarded by either an ingress NAT on the path to the OAL destination or by the OAL destination itself.

The bubble concept for establishing NAT state originated in [RFC4380] and was later updated by [RFC6081]. OAL bubbles may be employed by mobility services such as AERO.

6.12. OAL Requirements

In light of the above, OAL sources, destinations and intermediate systems observe the following normative requirements:

* OAL sources MUST forward original IP packets either larger than the OMNI interface minimum EMTU_R or smaller than the minimum OFS as atomic fragments (i.e., and not as multiple fragments).

* OAL sources MUST perform OAL fragmentation such that all non-final fragments are equal in length while the final fragment may be smaller.

* OAL sources MUST produce non-final fragments with payloads no smaller than the minimum OFS during fragmentation.

* OAL intermediate systems SHOULD and OAL destinations MUST unconditionally drop any non-final OAL fragments with payloads smaller than the minimum OFS.

* OAL destinations MUST drop any new OAL fragments that would overlap with other fragments and/or leave holes smaller than the minimum OFS between fragments that have already been received.

Note: Certain legacy network hardware of the past millennium was unable to accept IP fragment "bursts" resulting from a fragmentation event - even to the point that the hardware would reset itself when presented with a burst. This does not seem to be a common problem in the modern era, where fragmentation and reassembly can be readily demonstrated at line rate (e.g., using tools such as 'iperf3') even over fast links on ordinary hardware platforms. Even so, while the OAL destination is reporting reassembly congestion (see: Section 6.9) the OAL source could impose "pacing" by inserting an inter-fragment delay and increasing or decreasing the delay according to congestion indications.

6.13. OAL Fragmentation Security Implications

As discussed in Section 3.7 of [RFC8900], there are 4 basic threats concerning IPv6 fragmentation; each of which is addressed by effective mitigations as follows:

1. Overlapping fragment attacks - reassembly of overlapping fragments is forbidden by [RFC8200]; therefore, this threat does not apply to the OAL.

2. Resource exhaustion attacks - this threat is mitigated by providing a sufficiently large OAL reassembly cache and instituting "fast discard" of incomplete reassemblies that may be part of a buffer exhaustion attack. The reassembly cache should be sufficiently large so that a sustained attack does not cause excessive loss of good reassemblies but not so large that (timer-based) data structure management becomes computationally expensive. The cache should also be indexed based on the arrival underlay interface such that congestion experienced over a first underlay interface does not cause discard of incomplete reassemblies for uncongested underlay interfaces.

3. Attacks based on predictable fragment Identification values - in environments where spoofing is possible, this threat is mitigated through the use of Identification windows beginning with unpredictable values per Section 6.7. By maintaining windows of acceptable Identifications, OAL neighbors can quickly discard spurious carrier packets that might otherwise clutter the reassembly cache.

4. Evasion of Network Intrusion Detection Systems (NIDS) - since the OAL source employs a robust OFS, network-based firewalls can inspect and drop OAL fragments containing malicious data thereby disabling reassembly by the OAL destination. However, each OAL destination should also employ a (host-based) firewall.

IPv4 includes a 2 octet (16-bit) Identification (IP ID) field with only 65535 unique values such that even at moderate data rates the field could wrap and apply to new carrier packets while the fragments of old carrier packets using the same IP ID are still alive in the network [RFC4963]. However, IPv4 links that configure a small MTU are likely to occur only at extreme network edges where low data rate links occur [RFC3819]. Since IPv6 provides a 4 octet (32-bit) Identification value, IP ID wraparound for IPv6 fragmentation may only be a concern at extreme data rates (e.g., 1Tbps or more). These limitations are fully addressed through the 8 octet (64-bit) Extended Identification format supported by [I-D.templin-6man-ipid-ext2].

Unless the path is secured at the network layer or below (i.e., in environments where spoofing is possible), OMNI interfaces MUST NOT send OAL packets/fragments with Identification values outside the current window and MUST secure control messages used for address resolution or window state synchronization. OAL destinations SHOULD therefore discard without reassembling any out-of-window OAL fragments received over an unsecured path.

6.14. Control/Data Plane Considerations

The above sections primarily concern data plane aspects of the OMNI interface service and describe the data plane service model offered to the network layer. OMNI interfaces also internally employ a control plane service based on control messaging. These control plane messages are first subject to OAL encapsulation then forwarded over secured underlay interfaces (e.g., IPsec tunnels, secured direct point-to-point links, etc.) or over unsecured underlay interfaces and with an authentication signature included.

OMNI interfaces must send all control plane messages as "atomic OAL packets". This means that these messages must not be subjected to OAL fragmentation and reassembly, although they may be subjected to

underlay network fragmentation and reassembly along some paths. Fragmentation security concerns for large control messages are documented in [RFC6980].

7. Ethernet-Compatible Link Layer Frame Format

When the OMNI interface forwards original IP packets from the network layer it first invokes OAL encapsulation and fragmentation, then wraps each resulting OAL packet/fragment in any necessary underlay headers to produce carrier packets according to the native frame format of the underlay interface. For example, for Ethernet-compatible interfaces the frame format is specified in [RFC2464], for aeronautical radio interfaces the frame format is specified in standards such as ICAO Doc 9776 (VDL Mode 2 Technical Manual), for various forms of tunnels the frame format is found in the appropriate tunneling specification, etc.

When the OMNI interface encapsulates an OAL packet/fragment directly over an Ethernet-compatible link layer, the over-the-wire transmission format is shown in Figure 14:

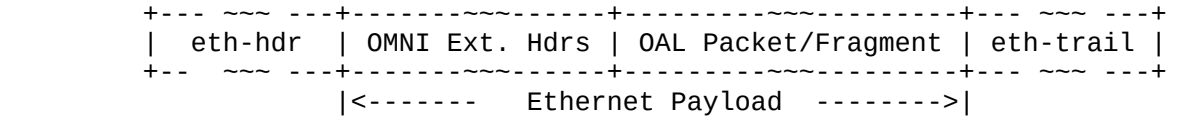

Figure 14: OMNI Ethernet Frame Format

The format includes a standard Ethernet Header ("eth-hdr") with EtherType TBD2 (see: Section 20.2) followed by an Ethernet Payload that includes zero or more OMNI Extension Headers followed by an OAL (or native IPv6/IPv4) Packet/Fragment. The Ethernet Payload is then followed by a standard Ethernet Trailer ("eth-trail").

The first OMNI extension header and the OAL Packet/Fragment both begin with a 4-bit "Type/Version" as discussed in Section 6.2. When "Type/Version" encodes an OMNI extension header type, the length of the extension headers is limited by [I-D.ietf-6man-eh-limits] and the length of the OAL Packet/Fragment is determined by the IP header fields that follow the extension headers.

When "Type/Version" encodes OMNI-OCH1/2, OMNI-IP4 or OMNI-IP6 the length of the OAL Packet/Fragment is determined by the {Total, Payload} Length field found in the full/compressed header according to the specific protocol rules.

See Figure 2 for a map of the various underlay encapsulation layering possibilities. For any layering combination, the final layer (e.g., UDP, IP, Ethernet, etc.) must have an assigned number and frame format representation that is compatible with the selected underlay interface.

8. OMNI Addressing

OMNI addressing observes the IPv6 addressing architecture [RFC4291] requirements: "IPv6 addresses of all types are assigned to interfaces, not nodes. An IPv6 unicast address refers to a single interface. Since each interface belongs to a single node, any of that node's interfaces' unicast addresses may be used as an identifier for the node [...]". OMNI addressing further follows the IPv6 address selection policies specified in [RFC6724] as updated by [I-D.ietf-6man-rfc6724-update].

Each OMNI interface is configured over a set of underlay interfaces as a virtual data link layer for the OAL. OMNI nodes assign IP addresses to their underlay interfaces according to the native *NET autoconfiguration service(s) or through manual configuration. OMNI nodes assign IPv6 addresses to their OMNI interfaces as specified in this section.

[RFC4861] requires that hosts and routers assign Link-Local Addresses (LLAs) to all interfaces including OMNI interfaces, and that routers use their LLAs as the Source Address for RA and Redirect messages. OMNI nodes assign different external and internal LLAs to their OMNI interfaces but need not test them for uniqueness over the entire OMNI link since each (locally-unique) LLA is mapped to an assured globally-unique Multilink Local Address (MLA). OMNI nodes that assign multiple external LLAs to an OMNI interface (e.g., as suggested by [I-D.link-6man-gulla]) map all LLAs to the (singular) OMNI interface MLA.

This specification further requires that each OMNI interface must assign a unique MLA per the address format specified in [RFC9374] and within the architecture of [I-D.templin-6man-mla]. The node assigns the MLA to an OMNI interface configured over its set of underlay interfaces per the IPv6 scoped addressing architecture "site" abstraction [RFC4007]. For OMNI interfaces configured over MANET underlay interfaces, the node also assigns the same MLA to each MANET interface. The node regards the OMNI interface MLA assignment as an adaptation layer address in the architecture and regards the underlay interface MLA assignments as node-local anycast addresses, with each underlay interface distinguished by its ifIndex.

The OMNI link extends across any underlying Internetworks to include all Proxy/Servers and other service nodes. All Clients are also considered to be connected to the OMNI link, however unnecessary encapsulations are omitted whenever possible to conserve bandwidth (see: Section 12). An OMNI domain consists of one or more OMNI links joined together to provide service for a common set of MSPs.

OMNI domains include one or more OMNI links that together coordinate a common set of MSPs delegated from an IP GUA prefix space [RFC4291] from which the MS delegates MNPs to support Client EUN addressing.

For IPv6, MSPs are assigned to an OMNI domain by IANA and/or an associated Regional Internet Registry [IPV6] such that the link(s) can be connected to the global IPv6 Internet without causing routing inconsistencies. Instead of GUAs, an OMNI link could use ULAs with the 'L' bit set to 0 (i.e., from the "ULA-C" prefix fc00::/8) [RFC4193], however this would require IPv6 NAT if the domain were ever connected to the global IPv6 Internet.

For IPv4, MSPs are assigned to an OMNI domain by IANA and/or an associated RIR [IPV4] such that the link(s) can be connected to the global IPv4 Internet without causing routing inconsistencies. An OMNI *NET could instead use private IPv4 prefixes (e.g., 10.0.0.0/8, etc.) [RFC6890], however this would require IPv4 NAT at the *NET boundary. OMNI interfaces advertise IPv4 MSPs into IPv6 routing systems as "6to4 prefixes" [RFC3056] (e.g., the IPv6 prefix for the IPv4 MSP "V4ADDR/24" is 2002:V4ADDR::/40).

IPv4 routers that configure OMNI interfaces advertise the prefix TBD3/N (see: IANA Considerations) into the routing systems of their connected *NETs and assign the IPv4 OMNI anycast address TBD3.1 to their *NET interfaces. IPv6 routers that configure OMNI interfaces advertise the prefix 2002:TBD3::/(N+16) into the routing systems of their connected *NETs and assign the IPv6 OMNI anycast address 2002:TBD3:: to their *NET interfaces.

OMNI interfaces use their OMNI IPv6 and IPv4 anycast addresses to support control plane Service Discovery in the spirit of [RFC7094], i.e., the addresses are not intended for use in supporting longer term data plane flows. Specific applications for OMNI IPv6 and IPv4 anycast addresses are discussed throughout the document as well as in [I-D.templin-6man-aero3].

9. Node Identification

OMNI Clients and Proxy/Servers that connect over open Internetworks include a unique node identification value for themselves in the IPv6 Source Address and/or in an OMNI option of their control messages (see: Section 10.2.9). Each node configures and includes an MLA as a node identification as discussed in Section 8. (The Universally Unique IDentifier (UUID) [RFC9562] is another example of a node identifier which can be self-generated by a node without supporting infrastructure with very low probability of collision.)

When a Client is truly outside the context of any infrastructure, it may have no topology-aggregated addressing information at all. In that case, the Client can use an MLA as an IPv6 Source/Destination Address for sustained communications in Vehicle-to-Vehicle (V2V) and (multihop) Vehicle-to-Infrastructure (V2I) scenarios. The Client can also propagate the MLA into the multihop routing tables of (collective) Mobile/Vehicular Ad-hoc Networks (MANETs/VANETs) using only the vehicles themselves as communications relays. MLAs provide an especially useful node identification construct since they appear as properly-formed IPv6 addresses.

10. Address Mapping - Unicast

OMNI interfaces maintain network layer conceptual Neighbor and Destination Caches per [RFC1256][RFC4861] the same as for any IP interface. The network layer maintains state through static and/or dynamic Neighbor/Destination Cache Entry (NCE/DCE) configurations.

Each OMNI interface also maintains internal ALNCEs that supplement the NLNCEs for each of its active neighbors. For each peer, neighbors also maintain AERO Flow Vectors (AFVs) as ALNCE state to map neighbor per-interface-pair parameters.

When a Client’s network layer sends or receives IPv6 Neighbor Discovery (ND) messages over an OMNI interface, it follows the procedures in [RFC4861] using the Source/Target Link-Layer Address Option (S/TLLAO) format defined for Ethernet [RFC2464]. The OAL then removes the S/TLLAO at the adaptation layer before transmission since the locally-assigned Ethernet address has no significance to external neighbors. On receipt of Neighbor Advertisement and Redirect messages, the OAL inserts a TLLAO with the OMNI interface internal link-layer address then re-calculates the ICMPv6 message checksum and forwards it to the network layer.

When a Client’s network layer sends or receives an ordinary IP packet over an OMNI interface, the OAL consults the IP Destination to OAL IPv6 address mappings established by earlier control message

exchanges. On transmission, the OAL uses the IP destination address to determine the Destination Address for an OAL encapsulation header while including an SRH extension. On reception, the OAL uses the IPv6 encapsulation header Source Address to determine the source address for the virtual Ethernet header.

The OMNI interface must therefore maintain ALNCEs that map IP Destination addresses to MLAs while exposing only the OMNI interface internal link-layer address to the IP layer. When the OMNI interface discovers a new neighbor (e.g., when it creates a new NCE based on receipt of an IPv6 ND message), it maps the MLA to the OMNI interface internal link-layer address. When the OMNI interface discards an existing neighbor, it deletes the now-expired NCE.

An OAL destination or intermediate system may also need to return ICMPv6 error messages (e.g., Destination Unreachable, Packet Too Big, Time Exceeded, Parameter Problem, etc.) [RFC4443] to the OAL source. The OAL destination or intermediate system marks the ICMPv6 error message as an OMNI control message and includes a trailing OMNI option with any necessary authentication sub-options as discussed below. The OAL source can then confirm that the message originated from a trusted OAL node on the path.

When the OAL forwards control messages from the network layer to the underlay, it replaces the Ethernet header with an adaptation layer IPv6 encapsulation header (plus an SRH extension) and a pseudo IPv6 ND option trailer encoding OMNI link-specific information. When the OAL forwards IPv6 ND messages from the underlay to the network layer, it performs decapsulation by parsing and removing the trailer while replacing the adaptation layer IPv6 header with a local Ethernet header.

When the OAL forwards a control message from the network layer to the underlay, it can verify the control message checksum to ensure integrity in the local network stack as an OPTIONAL first step before performing OAL encapsulation. When the OAL alters a network layer control message or creates an internally-generated control message that it will forward to the underlay, it sets the control message checksum to 0 since an OAL checksum will properly cover the same data (see: Section 10.1).

When the OAL receives an OAL-encapsulated control message from the underlay, it ignores the control message checksum if it will process the message internally since integrity is already verified by an OAL checksum in the trailing OMNI option (see below). For control messages that it will forward to the network layer that have the control message checksum set to 0, the OAL instead calculates and rewrites the checksum following OAL decapsulation. The network layer will then verify the control message checksum independently of the OAL the same as for any IPv6 interface.

Hence, this document defines a new pseudo IPv6 ND option type termed the "OMNI option" designed for these purposes. Since the pseudo-option is inserted and removed by the adaptation layer and never exposed to the network layer, it does not require a formal IPv6 ND option number assignment.

10.1. The OMNI Option

During OAL IPv6 encapsulation of each control message, the OAL source appends a single OMNI (pseudo-)option as a contiguous block of data immediately following the end of the (composite) packet. The OAL source then sets the DSCP value as specified in Section 6.1 to mark the message as control. If the composite packet does not end on an integral 8 octet boundary, the OAL source inserts padding octets following the final composite packet element to ensure 8 octet alignment before appending the rest of the OMNI option. The OAL source then adds the OMNI option length (including padding) to the OAL Payload Length.

During decapsulation of each control message, the OAL destination processes the OMNI option contents then removes the option before delivering the original control message (plus any additional original IP packets from the composite packet) to the network layer. The OAL instead consumes control messages specific to the adaptation layer internally without delivering them to the network layer.

The OMNI option therefore appears as a trailer in all OAL (composite) control packets. The OMNI option format is shown in Figure 15.

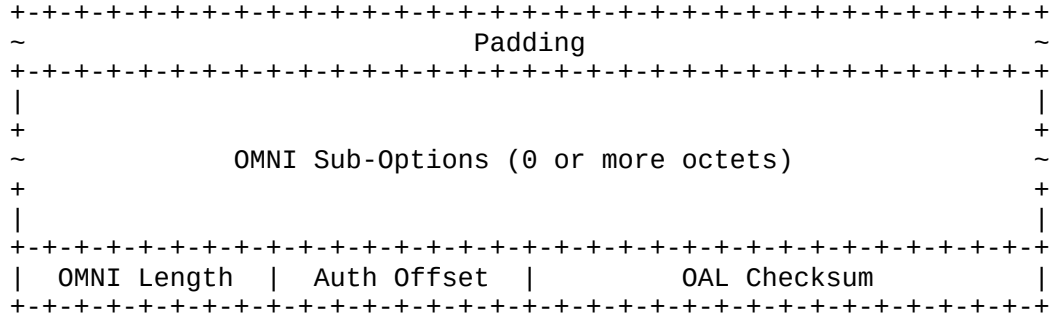

Figure 15: OMNI Option Format

In this format:

* Padding is included if necessary to begin the OMNI option on an even 8 octet boundary.

* OMNI Sub-Options is a variable-length concatenation of 0 or more sub-options formatted as specified in Section 10.2 such that the total length of all sub-options is an integer multiple of 8 octets. If an HMAC or RSA Signature authentication sub-option is included, it must appear as the final sub-option immediately preceded by any "authentication helper" sub-options necessary to support authentication.

* OMNI Length is a 1 octet field that encodes the length in 8 octet units of the immediately preceding OMNI Sub-Options data block. If no sub-options are present, OMNI Length must encode the value 0.

* Auth Offset is a 1 octet field that encodes the offset in 8 octet units to the beginning of the first authentication (or "authentication helper") sub-option relative to the beginning of the OMNI sub-options. If authentication sub-options are present, Auth Offset must encode a value smaller than OMNI Length; otherwise, it must encode a value no smaller than OMNI Length. For example, when OMNI sub-options includes 20 8-octet units and a first authentication/helper sub-option begins at the 12th 8-octet unit, OMNI Length encodes the value 20 and Auth Offset encodes the value 12. If there are no authentication sub-options, Auth Offset instead encodes the value 20 or larger.

* OAL Checksum is a 2 octet field used to ensure integrity and protect against misdelivery. If the OMNI option includes an authentication sub-option, the OAL source calculates and includes the authentication signature first. The OAL source then

calculates the OAL Checksum beginning with a pseudo-header of the OAL IPv6 header per Section 8.1 of [RFC8200]. The pseudo-header includes the OAL IPv6 Source and Destination Addresses, where the MLAs of the source and its FHS or LHS egress peer (e.g., a Client and its Proxy/Server) are used if there are SRH intermediate hops between them. The pseudo-header also sets Next Header to 41 (for IPv6 encapsulation), and sets Upper-Layer Packet Length to the OAL IPv6 Payload Length minus the length of any extension headers present between the OAL IPv6 header and the encapsulated control message. The checksum then extends over the length of the message beginning with the first octet of the OMNI-encapsulated control message and continuing over the OMNI option up to and including the OMNI Length and Auth Offset fields. The OAL source then writes the resulting value into the OAL Checksum field. The OAL destination verifies the checksum upon message receipt and processes the message further only if the checksum is correct.

OMNI encapsulated control messages exchanged over unsecured *NETs between peer Clients or Clients and their Proxy/Servers use either public-key based digital signatures per SEcure Neighbor Discovery (SEND) [RFC3971][RFC9374] or Hashed Message Authentication Codes (HMAC) per [RFC8754][RFC2104] as an adaptation layer authentication service. Since the adaptation layer already applies authentication from within the OMNI interface, the network layer need not also apply IPv6 ND message authentication over the OMNI interface. The OMNI option therefore provides sub-options to support either SEND or HMAC as adaptation layer authentication services. Alternate authentication sub-option types may be specified in future documents.

Although originally specified to operate with Cryptographically Generated Addresses (CGAs) per [RFC3972], SEND notes that: "This specification also allows a node to use non-CGAs with certificates that authorize their use. However, the details of such use are beyond the scope of this specification and are left for future work." OMNI is based on an alternate cryptographically generated IPv6 address type (the MLA), and therefore does not use CGAs.

The OMNI Sub-Options may include full or partial information for the neighbor. The OMNI interface therefore retains the union of the most recently received information in the corresponding NCE.

OMNI interface Clients such as aircraft typically have multiple wireless data link types (e.g. satellite-based, cellular, terrestrial, air-to-air directional, etc.) with diverse performance, cost and availability properties. The OMNI interface would therefore appear to have multiple L2 connections, and may include information for multiple underlay interfaces in a single OMNI option. OMNI interfaces manage their dynamically-changing multilink profiles by including OMNI sub-options as discussed in Section 10.2.

10.2. OMNI Sub-Options

The OMNI option includes a Sub-Options block containing zero or more individual sub-options. Each successive sub-option is concatenated immediately following its predecessor. All sub-options are encoded as follows:

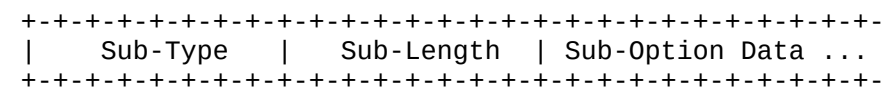

Figure 16: Sub-Option Format

* Sub-Type is a 1 octet field that encodes the sub-option type. Sub-option types defined in this document include:

Sub-Option Name	Sub-Type
Null	0
CGA	1
RSA Signature	2
Timestamp	3
Nonce	4
Trust Anchor	5
Certificate	6
HMAC	7
Node Identification	8
Neighbor Synchronization	9
Interface Attributes	10
Traffic Selector	11
Geo Coordinates	12
PIM-SM Message	13
Fragmentation Report	14
Proxy/Server Control	15
Prefix Information Option	16
Route Information Option	17
DHCPv6 Message	18

Figure 17

Sub-Types 19-252 are reserved for future sub-option assignments. Sub-Types 253 and 254 are reserved for experimentation while Sub-Type 255 is reserved by IANA.

* Sub-Length is a 1 octet field that encodes the length of the sub-option (including the type and length fields) in units of 8 octets. The value 0 is invalid; nodes MUST silently discard an ND packet that contains a sub-option with Sub-Length set to 0.

* Sub-Option Data is a block of data with format determined by Sub-Type and length determined by Sub-Length.

The OAL source codes all sub-options in a single OMNI option in the same control message, with each sub-option concatenated immediately following the previous and with padding added if necessary to end each sub-option on an even 8 octet boundary. If the total size of all sub-options would cause the control message to exceed the path MTU, the OAL source includes as many sub-options as possible and codes any remaining sub-options in additional control messages.

The OAL destination processes the OMNI option in a received control message beginning with the first sub-option while skipping over and ignoring any NULL or unrecognized sub-options. If an individual sub-option length would cause processing to exceed the OMNI option instance and/or control message lengths, the OAL destination drops the message.

Control messages that require OMNI authentication services include an RSA or HMAC authentication sub-option as the final sub-option immediately preceded by any "authentication helper" sub-options. A single control message includes at most one OMNI authentication service sub-option; if multiple appear (or if the sub-option is non-final) the message is dropped.

Note: large objects that exceed the maximum Sub-Option Data length are not supported under the current specification; if this proves too limiting in practice, future specifications may define support for fragmenting large sub-options across multiple control messages.

The following sub-option types and formats are defined in this document:

10.2.1. NULL Sub-Option

The NULL Sub-Option is skipped over and ignored on receipt regardless of the contents of the Sub-Option Data following the Sub-Length octet. The format and contents of the sub-option are shown in Figure 18:

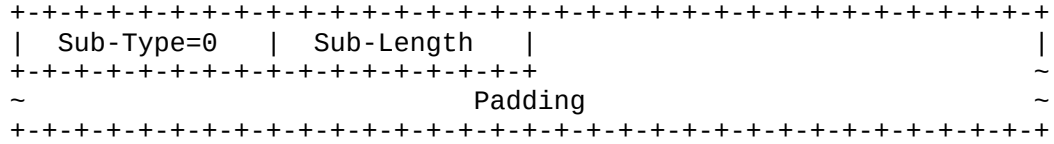

Figure 18: NULL Sub-Option

* Sub-Type is set to 0. Multiple instances may appear in the same OMNI option.

* Sub-Length is set to the number of 8 octet units in the sub-option including the Sub-Option Data that follows.

* Sub-Option Data includes non-offensive padding data which must be ignored on receipt, where any non-null padding is subject to the OAL checksum and authentication.

Note that an authorized intermediate system can convert any sub-option into a NULL sub-option simply by writing the value 0 into its Sub-Type and resetting the OAL checksum. Unauthorized intermediate systems cannot perform this conversion without invalidating authentication.

10.2.2. CGA

The OAL source includes a Cryptographically Generated Address (CGA) OMNI sub-option formatted as specified in Section 5.1 of [RFC3971] except that OMNI interfaces use MLAs (instead of CGAs) which are also cryptographically-generated [RFC9374]. The OMNI CGA sub-option has the following format:

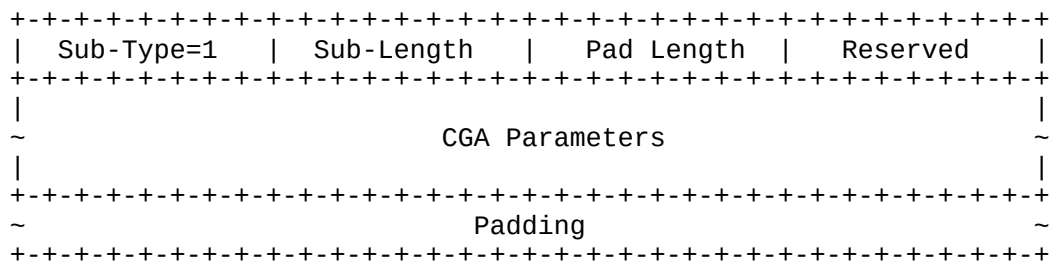

Figure 19: CGA

* Sub-Type is set to 1. The CGA sub-option may appear at most once in any OMNI option; if multiple appear the message is dropped.

* Sub-Length is set to the number of 8 octet units in the sub-option including the Sub-Option Data that follows.

Among other parameters, the sub-option includes a public key as specified in [RFC3972] applied to the MLA Source Address according to [RFC9374].

The CGA option must appear in all secured OMNI control messages that also include an RSA Signature sub-option.

10.2.3. RSA Signature

The OMNI RSA Signature sub-option includes a public key-based authentication signature extending over the length of the OMNI-encapsulated control message. When present, the RSA Signature sub-option must appear as the final OMNI sub-option and must be immediately preceded by any "authentication helper" sub-options such as CGA, Nonce, Timestamp, etc.

The OAL source fully populates the OMNI option and can then calculate a digital signature to include in an OMNI RSA Signature sub-option as discussed below. The OAL source sets Auth Offset to the offset of the first "authentication helper" sub-option relative to the beginning of the OMNI sub-options.

The RSA Signature sub-option is formatted as shown in Figure 20:

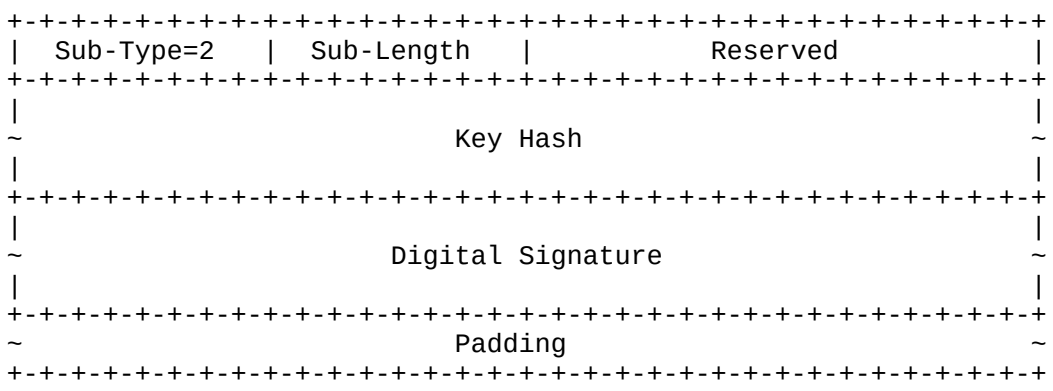

Figure 20: RSA Signature

* Sub-Type is set to 2. The RSA Signature sub-option may appear at most once in any OMNI option; if multiple appear the message is dropped.

* Sub-Length is set to the length of the sub-option in 8 octet units. The Key Hash is always the most significant (leftmost) 128 bits of the secure hash of the public key used for constructing the signature while the length of the Digital Signature plus Padding is constrained by the remaining available space for this sub-option.

After fully populating the OMNI sub-options, the OAL source constructs the Digital Signature according to Section 5.2 of [RFC3971] except beginning with the 128-bit Context ID value specified for MLAs in Section 3 of [RFC9374] instead of the CGA Message Type value.

The signature then continues over the OAL IPv6 Source and Destination Addresses, where the MLAs of the source and its FHS or LHS egress peer are used if there are SRH intermediate hops between them. The signature next continues over the encapsulated control message including all control message header and option contents. The signature next continues over any composite packet extensions then extends over the entire OMNI option up to the beginning of the RSA sub-option itself. The signature finally concludes by covering the trailing OMNI Length and Auth Offset fields that follow the RSA sub-option.

After calculating the signature, the node writes the value into the Digital Signature field before calculating the trailing OAL Checksum.

Note that the control message MLA Source Address encodes security suite information that determines the cryptographic algorithms and Digital Signature length [RFC9374]. The OMNI RSA Signature sub-option can therefore also carry non-RSA Digital Signatures.

10.2.4. Timestamp

The OAL source includes an OMNI Timestamp sub-option in control messages to ensure that unsolicited advertisements and redirects have not been replayed. If multiple Timestamp sub-options appear the message is dropped.

The Timestamp sub-option is processed exactly the same as in Section 5.3.1 of [RFC3971]. The OMNI Timestamp sub-option has the following format:

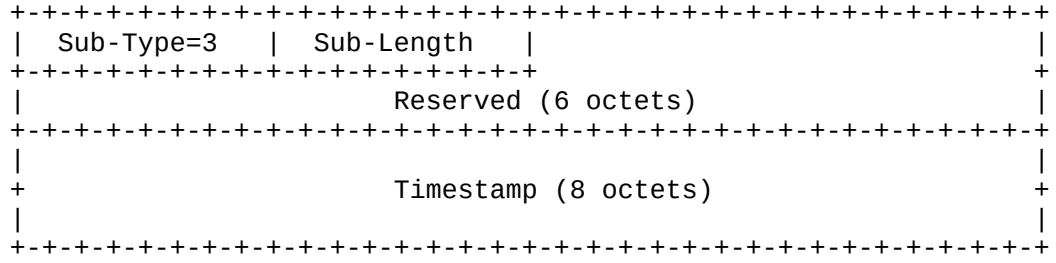

Figure 21: Timestamp

* Sub-Type is set to 3. The Timestamp sub-option may appear at most once in any OMNI option; if multiple appear the message is dropped.

* Sub-Length is set to 2.

10.2.5. Nonce

The OAL source includes an OMNI Nonce sub-option to ensure that an IPv6 ND advertisement is a fresh response to one of its earlier solicitations.

The Nonce sub-option is processed exactly the same as in Section 5.3.2 of [RFC3971] per the following format:

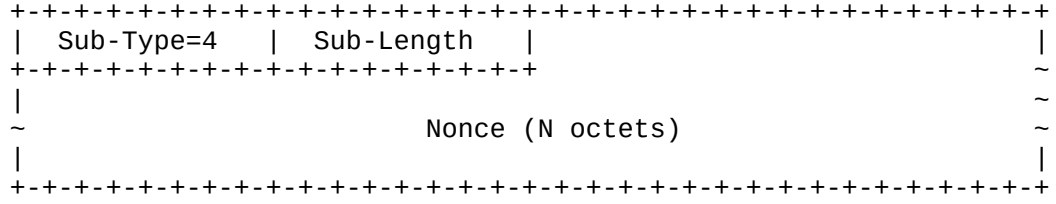

Figure 22: Nonce

* Sub-Type is set to 4. The Nonce sub-option may appear at most once in any OMNI option; if multiple appear the message is dropped.

* Sub-Length is set to the number of 8 octet units in the sub-option including the Sub-Option Data that follows, where N is ((Sub-Length * 8) - 2).

10.2.6. Trust Anchor

The OAL source includes a Trust Anchor OMNI sub-option the same as in Section 6.4.3 of [RFC3971] per the following format:

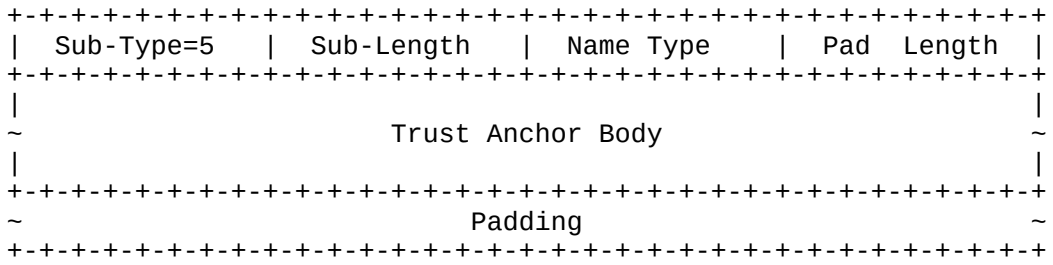

Figure 23: Trust Anchor

* Sub-Type is set to 5. The Trust Anchor sub-option may appear at most once in any OMNI option; if multiple appear the message is dropped.

* Sub-Length is set to the number of 8 octet units in the sub-option including the Sub-Option Data that follows.

10.2.7. Certificate

The OAL source includes a Certificate OMNI sub-option the same as in Section 6.4.4 of [RFC3971] per the following format:

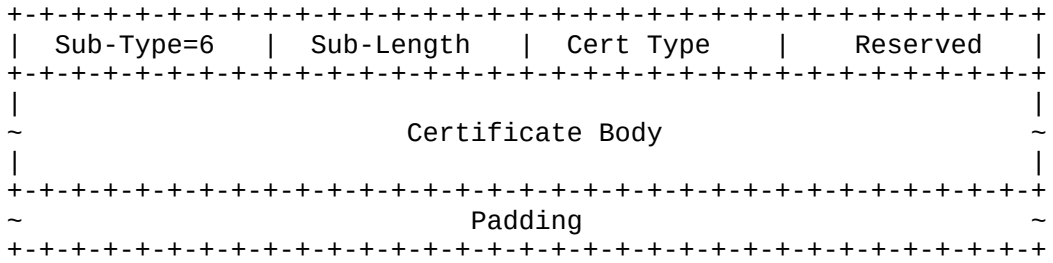

Figure 24: Certificate

* Sub-Type is set to 6. The Certificate sub-option may appear at most once in any OMNI option; if multiple appear the message is dropped.

* Sub-Length is set to the number of 8 octet units in the sub-option including the Sub-Option Data that follows.

10.2.8. Hashed Message Authentication Code (HMAC)

The OAL source may include a Hashed Message Authentication Code (HMAC) sub-option. When present, the HMAC sub-option appears as the final sub-option and must be immediately preceded by any "authentication helper" sub-options such as Nonce, Timestamp, etc. the same as specified for RSA Signature above.

The OAL source fully populates the OMNI option and can then calculate a digital signature to include in an OMNI HMAC sub-option as discussed below. The OAL source sets Auth Offset to the offset of the first "authentication helper" sub-option relative to the beginning of the OMNI sub-options.

The format of the HMAC option is taken directly from Section 2.1.2 of [RFC8754] as shown in Figure 25:

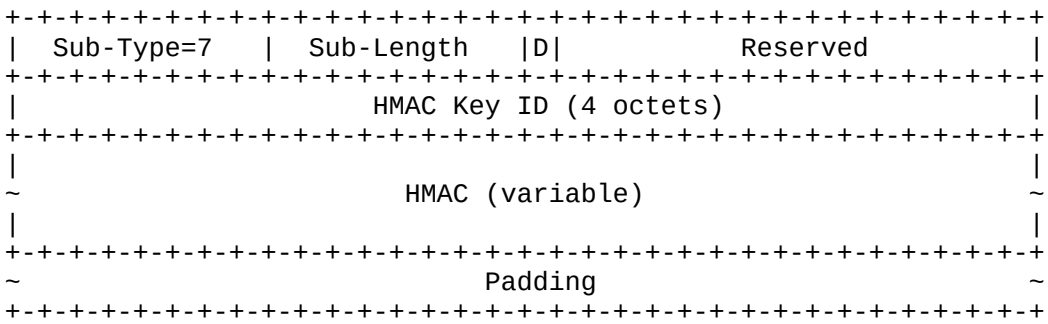

Figure 25: Hashed Message Authentication Code (HMAC)

Sub-Type is set to 7, and Sub-Length is set to the number of 8 octet units in the sub-option (note that this type value differs from the HMAC type defined for TLV options in [RFC8754] since the two (sub-)option numbering spaces are independent). The HMAC sub-option may appear at most once in any OMNI option; if multiple appear the message is dropped.

The HMAC digest is encoded and processed the same as specified in [RFC2104] and Section 2.1.2 of [RFC8754]. The HMAC secure hash algorithm (e.g., SHA-256, SHA3-256, etc.) is determined by consulting local network configuration indexed by the HMAC Key ID.

The OAL node applies the HMAC over the OAL IPv6 Source and Destination Addresses, where the MLAs of the source and its FHS or LHS egress peer are used if there are SRH intermediate hops between them. The HMAC then continues over the encapsulated control message including all control message header and option contents. The HMAC

next continues over any composite packet extensions then extends over the entire OMNI option up to and including the HMAC key ID. The HMAC finally concludes by covering the trailing OMNI Length and Auth Offset fields that follow the HMAC sub-option.

After calculating the HMAC digest, the OAL node writes the value into the HMAC field then includes padding octets if necessary for 8 octet alignment before calculating the trailing OAL Checksum. Only FHS and LHS OAL nodes need to include and verify the HMAC, since intermediate SRT hops engage the secured spanning tree. The HMAC is inserted and calculated by the FHS/LHS ingress node then verified and removed by the FHS/LHS egress node.

10.2.9. Node Identification

The OAL source may include the Node Identification sub-option as supplementary identification information in addition to the control message Source Address. If multiple instances appear in the same OMNI option, the first instance of a specific ID-Type is processed and all other instances of the same ID-Type are ignored. (A single control message can therefore include multiple distinct Node Identifications - each with a different ID-Type.)

The format and contents of the sub-option are shown in Figure 26:

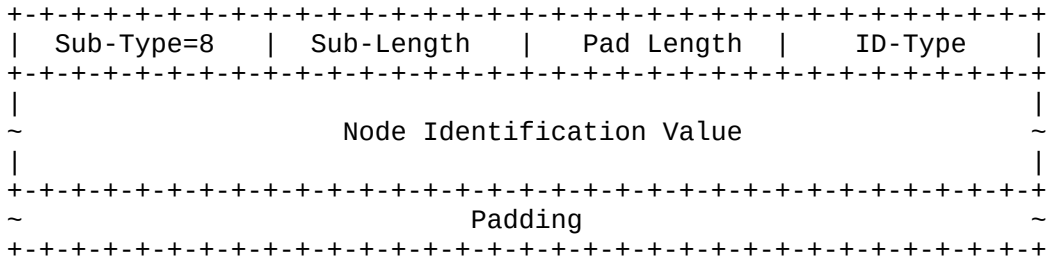

Figure 26: Node Identification

* Sub-Type is set to 8. Multiple instances are processed as discussed above.

* Sub-Length is set to the number of 8 octet units in the sub-option including the Sub-Option Data that follows.

* Pad Length encodes the number of trailing padding octets required to end the sub-option on an even 8 octet boundary. The ID-Type field is always present, and the maximum Node Identification Value length is limited by the remaining available space in this OMNI option.

* ID-Type is a 1 octet field that encodes the type of the Node Identification Value. The following ID-Type values are currently defined:

  - 0 - Multilink Local Address (MLA). A special-purpose IPv6 address assigned to an OMNI interface for adaptation layer addressing as discussed in Section 8. Indicates that Node Identification Value contains a 16 octet MLA.

  - 1 - Universally Unique IDentifier (UUID) [RFC9562]. Indicates that Node Identification Value contains a 16 octet UUID.

  - 2 - Network Access Identifier (NAI) [RFC7542]. Indicates that Node Identification Value contains an N octet NAI.

  - 3 - Fully-Qualified Domain Name (FQDN) [RFC1035]. Indicates that Node Identification Value contains an N octet FQDN.

  - 4 - IPv4 Address. Indicates that Node Identification contains a 4 octet IPv4 address. The IPv4 address type is determined with reference to the IANA IPv4 Address Space Registry [IPV4].

  - 5 - Router ID (RID). Indicates that Node Identification contains an N octet router ID other than an IPv4 or IPv6 address. May be useful for some MANET routing protocols that define their own RID formats.

  - 6 - IPv6 Address. Indicates that Node Identification contains a general-purpose 16 octet IPv6 address that is not an MLA. The IPv6 address type is determined according to the IPv6 addressing architecture [RFC4291] with reference to the IANA IPv6 Global Unicast Address Assignments Registry [IPV6].

  - 7 - 252 - Unassigned.

  - 253 - 254 - reserved for experimentation, as recommended in [RFC3692].

  - 255 - reserved by IANA.

* Node Identification Value is an N octet field encoded according to the appropriate the "ID-Type" reference above. The Node Identification Value is followed by the number of padding octets indicated by Pad Length.

The OAL source encodes Node Identification Values used for DHCPv6 messaging purposes as DHCP Unique IDentifiers (DUIDs) using the "DUID-EN for OMNI" format with enterprise number 45282 (see: Section 20) as shown in Figure 27:

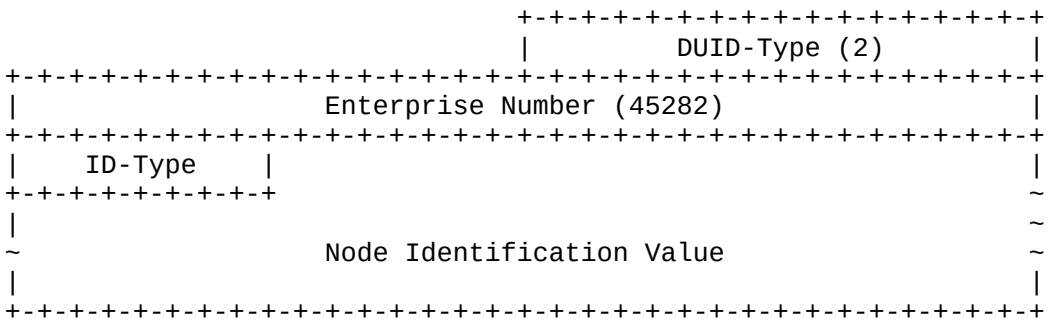

Figure 27: DUID-EN for OMNI Format

In this format, the OAL source codes the ID-Type and Node Identification Value fields from the OMNI sub-option following a 6 octet DUID-EN header, then includes the entire "DUID-EN for OMNI" in a DHCPv6 message per [RFC8415].

10.2.10. Neighbor Synchronization

The OAL source includes a Neighbor Synchronization OMNI sub-option in control messages that establish or update neighbor state between Clients and their Proxy/Servers or peers. Each control message includes at most one Neighbor Synchronization sub-option which must be specific to the underlying interface pair over which ND messages are exchanged.

The Neighbor Synchronization sub-option is formatted as follows:

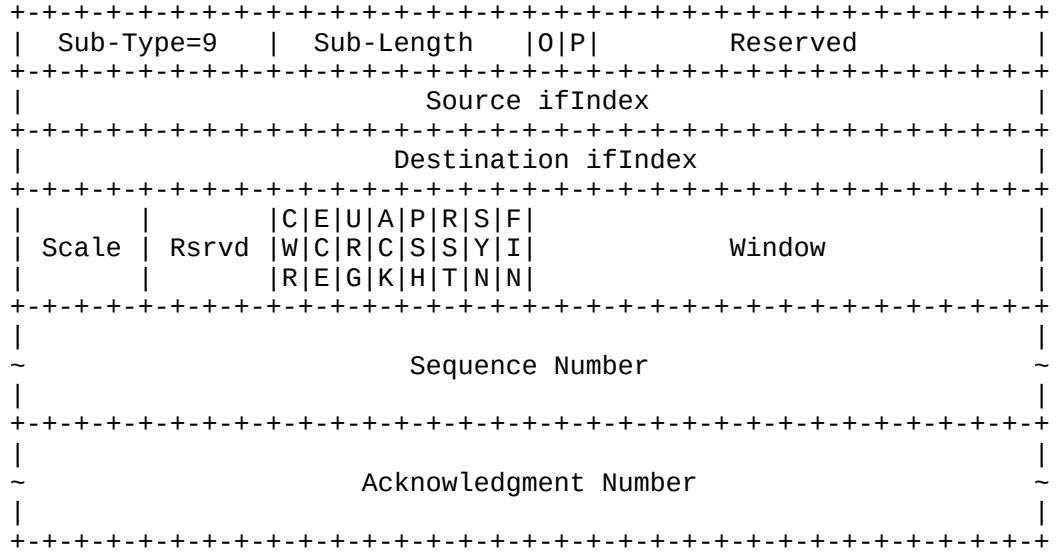

Figure 28: Neighbor Synchronization

* Sub-Type is set to 9. If multiple instances appear in the same OMNI option, the first is processed and all others are ignored. Sub-Length is set to 2, 3 or 4 depending on whether a Sequence Number and/or Acknowledgement Number is included - see below.

* the next 2 octets include functional and reserved flags. Neighbors set the (O)ptional (OPT) flag as discussed in Section 6.7 in a SYN/ACK synchronization message that does not require a responsive ACK. OAL intermediate systems set the (P)ath Change (PCH) flag in control messages used to report a change in a path established by multilink forwarding.

* the next 8 octets of Sub-Option Data includes the 4 octet ifIndex of the control message source node’s underlay interface followed by the 4 octet ifIndex of the destination node’s underlay interface.

* the remainder of Sub-Option Data is modeled from the Transmission Control Protocol (TCP) header specified in Section 3.1 of [RFC9293]. The data begins with a 4-bit window Scale, followed by a 4-bit Reserved field, followed by 8 flags, followed by a 2 octet Window size. An 8 octet Sequence Number follows only if the SYN flag is set, and an 8 octet Acknowledgement Number follows only if the ACK flag is set. (If the SYN/ACK flag settings and Sub-Length value are inconsistent, the message MUST be discarded.) Intermediate nodes always cache the Sequence Number, window Scale and Window size values when the SYN flag is set.

10.2.11. Interface Attributes

The Interface Attributes sub-option provides neighbors with forwarding information for the multilink conceptual sending algorithm discussed in Section 12. Neighbors use the forwarding information to select among candidate underlay interfaces that can be used to forward carrier packets to the neighbor based on factors such as traffic selectors and link metrics. Interface Attributes further include link-layer address information to be used for either direct INET encapsulation for targets in the local SRT segment or spanning tree forwarding for targets in remote SRT segments.

The OAL source includes Interface Attributes for some/all of a source or target Client’s underlay interfaces in control solicitation and response messages that exchange peer-to-peer Client information (see: [I-D.templin-6man-aero3]). The first Interface Attributes sub-option included MUST correspond to the interface used to transmit the control message. At most one Interface Attributes sub-option for each distinct ifIndex may be included; if a control message includes multiple Interface Attributes sub-options for the same ifIndex, the first is processed and all others are ignored. OMNI nodes that receive control messages can use all of the included Interface Attributes and/or Traffic Selectors to formulate a map of the prospective source or target node as well as to seed the information to be populated in future neighbor exchanges.

OMNI Clients and Proxy/Servers also include Interface Attributes sub-options in RS/RA messages used to initialize, discover and populate routing and addressing information. Each RS message MUST contain exactly one Interface Attributes sub-option with an ifIndex corresponding to the Client’s underlay interface used to transmit the message, and each RA message MUST echo the same Interface Attributes sub-option with any (proxyed) information populated by the FHS Proxy/Server to provide operational context.

When an FHS Proxy/Server receives an RS message destined to an anycast underlay address, it MUST include an additional Interface Attributes sub-option with ifIndex '0' that encodes its own unicast underlay address relative to the Client's underlay interface in the solicited RA response. Any additional Interface Attributes sub-options that appear in RS/RA messages (i.e., besides those for the Client's own ifIndex and ifIndex '0') are ignored.

The Interface Attributes sub-option is formatted as shown below:

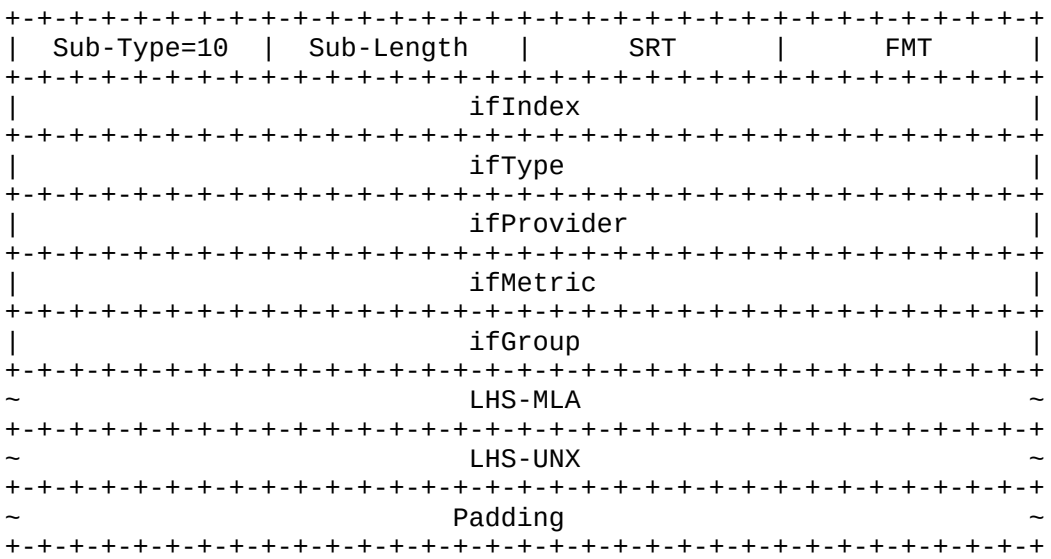

Figure 29: Interface Attributes

* Sub-Type is set to 10. Multiple instances are processed as discussed above. Sub-Length is set to the number of 8 octet units in the option. The SRT and FMT fields are specified below in conjunction with the LHS fields.

* Sub-Option Data contains an "Interface Attributes" option encoded as follows:

  - ifIndex is a 4 octet index value corresponding to a specific underlay interface. Client OMNI interfaces MUST number each distinct underlay interface with a unique non-zero ifIndex value assigned by network management per [RFC2863] and include the value in this field. The ifIndex value '0' denotes "unspecified".

- ifType is a 4 octet type value corresponding to this underlay interface. The value is coded per the IANA "https://www.iana.org/assignments/smi-numbers registry group Interface Types (ifType)" registry according to [RFC8892].

- ifProvider is a 4 octet provider identifier corresponding to this underlay interface. This document defines the single provider identifier value '0' (undefined). Future documents may define other values.

- ifMetric encodes a 4 octet interface metric. Lower values indicate higher priorities, and the highest value indicates an interface that should not be selected. The ifMetric setting provides an instantaneous indication of the interface bandwidth, link quality, signal strength, cost, etc.; hence, its value may change in successive control messages.

- ifGroup is a 4 octet identifier for a Link Aggregation Group (LAG) [IEEE802.1AX] corresponding to the underlay interface identified by ifIndex. Interface attributes for ifIndex members of the same group will encode the same value in ifGroup. This document defines the single ifGroup value '0' meaning "no group assigned". Future documents will specify the setting of other values.

- SRT is a 1 octet Segment Routing Topology control field. The field contains 8 Reserved flags which must be set to 0 on transmission. Future specifications may define new SRT control flags.

- FMT - a 1 octet "Forward/Mode/Type" code interpreted as follows:

  o The most significant 2 bits (i.e., "FMT-Forward" and "FMT-Mode") are interpreted in conjunction with one another. When FMT-Forward is clear, the LHS Proxy/Server performs OAL reassembly and decapsulation to obtain the original IP packet before forwarding. If the FMT-Mode bit is clear, the LHS Proxy/Server then forwards the original IP packet at L3; otherwise, it invokes the OAL to reassemble, re-fragment and re-encapsulate then sends the resulting carrier packets to the Client via the selected underlay interface. When FMT-Forward is set, the LHS Proxy/Server forwards unmodified OAL fragments to the Client without reassembling. If FMT-Mode is clear, all carrier packets destined to the Client must always be sent via the LHS Proxy/Server; otherwise the Client is eligible for direct forwarding over the open INET where it may be located behind one or more NATs.

o The least significant 6 bits ("FMT-Type") determines the type of underlay encapsulation needed to reach the target Client interface within its local *NET. When the most significant bit (msb) of FMT-Type is set, the interface has been determined to reside behind a Network Address Translator (NAT) as discovered during Client exchanges with their Proxy/Servers. The least significant 5 bits of FMT-Type encode an underlay encapsulation type value as follows:

+ 0 - underlay encapsulation type is unspecified. No UNX address is included and the msb is ignored.

+ 1 - Client interface is within a MANET where multihop forwarding occurs as an adaptation layer service. No UNX address is included and the msb is ignored.

+ 2 - underlay encapsulation type is EUI-48 only. UNX is 6 octets in length and encodes an EUI-48 address [EUI].

+ 3 - underlay encapsulation type is EUI-64 only. UNX is 8 octets in length and encodes an EUI-64 address [EUI].

+ 4 - underlay encapsulation type is IPv4 only. UNX is 4 octets in length and encodes an IPv4 address.

+ 6 - underlay encapsulation type is IPv6 only. UNX is 16 octets in length and encodes an IPv6 address.

+ 7 - underlay encapsulation type is UDP/IPv4. UNX is 6 octets in length and encodes a 4 octet IPv4 address followed by a 2 octet UDP port number.

+ 8 - underlay encapsulation type is UDP/IPv6. UNX is 18 octets in length and encodes a 16 octet IPv6 address followed by a 2 octet UDP port number.

+ 5, [9 - 31] - Reserved for future use.

- LHS-MLA is the 16 octet MLA of the LHS Proxy/Server for the specified target Client interface.

- LHS-UNX is L octets in length according to FMT-Type as discussed above. LHS-UNX identifies the LHS Client's *NET interface which may connect to an open INET or a private *NET behind one or more NATs. When LHS-UNX includes an IPv4 or IPv6 address, it appears in network byte order in ones-complement "obfuscated" form per [RFC4380]. Trailing padding is added following LHS-UNX if necessary to end the sub-option on an even 8 octet boundary.

- The LHS information therefore satisfies per-interface address resolution and SRT/FMT/LHS together inform the OMNI interface forwarding algorithm. If the FHS and LHS SRT segments are one and the same, the source can address the target Client either via its Proxy/Server or through direct underlay encapsulation (while engaging NAT traversal in the underlay if necessary) according to FMT. If the target Client is located on a different SRT segment, the path from the source must employ a combination of route optimization and spanning tree hop traversals.

10.2.12. Traffic Selector

The Traffic Selector sub-option provides flow binding information for the multilink conceptual sending algorithm discussed in Section 12. The sub-option includes an augmented traffic selector per [RFC6088] as ancillary information for an Interface Attributes sub-option with the same ifIndex value, or as discrete information for the included ifIndex when no Interface Attributes sub-option is present.

All packets of the same flow should include compatible traffic selector profiles, as the flow (i.e., and not individual packets) determines path selection.

Control messages may include multiple Traffic Selectors for some or all of the source/target Client's underlay interfaces (see: [I-D.templin-6man-aero3] for further discussion). Any included Traffic Selector sub-options MUST appear in the format shown below:

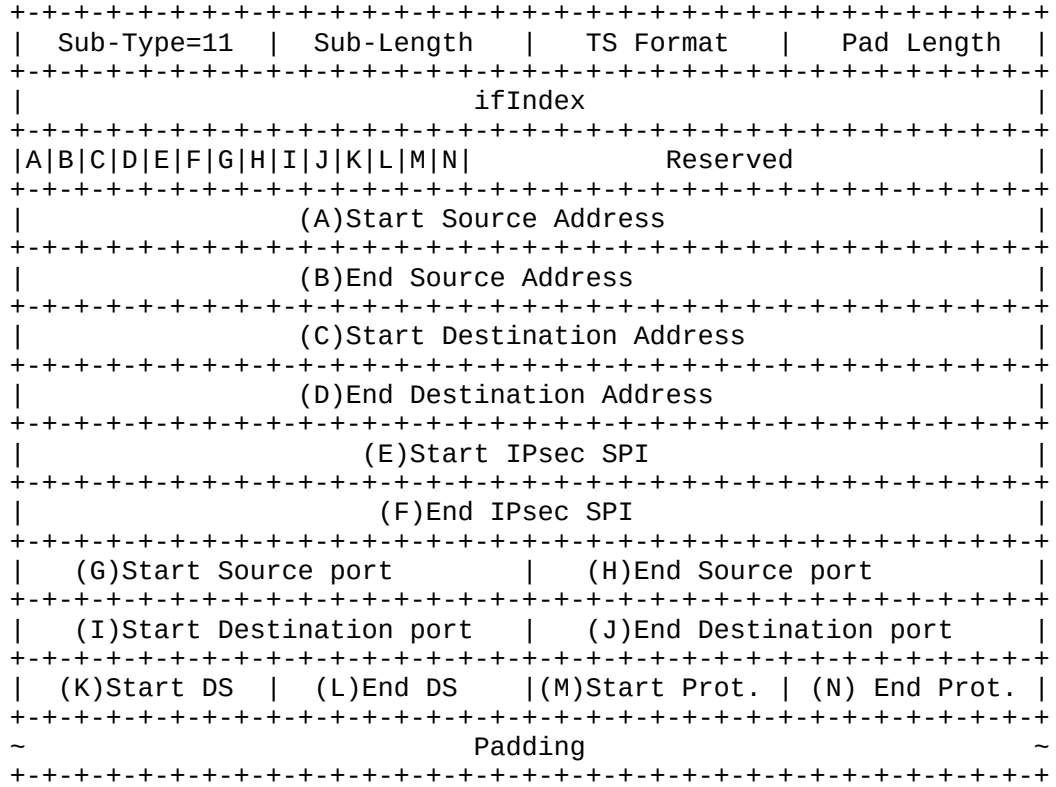

Figure 30: Traffic Selector

* Sub-Type is set to 11. Multiple instances with the same or different ifIndex values may appear in the same OMNI option. When multiple instances appear, all are processed and the cumulative information from all is accepted. Sub-Length is set to the number of 8 octet units in the sub-option.

* Sub-Option data begins with 1 octet "TS Format" and "Pad Length" fields, where Pad Length indicates the number of trailing padding octets. These fields are followed by a 4 octet ifIndex value corresponding to a specific underlay interface.

* When TS Format encodes the value 1 or 2, the Traffic Selector body encodes an IPv4 or IPv6 traffic selector per [RFC6088] beginning with 14 flag bits ("A-N"); when TS Format encodes any other value the Traffic Selector block is skipped and processing resumes beginning with the next Traffic Selector block (note that future specifications may define new TS Formats).

* The Traffic Selector block elements then follow an 18-bit Reserved field and encode the information corresponding to any set flag bit(s) in order the same as specified in [RFC6088].

10.2.13. Geo Coordinates

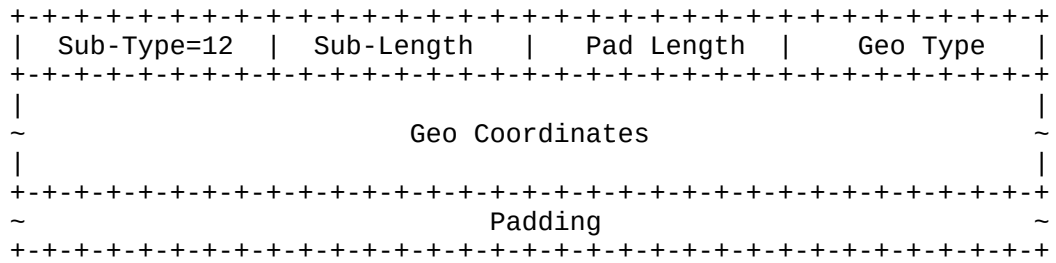

Figure 31: Geo Coordinates

* Sub-Type is set to 12. If multiple instances with different Geo Types appear in the same OMNI option all are processed.

* Sub-Length is set to the number of 8 octet units in the sub-option, including the Sub-Option Data.

* Pad Length is the length in octets of the trailing padding.

* Geo Type is a 1 octet field that encodes a type designator that determines the format and contents of the Geo Coordinates field that follows. The following types are currently defined:

  - 0 - NULL, i.e., the Geo Coordinates field is zero-length.

* Geo Coordinates is a type-specific format field of length up to the remaining available space for this OMNI option. Padding is added if necessary according to Geo Type to cause the option to end on an 8 octet boundary.

* New Geo Coordinate formats to be specified in future documents and may include attributes such as latitude/longitude, altitude, heading, speed, etc.

10.2.14. PIM-SM Message

The Protocol Independent Multicast - Sparse Mode (PIM-SM) Message sub-option may be included in the OMNI options of control messages. The PIM-SM message sub-option is formatted per Section 4.9 of [RFC7761] and as shown in Figure 32:

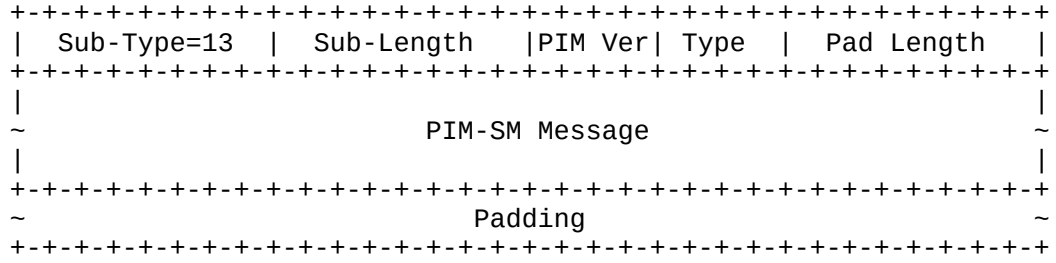

Figure 32: PIM-SM Message Option Format

* Sub-Type is set to 13. If multiple instances appear in a single OMNI option all are processed.

* Sub-Length is set to the number of 8 octet units in the sub-option, and Pad Length indicates the length of the trailing padding. The length of the entire PIM-SM message is therefore limited by the remaining available space for this OMNI option.

* The PIM-SM message is coded exactly as specified in Section 4.9 of [RFC7761], except that the Checksum field is omitted since message integrity is already assured by the OMNI option checksum. The "PIM Ver" field encodes the value 2, and the "Type" field encodes the PIM message type. (See Section 4.9 of [RFC7761] for a list of PIM-SM message types and formats.)

10.2.15. Fragmentation Report (FRAGREP)

Fragmentation Report (FRAGREP) sub-options may be included in the OMNI options of unsolicited IPv6 ND control messages sent from an OAL destination to an OAL source on behalf of a specific flow. The message is formatted and processed the same as specified for the Fragmentation Report option in [I-D.templin-6man-ipid-ext2].

The message consists of the 20-bit Flow Label value for the source's flow, followed by the 11 most significant bits of the 16-bit Maximum Receive Unit (MRU) for this flow followed by a (L)oss indication. The MRU field is then followed by an Identification for the specific packet from the OAL source that triggered the flow plus an optional Bitmap marking the ordinal positions of individual fragments received and missing.

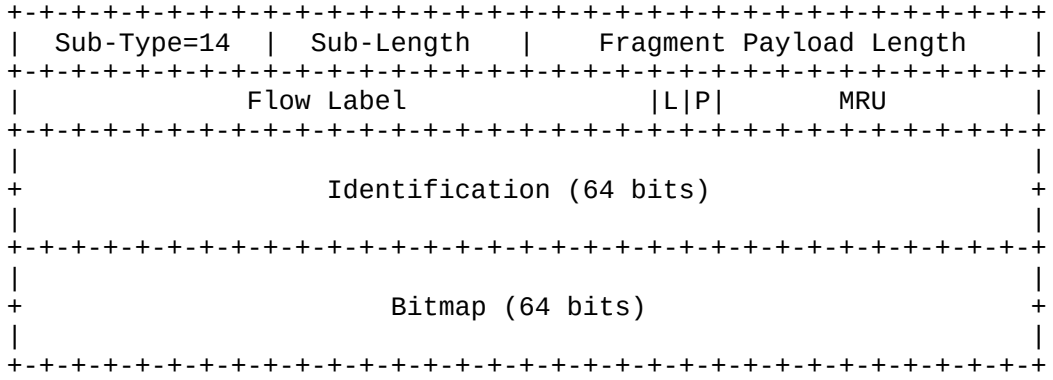

Figure 33: Fragmentation Report (FRAGREP)

* Sub-Type is set to 14. If multiple instances appear in the same OMNI option all are processed. Sub-Length is set to 3 if a Bitmap field is included; otherwise set to 2.

* Fragment Payload Length is the payload length of the invoking fragment beyond the (Extended) Fragment Header.

* Flow Label, L, P and MRU are 4 octets that include the same information as for the Fragmentation Report option in [I-D.templin-6man-ipid-ext2].

* Identification includes the 8 octet Identification value found in a received OAL fragment.

* Bitmap (optional) includes a 64-bit checklist of up to 64 ordinal fragments for this Identification, with each bit set to 1 for a fragment received or 0 for a fragment corrupted, lost or still in transit. For example, for a 20-fragment OAL packet with ordinal fragments #3, #10, #13 and #17 missing or corrupted and all other fragments received or still in transit, Bitmap(i) encodes the following:

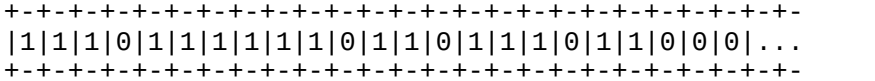

Figure 34

10.2.16. Proxy/Server Control

OMNI Clients include a Proxy/Server Control sub-option in RS messages when they associate with a current FHS and/or MAP Proxy/Server and/or need to send a departure indication to an old FHS and/or MAP Proxy/Server. When the FHS Proxy/Server forwards an RS to a different Proxy/Server acting as the MAP, the MAP also echoes the Proxy/Server Control sub-option in the responsive RA message.

Proxy/Servers also include the Proxy/Server Control sub-option in control messages sent as departure signals to departed FHS/MAP Proxy/Servers as specified in [I-D.templin-6man-aero3].

The Proxy/Server Control sub-option is formatted as shown below:

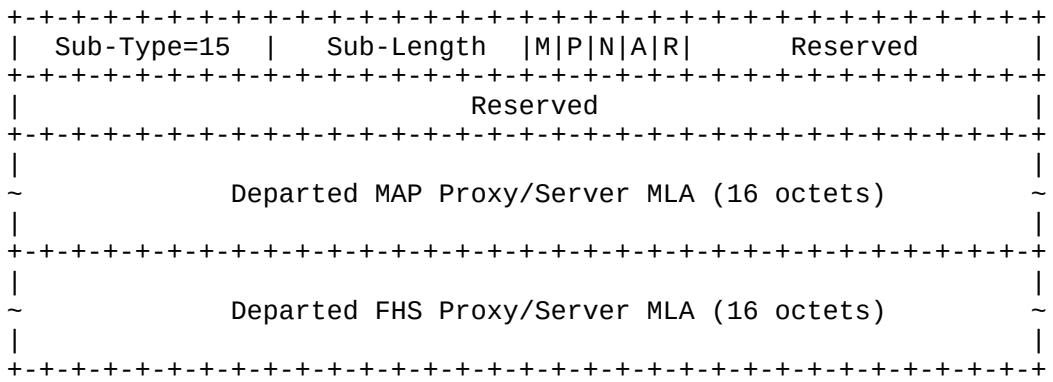

Figure 35: Proxy/Server Control

* Sub-Type is set to 15. If multiple instances appear the first instance is processed and all others are ignored. Sub-Length is set to 5 if departed Proxy/Server MLAs are included; otherwise, set to 1.

* Sub-Option Data contains 5 functional flags followed by 11 Reserved flags followed by 4 Reserved octets followed optionally by departed Proxy/Server MLAs.

* Clients set the (M)ap flag if the message must be forwarded to and/or processed by a MAP Proxy/Server.

* Clients set the (P)roxy flag to 0, and the FHS Proxy/Server resets the P flag to 1 as it forwards an RS message to a MAP Proxy/Server.

* Clients set the (N)eighbor Unreachability Detection (NUD), (A)ddress Resolution Responder (ARR) and (R)eport (RPT) flags in RS messages to control the operation of their FHS/MAP Proxy/Servers as discussed in Section 14.

* Clients finally include departed Proxy/Server MLAs if the Client has departed from a former FHS and/or MAP Proxy/Server. The departed FHS/MAP Proxy/Server address is set to an MLA for a departure; otherwise, set to ::/128.

10.2.17. Prefix Information Option (PIO)

OMNI Proxy/Servers include Prefix Information Option (PIO) sub-options in RA messages used to convey adaptation layer addressing information. The format corresponds to the PIO specified in [RFC4861] and [RFC9762] as shown below:

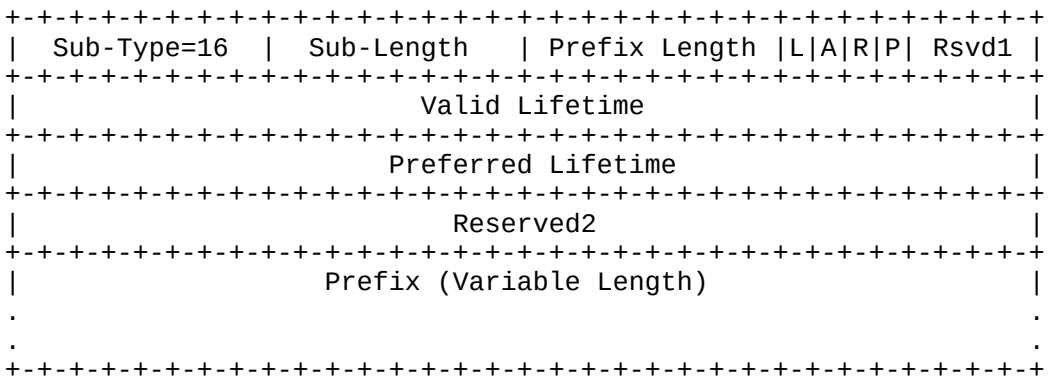

Figure 36: Prefix Information Option (PIO)

* Sub-Type is set to 16. If multiple instances appear in the same OMNI option, all are processed. Sub-Length is set to the number of 8 octet units in the sub-option.

* Sub-Option Data contains the same information as described in Section 4.6.2 of [RFC4861] and as updated by [RFC9762] with the exception that the included prefix is variable length as described in Section 2.3 of [RFC4191]. The Prefix field is 0, 8, or 16 octets according to Sub-Length and encodes an IPv6 prefix of length indicated by Prefix Length.

10.2.18. Route Information Option (RIO)

OMNI nodes include Route Information Option (RIO) sub-options in NS/NA messages used for Address Resolution. The format is the same as specified in [RFC4191] as shown below:

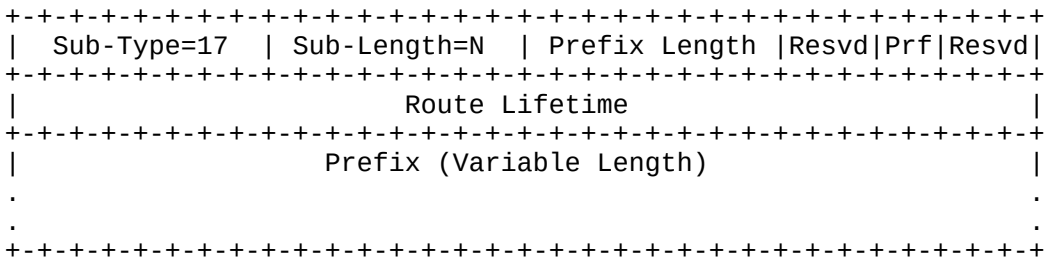

Figure 37: Route Information Option (RIO)

* Sub-Type is set to 17. If multiple instances appear in the same OMNI option, all are processed. Sub-Length is set to the number of 8 octet units in the sub-option.

* Sub-Option Data contains the same information as described in Section 2.3 of [RFC4191], where the Prefix field is 0, 8, or 16 octets according to Sub-Length.

10.2.19. DHCPv6 Message

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) sub-option may be included in the OMNI options of Client RS messages and Proxy/Server RA messages. The DHCPv6 sub-option is formatted per Section 8 of [RFC8415] as shown in Figure 38:

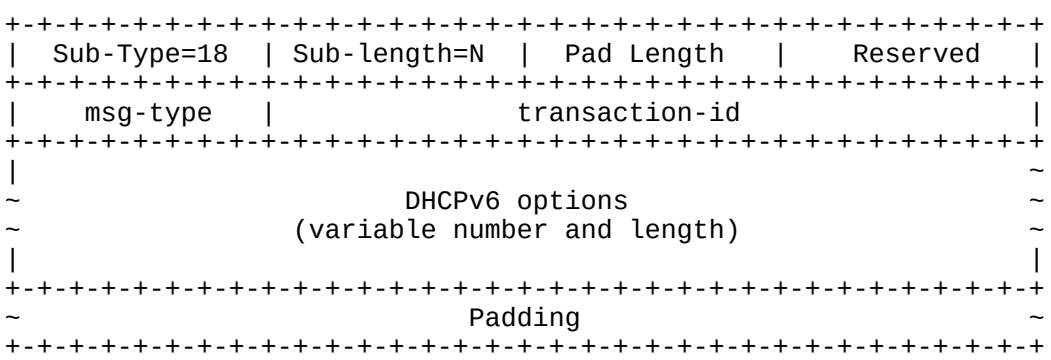

Figure 38: DHCPv6 Message

* Sub-Type is set to 18. At most 1 instance may appear in a single OMNI option. If multiple instances appear, the first is processed and all others are ignored.

* Sub-Length encodes the number of 8 octet units in the sub-option, and Pad Length encodes the number of trailing padding octets. Reserved is a 1 octet field set to 0 on transmission and ignored on reception.

* 'msg-type', 'transaction-id' and 'DHCPv6 options' are coded according to [RFC8415]. A Client's DHCPv6 message sub-option in an RS message is copied from a network layer DHCPv6 solicitation, and a Proxy/Server's DHCPv6 message sub-option in the RA message reply is copied from the DHCPv6 server reply. The Client and DHCPv6 server include their MLAs as DUID values as discussed in Section 10.2.9.

11. Address Mapping - Multicast

The multicast address mapping of the native underlay interface applies. The Client mobile router also serves as an IGMP/MLD Proxy for its EUNs and/or hosted applications per [RFC4605].

The Client uses Multicast Listener Discovery (MLDv2) [RFC9777] to coordinate with Proxy/Servers, and underlay network elements use MLD snooping [RFC4541]. The Client can also employ multicast routing protocols to coordinate with network-based multicast sources as specified in [I-D.templin-6man-aero3].

NS messages used for Duplicate Address Detection (DAD) include the unspecified address as the Source, the address being checked as the Target, and the solicited-node multicast address of the target as the Destination [RFC4862]. Since only MLAs that are already managed for uniqueness are assigned to the OMNI interface, the interface discards all NS(DAD) messages generated by the network layer.

Since the OMNI link model is NBMA, OMNI links support other link-scoped multicasting through iterative unicast transmissions to individual multicast group members (i.e., unicast/multicast emulation).

12. Multilink Conceptual Sending Algorithm

The Client's network layer selects the outbound OMNI interface according to SBM considerations when forwarding original IP packets from local or EUN applications to external correspondents. Each OMNI interface maintains a NLNC maintained the same as discussed in [RFC4861], but also includes ALNC state for multilink coordination.

For each original IP packet it forwards, the OMNI interface selects one or more source underlay interfaces based on PBM factors (e.g., traffic selectors, cost, performance, message size, etc.) and one or more target underlay interfaces for the neighbor based on Interface Attributes received in control messages (see: Section 10.2.10). Multilink forwarding may also direct carrier packet replication across multiple underlay interface pairs for increased reliability at the expense of duplication. The set of all Interface Attributes and Traffic Selectors received in control messages determines the multilink forwarding profile for selecting target underlay interfaces.

When the OMNI interface forwards an original IP packet over a selected source underlay interface, it first employs OAL encapsulation and fragmentation as discussed in Section 5, then performs underlay encapsulation as directed by the appropriate AFV. The OMNI interface also performs underlay encapsulation (following OAL encapsulation) when the nearest Proxy/Server is located multiple hops away as discussed in Section 14.2.

OMNI interface multilink service designers MUST observe the BCP guidance in Section 15 [RFC3819] in terms of implications for reordering when original IP packets from the same flow may be spread across multiple underlay interfaces having diverse properties.

12.1. Multiple OMNI Interfaces

Clients may connect to multiple independent OMNI links (and/or multiple independent physical links) within the same or different OMNI domains to support SBM. The Client configures a separate interface for each distinct link so that multiple interfaces (e.g., omni0, omni1, satcom2, etc.) are exposed to the network layer. Each OMNI interface is configured over a separate set of underlying interfaces and configures one or more OMNI link Subnet Router Anycast (SRA) addresses; the Client injects the corresponding SRA prefixes into the EUN routing system. Multiple distinct OMNI links can therefore be used to support fault tolerance, load balancing, reliability, hyperconnectivity, etc.

Applications in EUNs can use Segment Routing to select the desired OMNI interface based on SBM considerations. The application writes an OMNI link SRA address into the original IP packet's Destination Address, and writes the actual destination (along with any additional intermediate hops) into the Segment Routing Header. Standard IP routing directs the packet to the Client's mobile router entity, where the OMNI link SRA address identifies the correct OMNI interface for next hop forwarding. When the Client receives the packet, it replaces the IP Destination Address with the next Address found in the Segment Routing Header and forwards the message via the OMNI interface identified by the SRA address.

Note: The Client need not configure its OMNI interface indexes in one-to-one correspondence with the global OMNI Link-IDs configured for OMNI domain administration since the Client's indexes (i.e., omni0, omni1, omni2, etc.) are used only for its own local interface management.

12.2. Client-Proxy/Server Loop Prevention

After a Proxy/Server has registered an MNP for a Client (see: Section 14), the Proxy/Server will forward all original IP packets (or carrier packets) destined to an address within the MNP to the Client. Under normal circumstances, the Client will then forward the resulting original IP packet to the correct destination within its connected (downstream) EUNs.

If at some later time the Client loses state (e.g., after a reboot), it may begin returning original IP packets (or carrier packets) with destinations corresponding to its MNP to the Proxy/Server as its default router. The Proxy/Server therefore drops any original IP packets received from the Client with a Destination Address that corresponds to the Client's MNP, and drops any carrier packets with both Source and Destination Address corresponding to the same Client's MNP regardless of their origin.

Proxy/Servers support "hairpinning" for packets with MLA Source and Destination Addresses that would convey useful data from a source Client to a target Client both located in the same OMNI link segment. Proxy/Servers support this hairpinning according to [RFC6296], however direct forwarding between peer nodes within the same OMNI link segment is preferred whenever possible.

13. OAL Segment Routing

As discussed in Section 12.1, Segment Routing per [RFC8754] can be used by the network layer to select an OMNI link when there may be multiple alternatives. Once an OMNI link is selected, the OAL also applies Segment Routing internally to navigate multiple adaptation layer hops as discussed below.

The Segment Routing Header (SRH) is included as an extension to the OAL IPv6 encapsulation header and includes a Segment List with the MLAs of intermediate OAL hops. The SRH also includes a new TLV termed the AERO Flow Vector Index (AFVI) with the following format per Section 2.1 of [RFC8754]:

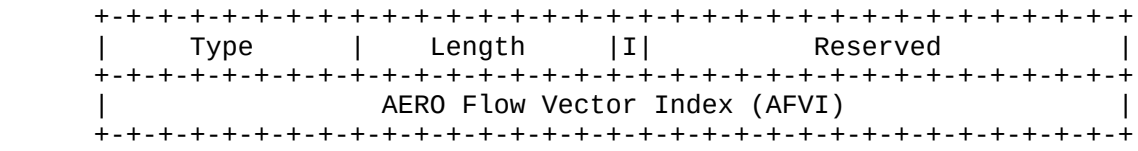

Figure 39: AERO Flow Vector Index (AFVI) TLV

In this format:

* Type is set to TBD4 (see: IANA considerations).

* Length is set to 6.

* I is an "Initialize" flag. When I is set to 1, the message is used to establish new AERO Flow Vector (AFV) state. When I is set to 0, the message follows existing AFV state. The I flag is followed by a 15 bit Reserved field which must be set to 0 (future specifications may define new values).

* AERO Flow Vector Index (AFVI) is a 32-bit field set by the OAL source for a specific flow and rewritten by each transit or endpoint OAL intermediate system on the path ending at the OAL destination. The special value 0 denotes "AFVI unspecified".

The FHS Client includes an OAL SRH to either discover and initialize the OAL intermediate systems on the forward path to the LHS Client or to follow a forward path previously established. When OAL header compression is applied the SRH is omitted and only the AFVI is transmitted.

For forward path discovery and initialization, the FHS Client includes its own MLA as Segment List[0], sets Segments Left and Last Entry to 0, sets I to 1 and sets AFVI to a locally-unique value that

can ideally be represented in 2 octets (i.e., a value less than 2**16). If no locally-unique values remain within the 2 octet limit, the OAL source instead selects a unique value that can be represented in 4 octets.

The FHS Client then includes the AFVI TLV in an SRH extension to the OAL header, calculates and includes an HMAC TLV if necessary and forwards the OAL IPv6 ND solicitation toward the FHS Proxy/Server. (The FHS Proxy/Server then either processes the message locally or forwards it on to the addressed LHS Client or MAP Proxy/Server.)

When an OAL intermediate system forwards the solicitation, it first verifies the HMAC then caches the AFVI and UNX address from the previous hop as AFV state for future forwarding purposes. If Segments Left is 0, the intermediate system then includes its own MLA as the new Segment List[0] and increments Last Entry. All intermediate systems rewrite the AFVI with their own unique value (i.e., the same as the OAL source had done) and recalculate the HMAC signature if necessary. The intermediate system then forwards the solicitation toward the FHS Proxy/Server and/or LHS Client.

When the I flag is set, each OAL intermediate system creates new AFV state if none previously existed or if it has stale AFV state older than (REACHABLE_TIME / 2) seconds. The OAL intermediate system refreshes the AFV state timer when it forwards a fresh secured control message for the flow. The OAL intermediate system retains any stale state up to REACHABLE_TIME seconds to accommodate any packets with stale AFVIs that may still be flowing.

Note that only FHS and LHS intermediate systems act as endpoints by adding their MLAs to the Segment List since all interdomain intermediate systems are globally addressable and can act as transits. For packets destined to targets in other OMNI link segments, the source adds the MLA of the LHS Proxy/Server as the penultimate and the MLA of the target as the ultimate Segment List entries.

The FHS Proxy/Server must also remove all FHS intermediate system MLAs from the Segment List when it forwards a message toward the LHS Proxy/Server and the LHS Proxy/Server must remove all LHS intermediate system MLAs when it forwards a message toward the FHS Proxy/server. The LHS Proxy/Server must then append its cached list of LHS intermediate system MLAs when it forwards a message toward an LHS Client and the FHS Proxy/Server must append its cached list of FHS intermediate system MLAs when it forwards a message toward an FHS Client. (The FHS/LHS Proxy/Server then sets Segments Left and Last Entry to non-zero values specific to their appended lists.)

Therefore, the FHS/LHS Proxy/Servers must cache the Segment Lists for their local domain Clients so that the only Segment List entries exposed in the interdomain area are those of the FHS/LHS/MAP Proxy/Servers themselves. Clients in turn must cache the Segment List including the FHS/LHS intermediate system addresses for future message transmissions to peers.

When the LHS Client receives a solicitation message, it returns a response that includes an OAL SRH with Segment List set to the reverse path list of n intermediate systems discovered in the solicitation and with Segments Left and Last Entry set to n. If the response will also establish AFV state in the reverse path, the LHS Client sets I to 1 and sets AFVI to a new locally-unique value for the previous OAL intermediate system; otherwise, the LHS Client sets both I and AFVI to 0. The LHS Client includes an HMAC signature if necessary, then forwards the response to the previous hop OAL intermediate system on the path to the FHS Client.

When an OAL intermediate node forwards the response, it first verifies the HMAC then (if I is 1) caches the AFVI and UNX address from the previous hop as AFV state for future forwarding purposes. FHS and LHS intermediate nodes then set the OAL destination to the next Segment List entry and decrement Segments Left. Intermediate nodes then rewrite AFVI with their own unique value and include a new HMAC signature if necessary. The node then forwards the advertisement to the next hop found in the Segment List.

The OAL peers (i.e., the FHS and LHS Clients) can then begin sending OAL packets with OCH headers that include the AFVI which each forwarding OAL intermediate system in the path can use to determine the next OAL hop. The network layer header, full OAL header and SRH therefore need not appear in these header-compressed OAL packets since the hop-by-hop AFVI values provide sufficient forwarding information. If either OAL peer needs to send a packet with full headers, it can include an OAL SRH with AFVI TLV with the I flag set to 0. OAL intermediate nodes will forward the packet based on matching cached AFVI values instead of initializing new AFVI values.

14. Router Discovery and Prefix Delegation

Clients that configure an OMNI interface also honor the specification for using RA messages to signal the availability of DHCPv6 Prefix Delegation (PD) [RFC9762]. When a Client initializes an OMNI interface, the adaptation layer within the interface delivers a locally-generated RA message to the network layer per [RFC4861]. The RA message includes the OMNI interface internal LLA as the Source Address, the Client’s MLA as the Destination Address, an SLLAO set to the OMNI interface’s internal link-layer address and standard RA

message body contents with the M, O flags both set to 1. The RA message also includes Prefix Information Option(s) with Prefix set to ::/0 or a more-specific prefix, with [A=0; P=1] and with L=0 to engage the "off-link" model or L=1 to engage the "on-link" model as discussed in Section 1. After the initial RA message delivery, the adaptation layer continuously delivers additional RAs before the Router Lifetime or any prefix lifetimes expire.

The receipt of these RA messages causes the network layer to regard prefix ranges as on/off-link over the OMNI interface and to install the OMNI interface internal LLA in the Default Router List. This will cause all packets to a new destination to invoke address resolution over the OMNI interface while regarding the local adaptation layer as a virtual default router. The receipt of the RA message will also cause the network layer to issue a DHCPv6 Solicit message for Prefix Delegation over the OMNI interface. The OMNI interface therefore maps the link-scoped multicast address "All_DHCP_Relay_Agents_and_Servers - (ff02::1:2)" [RFC8415] to the link-layer address of its internal virtual router function to support prefix delegation services.

Following the initial locally-generated RA and receipt of a DHCPv6 Solicit message for Prefix Delegation per [RFC9762] (or after a brief delay), Clients engage the OMNI link at the adaptation layer by sending OAL encapsulated RS messages with OMNI options to cause Proxy/Servers to process the message and respond. Each RS message is received by an FHS Proxy/Server, which may in turn forward a proxyed copy to a MAP Proxy/Server located in a local or remote SRT segment according to an included Proxy/Server Control OMNI sub-option. The MAP Proxy/Server then returns an OAL encapsulated RA message either directly to the Client or via the original FHS Proxy/Server acting as a proxy.

Clients send RS messages under the conditions specified in Section 6.3.7 of [RFC4861] which includes not only interface/link initialization conditions but also mobility factors. In particular, Clients may send RS messages when the OMNI interface or an underlay interface changes state, or when the Client moves to a new link and needs to discover addressing parameters for its new topological orientation. The Client’s RS/RA exchanges therefore maintain the most current OMNI link state information even across unanticipated mobility events.

To initiate Router Discovery and Address/Prefix Delegation, the Client’s adaptation layer uses the OMNI interface MLA as the IPv6 Source Address for RS messages it sends to candidate FHS Proxy/ Servers. The adaptation layer also includes its MLA as the OAL encapsulation source while also including OMNI authentication,

Interface Attributes and any other sub-options. (In particular, the adaptation layer includes a DHCPv6 message sub-option that encapsulates the Client's soliciting DHCPv6 Prefix Delegation request.) When the FHS Proxy/Server receives the RS, it can optionally consult an MLA registration service to determine whether the Client is authorized to use its claimed MLA and discards the RS message if authorization fails. Otherwise, the FHS Proxy/Server provides Router Discovery services for the Client per the remainder of this section.

To support Client to service coordination, the OMNI option provides flag bits in OMNI Proxy/Server Control sub-options as discussed in Section 10.2.16. Clients set or clear Proxy/Server control flags in RS messages as directives to the Mobility Service FHS/MAP Proxy/ Servers. Proxy/Servers interpret the flags as follows:

* When an FHS Proxy/Server forwards or processes an RS with the NUD flag set, it responds directly to future NS Neighbor Unreachability Detection (NUD) messages with the Client as the target by returning NA(NUD) replies; otherwise, it forwards NS(NUD) messages to the Client.

* When the MAP Proxy/Server receives an RS with the ARR flag set, it responds directly to future NS Address Resolution (NS(AR)) messages with the Client as the target by returning NA(AR) replies; otherwise, it forwards NS(AR) messages to the Client.

* When the MAP Proxy/Server receives an RS with the RPT flag set, it maintains a Report List of recent NS(AR) message sources for the source or target Client and sends unsolicited IPv6 ND control messages to all list members if any aspects of the Client's underlay interfaces change.

Mobility Service Proxy/Servers function according to the NUD, ARR and RPT flag settings received in the most recent RS message to support dynamic Client updates.

Clients and FHS Proxy/Servers include an authentication signature as an OMNI sub-option in their RS/RA exchanges when necessary but always include a valid OAL Checksum. Clients and Proxy/Servers use the information included in RS/RA messages to establish NCE state and OMNI link autoconfiguration information as discussed in this section.

For each underlay interface, the Client sends RS messages with OMNI options to coordinate with potentially different FHS Proxy/Servers for each interface but normally only with one MAP Proxy/Server at a given time. All Proxy/Servers accept original IP packets addressed to their MLAs or link-scoped multicast, OAL packets addressed to

their anycast/unicast MLA and carrier packets addressed to their anycast/unicast UNX addresses. The Client typically selects one MAP Proxy/Server among any of its FHS Proxy/Servers, but may instead select any other Proxy/Server on the OMNI link to serve as the MAP.

Example UNX address discovery methods appear in [RFC5214] and include data link login parameters, name service lookups, static configuration, a DHCP option, a static "hosts" file, etc. In the absence of other information, the Client can resolve the DNS Fully-Qualified Domain Name (FQDN) "linkupnetworks.[domainname]" where "linkupnetworks" is a constant text string and "[domainname]" is a DNS suffix for the OMNI link (e.g., "example.com"). The name resolution will return a set of DNS resource records to populate a Potential Router List (PRL) that contains addresses of Proxy/Servers for the local OMNI link segment. When the underlay *NET does not support standard unicast server-based name resolution [RFC1035] the Client can engage a multicast service such as mDNS [RFC6762] within the local OMNI link segment.

Each FHS Proxy/Server configures an MLA then advertises its UNX address(es) for discovery as above. Each FHS Proxy/Server may service one or more of a Client’s underlay interfaces which each become associated with the Proxy/Server’s MLA. The ifIndex included in the Neighbor Synchronization sub-option is used to determine which Client underlay interface is selected.

Clients configure OMNI interfaces that observe the properties discussed in previous sections. The OMNI interface and its underlay interfaces are said to be in either the "UP" or "DOWN" state according to administrative actions in conjunction with the interface connectivity status. An OMNI interface transitions to UP/DOWN through administrative action and/or through underlay interface state transitions. When a first underlay interface transitions to UP, the OMNI interface also transitions to UP. When all underlay interfaces transition to DOWN, the OMNI interface also transitions to DOWN.

When a Client OMNI interface transitions to UP, the interface returns a locally-generated RA to the network layer, and the network layer will issue a DHCPv6 Solicit message for Prefix Delegation. The OMNI interface then appends a single OMNI option trailer for each RS message while sending an interface-specific copy of the message over each underlay interface. These OMNI RS messages will register an initial set of underlay interfaces that are also UP and optionally also request an MNP delegation. The Client sends additional RS messages to refresh lifetimes and to register/deregister underlay interfaces as they transition to UP or DOWN. Guidelines for sending additional RS messages to generate corresponding RAs are found in Section 8.3.4 of [RFC5214], and are further extended to include proactive responses to mobility events.

The Client's OMNI interface sends initial RS messages over an UP underlay interface with source set to its MLA [RFC4861]. The Client further sets the RS Destination Address to either the MLA of a specific (MAP) Proxy/Server or the link-scoped All-Routers multicast address ff02::2 [RFC4291]. The Client's OMNI interface then appends an OMNI option per Section 10 with Proxy/Server Control sub-options with the MAP, NUD, ARR and RPT flags set or cleared as necessary. When the Client needs to coordinate with a MAP Proxy/Server other than the FHS Proxy/Server for a specific underlay interface, it sets the RS Destination Address to the MLA of the MAP.

If the Client will synchronize with this FHS Proxy/Server it also includes a Neighbor Synchronization sub-option with FHS ifIndex set to the ifIndex of its own underlay interface and with LHS ifIndex set to 0 (i.e., the default ifIndex configured by all Proxy/Servers). The Client also sets Sequence Number to a randomly-chosen 8 octet value, sets AFVI to a randomly-chosen initial value and sets the Flow Label in the IPv6 header to 0. The resulting exchange will establish symmetric Identification windows for the Client and FHS Proxy/Server. (Note that the Client may omit the Neighbor Synchronization sub-option if it only wants to test reachability for certain Proxy/Servers without establishing state.)

The Client next includes an Interface Attributes sub-option for the underlay interface with LHS-MLA set to ::/128. The Client then includes any other necessary OMNI sub-options such as Traffic Selectors, authentication, Timestamp, Nonce, etc. The OMNI interface finally sets or clears the Interface Attributes FMT-Forward and FMT-Mode bits according to its desired FHS Proxy/Server service model for this interface as described in Section 10.2.10.

The Client next prepares to forward the RS over the underlay interface using OAL encapsulation while including authentication sub-options if necessary followed by the OMNI option trailing fields.

The Client also includes a Segment Routing Header (SRH) extension to the OAL header with its own MLA as Segment List[0], as discussed in Section 13. The Client then sets the OAL Source Address to its own MLA and sets the OAL Destination Address to the known MLA of the FHS Proxy/Server, site-scoped All-Routers multicast (ff05::2) [RFC4291] or an anycast address. When the Client requires underlay encapsulation, it next includes the discovered FHS Proxy/Server UNX address or an anycast address as the underlay destination then forwards the resulting carrier packet into the underlay network. The Client also caches its RS message transmissions in the OAL to match against any received RA messages.

When an FHS Proxy/Server receives a carrier packet containing an RS it sets aside the underlay and OAL headers then verifies the checksum and authentication signature while also consulting an MLA registration service based on the Client's claimed certificate. If the RS message authenticity/integrity is verified, the FHS Proxy/Server then creates/updates an ALNCE indexed by the Client's MLA found in the RS Source Address. The FHS Proxy/Server then caches the OMNI Interface Attributes and any Traffic Selector sub-options while also caching the AFVI plus underlay (UDP/IP) and OAL Source/Destination Address information.

The FHS Proxy/Server next caches the RS Proxy/Server Control NUD flag and Neighbor Synchronization parameters if present (see: Section 10.1). If the RS includes a Proxy/Server Control sub-option with the M flag set and the RS is addressed either to itself or Multicast/Anycast it assumes the MAP role. If the RS includes an OMNI DHCPv6 Solicit sub-option, the FHS/MAP Proxy/Server uses the encapsulated message to coordinate an MNP Prefix Delegation exchange with the local DHCPv6 server. The FHS/MAP Proxy/Server then assumes the MAP role as a mobility service entry point for injecting the Client's MLA and MNP(s) into the OMNI link routing system. The FHS/MAP Proxy/Server then caches all of the delegated prefixes as L3 addressing information for this Client MLA, and caches the RS NUD, ARR and RPT flags to determine its role in processing NS(AR) messages and generating unsolicited IPv6 ND control messages (see: Section 10.1).

The FHS/MAP Proxy/Server then prepares to return an RA message directly to the Client by first populating the Cur Hop Limit, Flags, Router Lifetime, Reachable Time and Retrans Timer fields with values appropriate for the OMNI link. The RA message also includes a PIO that encodes the MSP and with (A=0; L=0; P=1) to cause the Client to regard all addresses covered by the MSP as reachable via the OMNI interface (while making no statement about on/off-link properties) and with per-Client Prefix Delegation enabled. The FHS/MAP Proxy/Server then sets the RA Source Address to its own MLA and sets the RA Destination Address to the RS Source Address.

The FHS/MAP Proxy/Server next includes an SRH with the RS Segments written into the Segment List as specified in Section 13. The FHS/MAP Proxy/Server then includes an OMNI option with a Neighbor Synchronization sub-option with responsive window synchronization information. The FHS/MAP Proxy/Server also includes a DHCPv6 Reply sub-option, then includes a copy of the Client's original Interface Attributes sub-option with its INET-facing interface information written in the SRT/FMT/LHS fields. The Proxy/Server sets the LHS-MLA field to its own MLA and sets LHS-UNX to the address it observes in the RS message underlay source address. If the Proxy/Server observes a different UNX address than the one supplied by the Client, it sets the NAT indication in FMT-Type.

The FHS/MAP Proxy/Server next sets or clears the FMT-Forward and FMT-Mode flags if necessary to convey its capabilities to the Client, noting that it should honor the Client's stated preferences for those parameters if possible or override otherwise. The FMT-Forward/Mode flags thereafter remain fixed unless and until a new RS/RA exchange establishes different values (see: Section 10.2.10 for further discussion). If the FHS/MAP Proxy/Server's Client-facing interface is different than its INET-facing interface, the Proxy/Server next includes a second Interface Attributes sub-option with ifIndex set to '0', with a unicast underlay address for its Client-facing interface in the LHS-UNX field and with its own MLA in the LHS-MLA.

The FHS/MAP Proxy/Server then includes any other necessary OMNI sub-options such as an authentication sub-option, Nonce, Timestamp, etc. The FHS/MAP Proxy/Server then calculates the authentication signature/checksum and performs OAL encapsulation while setting the OAL Source Address to its own MLA and Destination Address to the OAL Source Address that appeared in the RS. The FHS/MAP Proxy/Server then performs underlay encapsulation with Source and Destination address information reversed from the RS underlay information and returns the resulting carrier packet to the Client over the same underlay interface the RS arrived on.

When an FHS Proxy/Server receives an RS addressed to the MLA of a different MAP Proxy/Server, it acts as a proxy for the target MAP. The FHS Proxy/Server first processes the RS message locally the same as described above except that it does not process the DHCPv6 sub-option. The FHS Proxy/Server then sets the P flag in the Proxy/Server Control option, sets the OAL Source Address to its own MLA and sets the OAL Destination to the MLA of the MAP provided by the Client. The FHS Proxy/Server next creates or updates an ALNCE for the Client (i.e., based on the Client's MLA) and caches the OAL Source Address, Neighbor Synchronization and Interface Attributes information as above.

The FHS Proxy/Server then caches the RS SRH Segment List to include in the return RA message and writes the RS UNX address and its own MLA into the corresponding Interface Attributes fields. The FHS Proxy/Server next calculates and includes the OAL checksum then performs underlay encapsulation and sends the resulting carrier packet into the SRT secured spanning tree.

When the MAP Proxy/Server receives the carrier packet, it performs underlay decapsulation and OAL decapsulation to obtain the proxyed RS, verifies the checksum, then performs prefix delegation based on the Client's supplied DHCPv6 sub-option to obtain or update any MNPs for the Client. The MAP Proxy/Server then creates/updates an ALNCE indexed by the Client's MLA found in the RS Source Address and caches any state (including delegated MNPs, the ARR and RPT flags, Interface Attributes information and Traffic Selectors). The MAP Proxy/Server finally caches any delegated MNPs as L3 information for this Client and also injects them into the OMNI link routing protocol.

The MAP Proxy/Server then returns an OMNI encapsulated RA that echoes the Client's (proxyed) Interface Attributes and Proxy/Server Control sub-options and with any other RA parameters the same as specified for the FHS/MAP Proxy/Server case above while also including a DHCPv6 Reply sub-option with the prefix delegation transaction results. The MAP Proxy/Server sets the RA Source Address to its own MLA and sets the Destination Address to the RS Source Address (i.e., the MLA of the Client). The OMNI interface of the MAP Proxy/Server then sets the OAL Source Address to its own MLA and Destination Address to the MLA of the FHS Proxy/Server. The MAP Proxy/Server then calculates the OAL checksum and encapsulates the RA as an OAL packet. The MAP Proxy/Server finally performs underlay encapsulation and sends the resulting carrier packet into the secured spanning tree.

When the FHS Proxy/Server receives the carrier packet it performs underlay decapsulation, verifies the checksum then updates the OMNI interface ALNCE for the Client and creates/updates an ALNCE for the MAP. The FHS Proxy/Server then sets the P flag in the RA flags field

[RFC4389] and proxys the RA by changing the OAL source to its MLA and changing the OAL Destination Address to the MLA indicated by the cached FHS SRH node sequence discovered in the RS message.

The FHS Proxy/Server next includes a Neighbor Synchronization sub-option with responses to its cached solicitations from the Client. The FHS Proxy/Server also includes an Interface Attributes sub-option with ifIndex '0' and with its Client-facing interface unicast underlay address if necessary (see above) and includes authentication sub-options if necessary. The FHS Proxy/Server next includes a unique AFVI for this Client then calculates the OAL authentication signature and checksum. The FHS Proxy/Server then includes an SRH extension to the OAL header with an AFVI, with the MLAs of endpoint intermediate systems on the reverse path to the Client and with the OAL Destination Address adjusted accordingly. The FHS Proxy/Server finally performs underlay encapsulation with addresses taken from the Client's ALNCE and sends the resulting carrier packet via the same underlay interface over which the RS was received.

When the Client receives the carrier packet, it performs underlay decapsulation followed by OAL decapsulation to obtain the RA message. The Client next verifies the OAL checksum and authentication signature, then matches the RA with its previously-sent RS by comparing the RS Sequence Number with the RA Acknowledgement Number and also comparing the Nonce and/or Timestamp values. If the values match, the Client then creates/updates OMNI interface ALNCEs for both the MAP and FHS Proxy/Server and caches the information in the RA message. The Client next discovers the MLA associated with its interface over this FHS Proxy/Server by examining the proxyed Interface Attributes sub-option.

If the Client has multiple underlay interfaces, it creates additional FHS Proxy/Server ALNCEs as necessary when it receives RAs over those interfaces (noting that multiple of the Client's underlay interfaces may be serviced by the same or different FHS Proxy/Servers). If the RA message includes OMNI DHCPv6 Reply sub-option with the results of prefix delegation transactions, the Client returns the DHCPv6 Reply response to the network layer. This will cause the network layer to provision the delegated prefixes on downstream-facing links per [RFC9762].

For each underlay interface, the Client next caches the (filled-out) Interface Attributes for its own ifIndex including the SRT/FMT/LHS and Segment List information that it received in an RA message over that interface so that it can include them in future control messages to provide neighbors with accurate address resolution parameters. (If the message includes an Interface Attributes sub-option with ifIndex '0', the Client also caches UNX as the underlay network-local

unicast address of the FHS Proxy/Server via that underlay interface.) The Client then consults FMT-Type and the UNX address to determine whether there may be NATs on the path to the FHS Proxy/Server. The Client finally caches the Neighbor Synchronization responsive window synchronization parameters for use in future control message exchanges via this FHS Proxy/Server.

Following the initial exchange, the FHS Proxy/Server MAY later send additional periodic and/or event-driven unsolicited RA messages per [RFC4861]. (The unsolicited RAs may be initiated either by the FHS Proxy/Server itself or by the MAP via the FHS as a proxy.) The Client then continuously manages its underlay interfaces according to their states as follows:

* When an underlay interface transitions to UP, the Client sends an RS over the underlay interface with an OMNI option with sub-options as specified above.

* When an underlay interface transitions to DOWN, the Client sends unsolicited IPv6 ND control messages over any UP underlay interface with an OMNI option containing Interface Attributes sub-options for the DOWN underlay interface with ifMetric set to 'ffffffff'. The Client sends isolated unsolicited IPv6 ND control messages when reliability is not thought to be a concern (e.g., if redundant transmissions are sent on multiple underlay interfaces), or may instead send an IPv6 ND solicitation message to receive a solicited reply.

* When the Router Lifetime for the MAP Proxy/Server nears expiration, the Client sends an RS over any underlay interface to receive a fresh RA from the MAP. If no RA messages are received over a first underlay interface (i.e., after retrying), the Client marks the underlay interface as DOWN and should attempt to contact the MAP Proxy/Server via a different underlay interface. If the MAP Proxy/Server is unresponsive over additional underlay interfaces, the Client sends an RS message with Destination Address set to the MLA of another Proxy/Server which will then assume the MAP role.

* When all of a Client's underlay interfaces have transitioned to DOWN (or if a prefix delegation lifetime expires), the MAP Proxy/Server withdraws the Client's MLA and MNPs the same as if it had received a message with a release indication.

The Client is responsible for retrying each RS exchange up to MAX_RTR_SOLICITATIONS times separated by RTR_SOLICITATION_INTERVAL seconds until an RA is received. If no RA is received over an UP underlay interface (i.e., even after attempting to contact alternate

Proxy/Servers), the Client can either declare this underlay interface as DOWN or continue to use the interface to support any peer-to-peer local communications with peers located in the same *NET. When changing to a new FHS/MAP Proxy/Server, the Client also includes a Proxy/Server Control OMNI sub-option in new RS messages; the (new) FHS Proxy/Server will in turn send unsolicited IPv6 ND control messages to the departed FHS and/or MAP Proxy/Server to announce the Client's departure as specified in [I-D.templin-6man-aero3].

Note: IPv6 ND messaging is internally-generated by the adaptation layer or in response to a network layer event such as receipt of a DHCPv6 Solicit message for Prefix Delegation. Most RS/RA messaging occurs at the adaptation layer only, with only locally-generated RA and DHCPv6 Reply messages returned to the network layer. The adaptation layer therefore behaves as a singular IPv6 router from the perspective of the adaptation layer.

Note: The Router Lifetime value in RA messages indicates the time before which the Client must send another RS message over this underlay interface (e.g., 600 seconds), however that timescale may be significantly longer than the lifetime the MS has committed to retain the prefix registration (e.g., REACHABLE_TIME seconds). Proxy/ Servers are therefore responsible for updating MS state on a shorter timescale than the Client may be required to do on its own behalf.

Note: On certain multicast-capable underlay interfaces, Clients should send periodic unsolicited multicast NA messages and Proxy/ Servers should send periodic unsolicited multicast RA messages as "beacons" that can be heard by other nodes on the link. If a node fails to receive a beacon after a timeout value specific to the link, it can initiate Neighbor Unreachability Detection (NUD) exchanges to test reachability.

Note: Although the Client's FHS Proxy/Server is a first-hop segment node from its own perspective, the Client stores the Proxy/Server's FMT, SRT, and addresses as last-hop segment (LHS) information to supply to neighbors. This allows both the Client and MAP Proxy/ Server to supply the information to neighbors that will perceive it as LHS information on the return path to the Client.

Note: The MAP Proxy/Server injects Client MLA and MNPs into the OMNI link routing system by simply creating a route-to-interface forwarding table entry for MLA::/128 and MNP::/N via the OMNI interface. The dynamic routing protocol will notice the new entries and propagate the routes to its peers. If the MAP receives additional RS messages, it need not re-create the forwarding table entries (nor disturb the dynamic routing protocol) if the entries are already present. If the MAP ceases to receive RS messages from any

of the Client's interfaces, it removes the Client MLA and MNP(s) from the forwarding table (i.e., after a short delay) which also results in their removal from the routing system.

Note: If the Client's initial RS message includes an anycast underlay Destination Address, the FHS Proxy/Server returns the solicited RA using the same anycast address as the underlay source while including an Interface Attributes sub-option with ifIndex '0' and its true unicast address in the UNX. When the Client sends additional RS messages, it includes this FHS Proxy/Server unicast address as the underlay Destination Address and the FHS Proxy/Server returns the solicited RA using the same unicast address as the underlay Source Address. This will ensure that RS/RA exchanges are not impeded by any NATs on the path while avoiding long-term exposure of messages that use an anycast address as the source.

Note: Clients should set the NUD, ARR and RPT flags consistently in successive RS messages and only change those settings when an FHS/MAP Proxy/Server service profile update is necessary.

14.1. Client-Proxy/Server Window Synchronization

The RS/RA exchanges discussed above observe the principles specified in Section 6.7. Window synchronization is conducted between the Client and each FHS Proxy/Server used to contact the MAP Proxy/ Server, i.e., and not between the Client and the MAP. This is due to the fact that the MAP Proxy/Server is responsible only for forwarding messages via the secured spanning tree to FHS Proxy/Servers, and is not responsible for forwarding messages directly to the Client over unsecured networks.

When a Client sends an RS to perform window synchronization via a new FHS Proxy/Server, it includes an OMNI Neighbor Synchronization sub-option with window synchronization parameters with FHS ifIndex set to its own interface index, with LHS ifIndex set to 0, with the SYN flag set and ACK flag clear, and with an initial Sequence Number. The Client also includes an Interface Attributes sub-option then performs OAL encapsulation and underlay encapsulation and sends the resulting carrier packet to the FHS Proxy/Server. When the FHS Proxy/Server receives the carrier packet, it performs underlay decapsulation, then extracts the RS message and caches the Neighbor Synchronization parameters. In the process, the FHS Proxy/Server removes the Neighbor Synchronization sub-option itself, since the path to the MAP Proxy/Server is not included in window synchronization.

The FHS Proxy/Server then performs underlay encapsulation and sends the resulting carrier packet via the secured spanning tree to the MAP Proxy/Server, which updates the Client's Interface Attributes and

returns a unicast RA message. The MAP Proxy/Server performs OAL encapsulation followed by underlay encapsulation and sends the resulting carrier packet via the secured spanning tree to the FHS Proxy/Server. The FHS Proxy/Server then proxys the message as discussed in the previous section and includes a Neighbor Synchronization sub-option with responsive window synchronization information. The FHS Proxy/Server then forwards the message to the Client via OAL encapsulation which updates its window synchronization information for the FHS Proxy/Server as necessary.

Following the initial RS/RA-driven window synchronization, the Client can re-assert new windows with specific FHS Proxy/Servers by performing RS/RA exchanges between its own MLA and the MLAs of the FHS Proxy/Servers at any time without having to disturb the MAP. When the Client also needs to refresh MAP state, it can set the RS Destination Address to the MAP MLA and include an SRH to support FHS Proxy/Server RS forwarding.

This window synchronization is necessary only for MANET and INET Clients that must include authentication signatures with their IPv6 ND messages; Clients in secured ANETs can omit window synchronization. When Client-to-Proxy/Server window synchronization is used, subsequent control messages exchanged between peers include IPv6 Extended Fragment Headers in the OAL encapsulations with in-window Identification values to support message authentication. No header compression state is maintained by OAL intermediate systems, which only maintain state for per-flow data plane windows.

14.2. IP Multihop and IPv4-Only Networks

On some *NETs, a Client may be located multiple intermediate OAL hops away from the nearest OMNI link Proxy/Server. Clients in multihop networks perform route discovery through the application of an adaptation layer routing protocol (e.g., a MANET routing protocol over omnidirectional wireless interfaces, etc.). Example routing protocols optimized for MANET operations include OSPFv3 [RFC5340] with MANET Designated Router (OSPF-MDR) extensions [RFC5614], OLSRv2 [RFC7181], Babel [RFC8966], AODVv2 [I-D.perkins-manet-aodvv2] and others. Clients employ the routing protocol according to the link model found in [RFC5889] and subnet model articulated in [RFC5942]. For unique identification within the MANET, Clients use an MLA either directly as a Router ID or as an extended identifier for a shorter Router ID.

The Client configures the top-level MLA prefix (e.g., 2001:30::/28 [RFC9374]) on the OMNI interface and configures the MANET routing protocol to populate discovered MLA-specific routes in an alternate kernel routing table. The OAL then engages the Netlink socket

[NETLINK] (or an alternate operating system primitive) to monitor the table. When the MANET routing protocol adds, modifies or removes an MLA route the OAL adds/modifies/removes a corresponding route within the OMNI interface to allow MLA-addressed packets forwarded by the network layer to flow through the OMNI interface instead of directly via a MANET interface. The OMNI interface virtual router entity then engages OAL encapsulation and header compression before forwarding an OAL packet or fragment via the correct MANET underlay interface. The next hop's OMNI interface will then receive the packets from its MANET interface due to the OAL encapsulation port/protocol/ethertype.

MANETs can be compartmentalized internally with some nodes configured as simple Clients and others (that may have both "upstream" and "downstream" underlay interfaces) configured as cluster heads that act as Proxy/Clients. These cluster heads configure and listen to the same multicast and anycast addresses as for the downstream interfaces of Proxy/Servers in order to act as endpoint OAL intermediate node proxys for other downstream Clients. Clusters within clusters based on these cluster head Proxy/Clients can then be recursively nested to arbitrary depths as long as at least one ultimate Proxy/Client configures an upstream interface that can directly address a Proxy/Server with outside Internetwork connectivity.

A Client located potentially multiple OAL hops away from the nearest Proxy can optionally discover Proxy MLAs using name resolution services such as mDNS as discussed in Section 14, where multicast exchanges can be propagated by Simplified Multicast Forwarding (SMF) [RFC6621]. The Client next prepares an RS message, sets the Source Address to its MLA, and sets the Destination Address to link-scoped All-Routers multicast (ff02::2) or a discovered Proxy MLA. The OMNI interface then employs OAL encapsulation, sets the OAL Source Address to its MLA and sets the OAL Destination Address to the MLA of the Proxy, the site-scoped All-Routers multicast address (ff05::2) or the OMNI IPv6 anycast address.

For IPv6-enabled *NETs where the underlay interface observes the MANET properties discussed above, the Client injects its MLA into the IPv6 multihop routing system and forwards the RS message without further encapsulation. Otherwise, the Client encapsulates the message in UDP/IPv6 underlay headers with Source Address set to the underlay interface IPv6 address and Destination Address set to the discovered unicast or anycast address of a Proxy. The Client then forwards the message into the IPv6 multihop routing system which conveys it to the nearest Proxy.

For IPv4-only *NETs, the Client encapsulates the RS message in UDP/IPv4 underlay headers with Source Address set to the underlay interface IPv4 address and Destination Address set to the discovered unicast address of a Proxy/Server or the OMNI IPv4 anycast address. The Client then forwards the message into the IPv4 multihop routing system which conveys it to the nearest Proxy/Server advertising the corresponding IPv4 prefix. If the nearest Proxy/Server is too busy, it should forward (without Proxying) the OAL-encapsulated RS to another nearby Proxy/Server connected to the same IPv4 (multihop) network that configures the OMNI IPv6 anycast address. (In environments where reciprocal RS forwarding cannot be supported, the first Proxy/Server should instead return an RA based on its own MSP(s).)

When an OAL intermediate node that participates in the routing protocol receives the encapsulated RS, it appends its MLA to the RS SRH Segment List, caches the AFVI value and rewrites the SRH AFVI with a new unique value and forwards the message according to its OAL IPv6 forwarding table (note that an OAL intermediate system could be a fixed infrastructure element such as a roadside unit or another MANET/VANET Client). This process repeats iteratively over successive OAL intermediate nodes until the RS message is received by a penultimate OAL hop within single-hop communications range of a Proxy/Server, which forwards the message to the Proxy/Server final hop.

When a Proxy/Server that configures the OMNI IPv6 anycast address receives the message, it decapsulates the RS and assumes either the MAP or FHS role (in which case, it may forward the RS to a candidate MAP). The MAP/FHS Proxy/Server then prepares an RA message using the same addressing disciplines as discussed in Section 14 and forwards the RA either to the FHS Proxy/Server or directly to the Client.

When the MAP or FHS Proxy/Server forwards the RA to the Client, it encapsulates the message in an OAL header, includes an SRH copied from the RS as specified in Section 13 and includes underlay encapsulation headers if necessary. The Proxy/Server then forwards the message to an OAL node within communications range, which forwards the message according to the next OAL hop. The multihop forwarding process within the *NET continues repetitively until the message arrives at the original Client, which decapsulates and performs autoconfiguration the same as if it had received the RA directly from a Proxy/Server on the same physical link.

MANETs often include Clients that configure multiple interfaces, with downstream interfaces internal to the MANET and upstream interfaces connected to external *NETs. Such Clients can provide proxy services to enable router discovery for peer Clients that connect only

internally within the MANET. These Proxy/Clients first perform router discovery to associate with true Proxy/Servers located on upstream *NETs. The Proxy/Clients also subscribe to the site-scoped all-routers multicast group (i.e., ff05::2) and advertise reachability for the OMNI IPv6 anycast address over their MANET interfaces.

When a source Client sends an initial RS message seeking service, MANET routing will direct the RS to one or more nearby Proxy/Clients which in turn forward the RS to one or more upstream interface Proxys. Each such Proxy/Client writes its MLA as the final Segment List IPv6 address at the end of the RS SRH and also includes a new unique AFVI. The natural progression continues from innermost Proxy/Clients to outermost Proxy/Clients until the RS message reaches a Proxy/Server. By that time, the SRH Segment List will contain an ordered list of MLAs of all Proxy/Clients in the reverse path.

The MANET Proxy/Client model recursively extends to include arbitrarily many layers of nested MANET clusters between the source Client and external Proxy/Servers. When the source Client's first-hop Proxy/Client forwards an RS, it forwards to the next upstream Proxy/Client in succession toward the Proxy/Server while recording its MLA in the SRH as above. The progression continues until the RS reaches an ultimate upstream Proxy/Client that can directly contact a Proxy/Server via underlay encapsulation over an upstream *NET interface.

The Proxy/Server processes the RS and returns an RA while including its own MLA in the OAL Source Address and the MLA of the outermost Proxy/Client in the OAL Destination Address. The Proxy/Server also includes the RS SRH information as an SRH extension to the RA OAL header with the ordered Segment List of Proxy/Client MLAs plus a unique AFVI value. When the outermost Proxy/Client receives the RA, it forwards the message to the MLA of the next hop Proxy/Client in succession based on the SRH information until the message arrives at the source Client. The source Client can then update its ALNCE information based on the information returned by the Proxy/Server. The Client also retains the Segment List information in the ALNCE for inclusion in OAL SRH headers for future multihop forwarding purposes.

When the Proxy/Server returns an RA, each upstream Proxy/Client forwards the RA through the recursively descending chain of downstream Proxy/Clients on the path to the source Client. Each Proxy/Client rewrites the OAL Destination Address according to the SRH next hop MLA address for the next downstream Proxy/Client hop toward the source Client and also includes a new unique AFVI value.

Note that this service model applies equally for MANETs that have only Proxy/Client access to external *NET Proxy/Servers as well as those that have some mix of Proxy/Clients and true Proxy/Servers at the MANET border. True Proxy/Servers at the MANET border will service MANET Client router discovery requests the same as for any *NET, while external Proxy/Servers will discover potentially many MANET Clients all using the same UNX address belonging to a single Proxy/Client. This arrangement ensures that MANET-internal Clients are able to access external Internetworking services the same as for MANET border Clients that also have direct connections to external *NETs.

Note: When the RS message includes an anycast underlay encapsulation Destination Address, the FHS Proxy/Server must use the same anycast addresses as the underlay encapsulation Source Address to support forwarding of the RA message plus any initial data messages. The FHS Proxy/Server then sends the resulting carrier packets over any NATs on the path. When the outermost (Proxy/)Client receives the RA, it will discover the FHS Proxy/Server unicast underlay encapsulation address and can send future carrier packets using the unicast (instead of anycast) addresses to populate NAT state in the forward path. (If the Client does not have immediate data to send to the FHS Proxy/Server, it can instead send an OAL "bubble" - see: Section 6.11.) After the Client begins using unicast underlay encapsulation addresses in this way, the FHS Proxy/Server should also begin using the same unicast address in the reverse direction.

Note: When an OMNI interface configures an MLA, any nodes that forward an encapsulated RS message with the MLA as the OAL source must not consider the message as being specific to a particular OMNI link segment. MLAs can therefore also serve as the Source and Destination Addresses of unencapsulated IPv6 data communications within the local routing region; if the MLAs are injected into the local network routing protocol their prefix length must be set to 128 per [RFC5889].

15. Secure Redirection

If the *NET link model is multiple access, the FHS Proxy/Server is responsible for assuring that address duplication cannot corrupt the neighbor caches of other nodes on the link through the use of the DHCPv6 address delegation service. When the Client sends an RS message on a multiple access *NET, the Proxy/Server verifies that the Client is authorized to use the MLA Source Address and responds with an RA (or forwards the RS to the MAP) only if the Client is authorized.

After verifying Client authorization and returning an RA, the Proxy/Server MAY return IPv6 ND Redirect messages in response to subsequent data plane packet transmissions to direct Clients located on the same *NET to exchange OAL packets directly without transiting the Proxy/Server. The Redirect messages use MLAs instead of LLAs to uniquely identify peers on the link, and include Interface Attribute OMNI sub-options for address resolution purposes.

Following secure redirection, the Clients can exchange OAL packets according to their unicast underlay addresses discovered from the Redirect message instead of using the dogleg path through the Proxy/Server. In some *NETs, however, such direct communications may be undesirable and continued use of the dogleg path through the Proxy/Server may provide better performance. In that case, the Proxy/Server can refrain from sending Redirects, and/or Clients can ignore them.

16. Proxy/Server Resilience

*NETs SHOULD deploy Proxy/Servers in Virtual Router Redundancy Protocol (VRRP) [RFC5798] configurations so that service continuity is maintained even if one or more Proxy/Servers fail. Using VRRP, the Client is unaware which of the (redundant) FHS Proxy/Servers is currently providing service, and any service discontinuity will be limited to the failover time supported by VRRP. Widely deployed public domain implementations of VRRP are available.

Proxy/Servers SHOULD use high availability clustering services so that multiple redundant systems can provide coordinated response to failures. As with VRRP, widely deployed public domain implementations of high availability clustering services are available. Note that special-purpose and expensive dedicated hardware is not necessary, and public domain implementations can be used even between lightweight virtual machines in cloud deployments.

17. Detecting and Responding to Proxy/Server Failures

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

FHS Proxy/Servers perform proactive NUD for MAP Proxy/Servers for which there are currently active Clients. If a MAP Proxy/Server fails, the FHS Proxy/Server can quickly inform Clients of the outage by sending multicast RA messages. The FHS Proxy/Server sends RA messages to Clients with Source Address set to the GUA of the MAP, with Destination Address set to link-scoped All-Nodes multicast (ff02::1) [RFC4291] and with 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 have been using the (now defunct) MAP Proxy/Server will receive the RA messages.

18. OMNI Interfaces on Open Internetworks

Client OMNI interfaces configured over IPv6-enabled underlay interfaces on an open Internetwork without an OMNI-aware first-hop router receive IPv6 RA messages with no OMNI options, while OMNI interfaces configured over IPv4-only underlay interfaces receive no IPv6 RA messages at all (but may receive IPv4 RA messages per [RFC1256]). Client OMNI interfaces that receive RA messages with OMNI options configure addresses, on-link prefixes, etc. on the underlay interface that received the RA according to standard IPv6 ND and address resolution conventions [RFC4861][RFC4862]. Client OMNI interfaces configured over IPv4-only underlay interfaces configure IPv4 address information on the underlay interfaces using mechanisms such as DHCPv4 [RFC2131].

Client OMNI interfaces configured over underlay interfaces connected to open Internetworks can apply lower layer security services such as VPNs (e.g., IPsec tunnels) to connect to a Proxy/Server, or can establish a secured direct point-to-point link to the Proxy/Server through some other means (see: Section 4). In environments where lower layer security may be impractical or undesirable, Client OMNI interfaces can instead send control messages with OMNI sub-options that include authentication parameters.

OMNI interfaces use UDP/IP as underlay encapsulation headers for transmission over open Internetworks with UDP service port number 8060 for both IPv4 and IPv6 underlay interfaces. The OMNI interface submits original IP packets for OAL encapsulation, then encapsulates the resulting OAL fragments in UDP/IP underlay headers to form carrier packets. (The first 4 bits following the UDP header determine whether the OAL headers are uncompressed/compressed as discussed in Section 6.5.) The OMNI interface sets the UDP length to the encapsulated OAL fragment length and sets the IP length to an appropriate value at least as large as the UDP datagram.

When necessary, sources include an OMNI option with an authentication sub-option in control messages. Procedures for including OMNI authentication sub-options are discussed in Section 10.

After establishing a secured underlay link or preparing for UDP/IP encapsulation, OMNI interfaces send RS/RA messages for Client-Proxy/Server coordination (see: Section 14) and peer-to-peer control solicitation and response messages for multilink forwarding, route optimization, and mobility management (see: [I-D.templin-6man-aero3]). These control plane messages must be authenticated while other control and data plane messages are delivered the same as for ordinary best effort traffic with Source Address and/or Identification window-based data origin verification. Transport and higher layer protocol sessions over OMNI interfaces that connect over open Internetworks without an explicit underlay link security services should therefore employ security at their layers to ensure authentication, integrity and/or confidentiality.

Clients should avoid using INET Proxy/Servers as general-purpose routers for steady streams of carrier packets that do not require authentication. Clients should therefore perform route optimization to coordinate with other INET nodes that can provide forwarding services (or preferably coordinate with peer Clients directly) instead of burdening the Proxy/Server. Procedures for coordinating with peer Clients and discovering INET nodes that can provide better forwarding services are discussed in [I-D.templin-6man-aero3].

Clients that attempt to contact peers over INET underlay interfaces often encounter NATs in the path. OMNI interfaces accommodate NAT traversal using UDP/IP encapsulation and the mechanisms discussed in [I-D.templin-6man-aero3]. FHS Proxy/Servers include UNX information in an Interface Attributes sub-option in RA messages to allow Clients to detect the presence of NATs.

Note: Following the initial control message exchange, OMNI interfaces configured over INET underlay interfaces maintain neighbor relationships by transmitting periodic control messages with OMNI options that include authentication signatures.

Note: OMNI interfaces configured over INET underlay interfaces should employ the Identification window synchronization mechanisms specified in Section 6.7 in order to exclude spurious carrier packets that might otherwise clutter the reassembly cache. This is especially important in environments where carrier packet spoofing and/or corruption is a threat.

Note: NATs may be present on the path from a Client to its FHS Proxy/Server, but never on the path from the FHS Proxy/Server to the MAP where only INET and/or spanning tree hops occur. Therefore, the FHS Proxy/Server does not communicate Client origin information to the MAP where it would serve no purpose.

19. 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 Client may be willing to sacrifice a modicum of efficiency in order to have time-varying MNPs that can be changed occasionally to defeat adversarial tracking.

The OMNI link prefix delegation service allows Clients that desire time-varying MNPs to obtain short-lived prefixes to send RS messages with an OMNI option with DHCPv6 IA_PD sub-options. The Client would then be obligated to renumber its internal networks whenever its MNPs change. This should not present a challenge for Clients with automated network renumbering services, but may disrupt persistent sessions that would prefer to use a constant address.

20. IANA Considerations

The following IANA actions are requested in accordance with [RFC8126]. Both existing registries and new registries specific to OMNI are affected. Existing registries should be updated according to standard IANA practices. New registries should be created under a new registry group for "Overlay Multilink Network (OMNI) Interface".

20.1. Protocol Numbers

The IANA is instructed to allocate an Internet Protocol number TBD1 from the https://www.iana.org/assignments/protocol-numbers registry for the Overlay Multilink Network (OMNI) Interface as a non IPv6 Extension Header protocol. Guidance is found in [RFC5237] (registration procedure is IESG Approval or Standards Action).

20.2. IEEE 802 Numbers

During final publication stages, the IESG will be requested to procure an IEEE EtherType value TBD2 for OMNI according to the statement found at https://www.ietf.org/about/groups/iesg/statements/ethertypes/.

Following this procurement, the IANA is instructed to register the value TBD2 in the Ethertypes registry of the https://www.iana.org/assignments/ieee-802-numbers registry group for "Overlay Multilink Network (OMNI) Interface encapsulation on Ethernet networks". Guidance is found in [RFC7042] (registration procedure is Expert Review).

20.3. IPv4 Special-Purpose Address

The IANA is instructed to assign TBD3/N as an "OMNI IPv4 anycast" address/prefix in the https://www.iana.org/assignments/iana-ipv4-special-registry registry in a similar fashion as for [RFC3068]. The assignment also automatically provides the basis for an OMNI IPv6 subnet router anycast address configured as 2002:TBD3::. The IANA is requested to assist the author's efforts to obtain a TBD3/N public IPv4 prefix, whether through an RIR allocation, a delegation from the "Current Recovered IPv4 Pool" of the https://www.iana.org/assignments/ipv4-recovered-address-space registry or through an unspecified third party donation.

20.4. Segment Routing Header TLVs

The IANA is instructed to allocate a new Type value TBD4 with Description "AFVI TLV" and Reference [RFCXXXX] in the "Segment Routing Header TLVs" registry found in https://www.iana.org/assignments/ipv6-parameters (registration procedure is "IETF Review").

20.5. IANA OUI Ethernet Numbers

The IANA is instructed to allocate one Ethernet unicast address TBD5 (suggested value '00-52-14') in the "IANA Unicast 48-bit MAC Addresses" registry in the https://www.iana.org/assignments/ethernet-numbers registry group (registration procedure is Expert Review). The registration should appear as follows:

Addresses	Usage	Reference
00-52-14	Overlay Multilink Network (OMNI) Interface	[RFCXXXX]

Figure 40: IANA Unicast 48-bit MAC Addresses

20.6. Overlay Multilink Network (OMNI) Interface Registry Group

The IANA is instructed to create a new 'omni-interface' registry group for "Overlay Multilink Network (OMNI) Interface". The registry group must include the following new registries:

20.6.1. OMNI Option Sub-Types (New Registry)

The OMNI option defines a 1 octet Sub-Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI Option Sub-Type Values". Initial values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
0	NULL	[RFCXXXX]
1	CGA	[RFCXXXX]
2	RSA Signature	[RFCXXXX]
3	Timestamp	[RFCXXXX]
4	Nonce	[RFCXXXX]
5	Trust Anchor	[RFCXXXX]
6	Certificate	[RFCXXXX]
7	HMAC	[RFCXXXX]
8	Node Identification	[RFCXXXX]
9	Neighbor Synchronization	[RFCXXXX]
10	Interface Attributes	[RFCXXXX]
11	Traffic Selector	[RFCXXXX]
12	Geo Coordinates	[RFCXXXX]
13	PIM-SM Message	[RFCXXXX]
14	Fragmentation Report	[RFCXXXX]
15	Proxy/Server Control	[RFCXXXX]
16	Prefix Information Option	[RFCXXXX]
17	Route Information Option	[RFCXXXX]
18	DHCPv6 Message	[RFCXXXX]
19-252	Unassigned	[RFCXXXX]
253-254	Reserved for Experimentation	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 41: OMNI Option Sub-Type Values

20.6.2. OMNI Node Identification ID-Types (New Registry)

The OMNI Node Identification sub-option (see: Section 10.2.9) contains a 1 octet ID-Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI Node Identification ID-Type Values". Initial values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
0	MLA	[RFCXXXX]
1	UUID	[RFCXXXX]
2	Network Access Identifier	[RFCXXXX]
3	FQDN	[RFCXXXX]
4	IPv4 Address	[RFCXXXX]
5	Unassigned	[RFCXXXX]
6	IPv6 Address	[RFCXXXX]
7-252	Unassigned	[RFCXXXX]
253-254	Reserved for Experimental Use	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 42: OMNI Node Identification ID-Type Values

20.6.3. OMNI Geo Coordinates Types (New Registry)

The OMNI Geo Coordinates sub-option (see: Section 10.2.13) contains an 1 octet Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI Geo Coordinates Type Values". Initial values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
0	NULL	[RFCXXXX]
1-252	Unassigned	[RFCXXXX]
253-254	Reserved for Experimental Use	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 43: OMNI Geo Coordinates Type

20.7. Additional Considerations

IANA has assigned UDP port number "8060" for an earlier experimental version of AERO [RFC6706]. This document reclaims UDP port number "8060" for 'aero' as the service port for OMNI interface UDP/IP encapsulation. (Note that, although [RFC6706] is not widely implemented or deployed, any messages coded to that specification can be easily distinguished and ignored since they include an invalid ICMPv6 message type number '0'.) IANA is therefore instructed to update the reference for UDP port number "8060" from "RFC6706" to "RFCXXXX" (i.e., this document) while retaining the existing name 'aero'.

IANA has assigned a 4 octet Private Enterprise Number (PEN) code "45282" in the https://www.iana.org/assignments/enterprise-numbers registry. This document is the normative reference for using code

"45282" in DHCP Unique IDentifiers based on Enterprise Numbers ("DUID-EN for OMNI Interfaces") (see: Section 9). IANA is therefore instructed to change the enterprise designation for PEN code "45282" from "LinkUp Networks" to "Overlay Multilink Network (OMNI) Interface".

IANA has assigned the ifType code "301 - omni - Overlay Multilink Network (OMNI) Interface" in accordance with Section 6 of [RFC8892]. The registration appears under the IANA https://www.iana.org/assignments/smi-numbers registry group Interface Types (ifType)" registry, and the IANA is instructed to change the reference to [RFCXXXX] (i.e., this document).

No further IANA actions are required.

21. Security Considerations

Security considerations for IPv4 [RFC0791], IPv6 [RFC8200] and IPv6 Neighbor Discovery [RFC4861] apply. For end-to-end peer exchanges not fully protected by security associations, OMNI interfaces SHOULD use a modified version of SEcure Neighbor Discovery (SEND) or a Hashed Message Authentication Code (HMAC) per Section 10.1 as an adaptation layer service to ensure authentic exchanges. (Alternate OMNI interface authentication service types may be specified in future documents.) These services provide authentication for unsecured FHS and LHS *NETs, while intermediate hops are protected by the secured spanning tree (see below).

OMNI interfaces configured over secured ANET interfaces inherit the physical and/or link layer security services (i.e., protected spectrum and/or [MACSEC]) of the connected networks. OMNI interfaces configured over open *NET interfaces include message authentication codes as above; they can instead (or in addition) use symmetric securing services such as IPsec tunnels [RFC4301] or can by some other means establish a direct point-to-point link secured at lower layers.

OMNI link mobility services MUST support strong authentication for control plane messages and forwarding path integrity for data plane messages. In particular, the AERO service [I-D.templin-6man-aero3] constructs a secured spanning tree with Proxy/Servers as leaf nodes and secures the spanning tree links with network layer security services based on IPsec [RFC4301] with IKEv2 [RFC7296]. (Note that direct point-to-point links secured at lower layers can also be used instead of or in addition to network layer security.) Together, these services provide connectionless integrity and data origin authentication with optional protection against replays.

Control plane messages that affect the routing system or neighbor state either include authentication signatures or are constrained to travel only over secured spanning tree paths; in both cases, the messages are protected by network (and/or lower-layer) security. Other control and data plane messages can travel over unsecured route optimized paths that do not strictly follow the spanning tree, therefore end-to-end sessions should employ transport or higher layer security services (e.g., TLS/SSL [RFC8446], DTLS [RFC6347], etc.). Additionally, the OAL Identification value can provide a first level of data origin authentication to mitigate off-path spoofing.

Identity-based key verification infrastructure services such as iPSK may be necessary for verifying the identities claimed by Clients. This requirement should be harmonized with the manner in which identifiers such as MLAs are certified in a given operational environment.

Security considerations for specific access network interface types are covered under the corresponding IP-over-(foo) specification (e.g., [RFC2464], [RFC2492], etc.).

Security considerations for IPv6 fragmentation and reassembly are discussed in Section 6.13. In environments where spoofing is considered a threat, OMNI nodes SHOULD employ Identification window synchronization and OAL destinations SHOULD configure an (end-system-based) firewall.

22. 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.pre8 tagged on January 16, 2026. Future versions will be made available for public release.

23. Document Updates

This document suggests that the following could be updated through future IETF initiatives:

* [RFC1191]

* [RFC4443]

* [RFC8200]

* [RFC8201]

Updates can be through, e.g., standards action, the errata process, etc. as appropriate.

24. Acknowledgements

The first version of this document was prepared per the consensus decision at the 7th Conference of the International Civil Aviation Organization (ICAO) Working Group-I Mobility Subgroup on March 22, 2019. Consensus to take the document forward to the IETF was reached at the 9th Conference of the Mobility Subgroup on November 22, 2019. Attendees and contributors included: Guray Acar, Danny Bharj, Francois D'Humieres, Pavel Drasil, Nikos Fistas, Giovanni Garofolo, Bernhard Haindl, Vaughn Maiolla, Tom McParland, Victor Moreno, Madhu Niraula, Brent Phillips, Liviu Popescu, Jacky Pouzet, Aloke Roy, Greg Saccone, Robert Segers, Michal Skorepa, Michel Solery, Stephane Tamalet, Fred Templin, Jean-Marc Vacher, Bela Varkonyi, Tony Whyman, Fryderyk Wrobel and Dongsong Zeng.

The following individuals are acknowledged for their useful comments: Felipe Magno de Almeida, Amanda Baber, Scott Burleigh, Stuart Card, Donald Eastlake, Adrian Farrel, Michael Matyas, Robert Moskowitz, Madhu Niraula, Greg Saccone, Stephane Tamalet, Eliot Lear, Eduard Vasilenko, Eric Vyncke. Pavel Drasil, Zdenek Jaron and Michal Skorepa are especially recognized for their many helpful ideas and suggestions. Akash Agarwal, Madhuri Madhava Badgandi, Sean Dickson, Don Dillenburg, Joe Dudkowski, Vijayasarathy Rajagopalan, Ron Sackman, Bhargava Raman Sai Prakash and Katherine Tran are acknowledged for their hard work on the implementation and technical insights that led to improvements for the spec.

Discussions on the IETF 6man and atn mailing lists during the fall of 2020 suggested additional points to consider. The authors gratefully acknowledge the list members who contributed valuable insights through those discussions. Eric Vyncke and Erik Kline were the intarea ADs, while Bob Hinden and Ole Troan were the 6man WG chairs at the time the document was developed; they are all gratefully acknowledged for their many helpful insights. Many of the ideas in this document have further built on IETF experiences beginning in the 1990s, with insights from colleagues including Ron Bonica, Brian Carpenter, Ralph Droms, Tom Herbert, Bob Hinden, Christian Huitema, Thomas Narten, Dave Thaler, Joe Touch, Pascal Thubert, and many others who deserve recognition.

Early observations on IP fragmentation performance implications were noted in the 1986 Digital Equipment Corporation (DEC) "qe reset" investigation, where fragment bursts from NFS UDP traffic triggered

hardware resets resulting in communication failures. Jeff Chase, Fred Glover and Chet Juzsczak of the Ultrix Engineering Group led the investigation, and determined that setting a smaller NFS mount block size reduced the amount of fragmentation and suppressed the resets. Early observations on L2 media MTU issues were noted in the 1988 DEC FDDI investigation, where Raj Jain, KK Ramakrishnan and Kathy Wilde represented architectural considerations for FDDI networking in general including FDDI/Ethernet bridging. Jeff Mogul (who led the IETF Path MTU Discovery working group) and other DEC colleagues who supported these early investigations are also acknowledged.

Throughout the 1990's and into the 2000's, many colleagues supported and encouraged continuation of the work. Beginning with the DEC Project Sequoia effort at the University of California, Berkeley, then moving to the DEC research lab offices in Palo Alto CA, then to Sterling Software at the NASA Ames Research Center, then to SRI in Menlo Park, CA, then to Nokia in Mountain View, CA and finally to the Boeing Company in 2005 the work saw continuous advancement through the encouragement of many. Those who offered their support and encouragement are gratefully acknowledged.

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 Information Technology (BIT) Mobility Vision Lab (MVL) program.

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

Honoring life, liberty and the pursuit of happiness.

25. References

25.1. Normative References

[I-D.templin-6man-ipid-ext2]
Templin, F. and T. Herbert, "IPv6 Extended Fragment Header (EFH)", Work in Progress, Internet-Draft, draft-templin-6man-ipid-ext2-27, 12 January 2026, <https://datatracker.ietf.org/doc/html/draft-templin-6man-ipid-ext2-27>.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

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

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

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

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

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

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

[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>.

[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>.

[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>.

[RFC6088] Tsirtsis, G., Giarreta, G., Soliman, H., and N. Montavont, "Traffic Selectors for Flow Bindings", RFC 6088, DOI 10.17487/RFC6088, January 2011, <https://www.rfc-editor.org/info/rfc6088>.

[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>.

[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>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

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

[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>.

[RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/info/rfc9293>.

25.2. Informative References

[ATN] Maiolla, V., "The OMNI Interface - An IPv6 Air/Ground Interface for Civil Aviation, IETF Liaison Statement #1676, https://datatracker.ietf.org/liaison/1676/", 3 March 2020.

[ATN-IPS] "ICAO Document 9896 (Manual on the Aeronautical Telecommunication Network (ATN) using Internet Protocol Suite (IPS) Standards and Protocol), Draft Edition 3 (work-in-progress)", 10 December 2020.

[CRC] Jain, R., "Error Characteristics of Fiber Distributed Data Interface (FDDI), IEEE Transactions on Communications", August 1990.

[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.gont-dhcwg-dhcpv6-iids] Gont, F., "A Method for Generating Semantically Opaque IPv6 Interface Identifiers (IIDs) with Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", Work in Progress, Internet-Draft, draft-gont-dhcwg-dhcpv6-iids-00, 10 February 2025, <https://datatracker.ietf.org/doc/html/draft-gont-dhcwg-dhcpv6-iids-00>.

[I-D.herbert-ipv4-eh]
Herbert, T., "IPv4 Extension Headers and Flow Label", Work in Progress, Internet-Draft, draft-herbert-ipv4-eh-03, 22 February 2024, <https://datatracker.ietf.org/doc/html/draft-herbert-ipv4-eh-03>.

[I-D.ietf-6man-eh-limits]
Herbert, T., "Limits on Sending and Processing IPv6 Extension Headers", Work in Progress, Internet-Draft, draft-ietf-6man-eh-limits-19, 27 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-eh-limits-19>.

[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-25, 11 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-rfc6724-update-25>.

[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-tsvwg-udp-options]
Touch, J. D. and C. M. Heard, "Transport Options for UDP", Work in Progress, Internet-Draft, draft-ietf-tsvwg-udp-options-45, 16 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-udp-options-45>.

[I-D.ietf-v6ops-ula-usage-considerations]
Jiang, S., Liu, B., and N. Buraglio, "Considerations For Using Unique Local Addresses", Work in Progress, Internet-Draft, draft-ietf-v6ops-ula-usage-considerations-05, 11 December 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-ula-usage-considerations-05>.

[I-D.link-6man-gulla]
Linkova, J., "Using Prefix-Specific Link-Local Addresses to Improve SLAAC Robustness", Work in Progress, Internet-Draft, draft-link-6man-gulla-01, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-link-6man-gulla-01>.

[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-aodvv2-06, 20 June 2025, <https://datatracker.ietf.org/doc/html/draft-perkins-manet-aodvv2-06>.

[I-D.templin-6man-aero3]
Templin, F., "Automatic Extended Route Optimization (AERO)", Work in Progress, Internet-Draft, draft-templin-6man-aero3-52, 23 January 2026, <https://datatracker.ietf.org/doc/html/draft-templin-6man-aero3-52>.

[I-D.templin-6man-mla]
Templin, F., "IPv6 Addresses for Ad Hoc Networks", Work in Progress, Internet-Draft, draft-templin-6man-mla-30, 11 November 2025, <https://datatracker.ietf.org/doc/html/draft-templin-6man-mla-30>.

[I-D.templin-manet-inet]
Templin, F. and D. J. Jakubisin, "MANET Internetworking: Problem Statement and Gap Analysis", Work in Progress, Internet-Draft, draft-templin-manet-inet-02, 12 January 2026, <https://datatracker.ietf.org/doc/html/draft-templin-manet-inet-02>.

[IANA-CGA] Postel, J., "Cryptographically Generated Addresses (CGA) Message Type Name Space, https://www.iana.org/assignments/cga-message-types/cga-message-types.xhtml", 15 March 2023.

[IEEE802.1AX]
"Institute of Electrical and Electronics Engineers, Link Aggregation, IEEE Standard 802.1AX-2008, https://standards.ieee.org/ieee/802.1AX/6768/", 29 May 2020.

[IPV4] Postel, J., "IPv4 Address Space Registry, https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml", 14 December 2020.

[IPV6] Postel, J., "IPv6 Global Unicast Address Assignments, https://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml", 14 December 2020.

[MACSEC] Seaman, M., "IEEE Standard for Local and metropolitan area networks Media Access Control (MAC) Security (IEEE Std. 802.1AE), https://1.ieee802.org/security/802-1ae-2018/", September 2018.

[NETLINK] "https://en.wikipedia.org/wiki/Netlink", 1 January 2026.

[RFC0863] Postel, J., "Discard Protocol", STD 21, RFC 863, DOI 10.17487/RFC0863, May 1983, <https://www.rfc-editor.org/info/rfc863>.

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

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC1149] Waitzman, D., "Standard for the transmission of IP datagrams on avian carriers", RFC 1149, DOI 10.17487/RFC1149, April 1990, <https://www.rfc-editor.org/info/rfc1149>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

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

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[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>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC2492] Armitage, G., Schulter, P., and M. Jork, "IPv6 over ATM Networks", RFC 2492, DOI 10.17487/RFC2492, January 1999, <https://www.rfc-editor.org/info/rfc2492>.

[RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <https://www.rfc-editor.org/info/rfc2675>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <https://www.rfc-editor.org/info/rfc2863>.

[RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, DOI 10.17487/RFC2923, September 2000, <https://www.rfc-editor.org/info/rfc2923>.

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

[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>.

[RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, DOI 10.17487/RFC3068, June 2001, <https://www.rfc-editor.org/info/rfc3068>.

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

[RFC3366] Fairhurst, G. and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)", BCP 62, RFC 3366, DOI 10.17487/RFC3366, August 2002, <https://www.rfc-editor.org/info/rfc3366>.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <https://www.rfc-editor.org/info/rfc3692>.

[RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, DOI 10.17487/RFC3819, July 2004, <https://www.rfc-editor.org/info/rfc3819>.

[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>.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <https://www.rfc-editor.org/info/rfc4213>.

[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>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[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>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <https://www.rfc-editor.org/info/rfc4429>.

[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>.

[RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, DOI 10.17487/RFC4963, July 2007, <https://www.rfc-editor.org/info/rfc4963>.

[RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <https://www.rfc-editor.org/info/rfc5213>.

[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>.

[RFC5237] Arkko, J. and S. Bradner, "IANA Allocation Guidelines for the Protocol Field", BCP 37, RFC 5237, DOI 10.17487/RFC5237, February 2008, <https://www.rfc-editor.org/info/rfc5237>.

[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>.

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

[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>.

[RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <https://www.rfc-editor.org/info/rfc5798>.

[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>.

[RFC5889] Baccelli, E., Ed. and M. Townsley, Ed., "IP Addressing Model in Ad Hoc Networks", RFC 5889, DOI 10.17487/RFC5889, September 2010, <https://www.rfc-editor.org/info/rfc5889>.

[RFC5942] Singh, H., Beebee, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, DOI 10.17487/RFC5942, July 2010, <https://www.rfc-editor.org/info/rfc5942>.

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

[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>.

[RFC6214] Carpenter, B. and R. Hinden, "Adaptation of RFC 1149 for IPv6", RFC 6214, DOI 10.17487/RFC6214, April 2011, <https://www.rfc-editor.org/info/rfc6214>.

[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>.

[RFC6543] Gundavelli, S., "Reserved IPv6 Interface Identifier for Proxy Mobile IPv6", RFC 6543, DOI 10.17487/RFC6543, May 2012, <https://www.rfc-editor.org/info/rfc6543>.

[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>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[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>.

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

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

[RFC6980] Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, DOI 10.17487/RFC6980, August 2013, <https://www.rfc-editor.org/info/rfc6980>.

[RFC7042] Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", RFC 7042, DOI 10.17487/RFC7042, October 2013, <https://www.rfc-editor.org/info/rfc7042>.

[RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <https://www.rfc-editor.org/info/rfc7094>.

[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>.

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <https://www.rfc-editor.org/info/rfc7323>.

[RFC7343] Laganier, J. and F. Dupont, "An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers Version 2 (ORCHIDv2)", RFC 7343, DOI 10.17487/RFC7343, September 2014, <https://www.rfc-editor.org/info/rfc7343>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/info/rfc7542>.

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

[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>.

[RFC7847] Melia, T., Ed. and S. Gundavelli, Ed., "Logical-Interface Support for IP Hosts with Multi-Access Support", RFC 7847, DOI 10.17487/RFC7847, May 2016, <https://www.rfc-editor.org/info/rfc7847>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <https://www.rfc-editor.org/info/rfc8504>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

[RFC8892] Thaler, D. and D. Romascanu, "Guidelines and Registration Procedures for Interface Types and Tunnel Types", RFC 8892, DOI 10.17487/RFC8892, August 2020, <https://www.rfc-editor.org/info/rfc8892>.

[RFC8899] Fairhurst, G., Jones, T., T端xen, M., R端ngeler, I., and T. V旦lker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <https://www.rfc-editor.org/info/rfc8899>.

[RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <https://www.rfc-editor.org/info/rfc8900>.

[RFC8966] Chroboczek, J. and D. Schinazi, "The Babel Routing Protocol", RFC 8966, DOI 10.17487/RFC8966, January 2021, <https://www.rfc-editor.org/info/rfc8966>.

[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>.

[RFC9374] Moskowitz, R., Card, S., Wiethuechter, A., and A. Gurtov, "DRIP Entity Tag (DET) for Unmanned Aircraft System Remote ID (UAS RID)", RFC 9374, DOI 10.17487/RFC9374, March 2023, <https://www.rfc-editor.org/info/rfc9374>.

[RFC9562] Davis, K., Peabody, B., and P. Leach, "Universally Unique IDentifiers (UUIDs)", RFC 9562, DOI 10.17487/RFC9562, May 2024, <https://www.rfc-editor.org/info/rfc9562>.

[RFC9602] Krishnan, S., "Segment Routing over IPv6 (SRv6) Segment Identifiers in the IPv6 Addressing Architecture", RFC 9602, DOI 10.17487/RFC9602, October 2024, <https://www.rfc-editor.org/info/rfc9602>.

[RFC9663] Colitti, L., Linkova, J., Ed., and X. Ma, Ed., "Using DHCPv6 Prefix Delegation (DHCPv6-PD) to Allocate Unique IPv6 Prefixes per Client in Large Broadcast Networks", RFC 9663, DOI 10.17487/RFC9663, October 2024, <https://www.rfc-editor.org/info/rfc9663>.

[RFC9762] Colitti, L., Linkova, J., Ma, X., Ed., and D. Lamparter, "Using Router Advertisements to Signal the Availability of DHCPv6 Prefix Delegation to Clients", RFC 9762, DOI 10.17487/RFC9762, June 2025, <https://www.rfc-editor.org/info/rfc9762>.

[RFC9777] Haberman, B., Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", STD 101, RFC 9777, DOI 10.17487/RFC9777, March 2025, <https://www.rfc-editor.org/info/rfc9777>.

[TUNTAP] Krasnyansky, M., "Universal TUN/TAP device driver, https://docs.kernel.org/networking/tuntap.html", January 2000.

[VETH] Interfaces, K., "veth(4) - Linux manual page, https://man7.org/linux/man-pages/man4/veth.4.html", May 2025.

Appendix A. VDL Mode 2 Considerations

ICAO Doc 9776 is the "Technical Manual for VHF Data Link Mode 2" (VDLM2) that specifies an essential radio frequency data link service for aircraft and ground stations in worldwide civil aviation air traffic management. The VDLM2 link type is "multicast capable" [RFC4861], but with considerable differences from common multicast links such as Ethernet and IEEE 802.11.

First, the VDLM2 link data rate is only 31.5Kbps - multiple orders of magnitude less than most modern wireless networking gear. Second, due to the low available link bandwidth only VDLM2 ground stations (i.e., and not aircraft) are permitted to send broadcasts, and even so only as compact link layer "beacons". Third, aircraft employ the services of ground stations by performing unicast RS/RA exchanges upon receipt of beacons instead of listening for multicast RA messages and/or sending multicast RS messages.

This beacon-oriented unicast RS/RA approach is necessary to conserve the already-scarce available link bandwidth. Moreover, since the numbers of beaconing ground stations operating within a given spatial range must be kept as sparse as possible, it would not be feasible to have different classes of ground stations within the same region

observing different protocols. It is therefore highly desirable that all ground stations observe a common language of RS/RA as specified in this document.

Note that links of this nature may benefit from compression techniques that reduce the bandwidth necessary for conveying the same amount of data. The IETF lpwan working group is considering possible alternatives: [https://datatracker.ietf.org/wg/lpwan/documents].

Appendix B. Client-Proxy/Server Isolation Through Link-Layer Address Mapping

Per [RFC4861], IPv6 ND control messages may be sent to either a multicast or unicast link-scoped IPv6 Destination Address. However, IPv6 ND messaging should be coordinated between the Client and Proxy/Server only without invoking other nodes on the underlay network. This implies that Client-Proxy/Server control messaging should be isolated and not overheard by other nodes on the link.

To support Client-Proxy/Server isolation on some links, Proxy/Servers can maintain an OMNI-specific unicast link-layer address ("MSADDR"). For Ethernet-compatible links, this specification reserves one Ethernet unicast address TBD5 (see: IANA Considerations). For non-Ethernet statically-addressed links MSADDR is reserved per the assigned numbers authority for the link-layer addressing space. For still other links, MSADDR may be dynamically discovered through other means, e.g., link layer beacons.

Clients map the L3 addresses of all IPv6 ND messages they send (i.e., both multicast and unicast) to MSADDR instead of to an ordinary unicast or multicast link-layer address. In this way, all of the Client's IPv6 ND messages will be received by Proxy/Servers that are configured to accept carrier packets destined to MSADDR. Note that multiple Proxy/Servers on the link could be configured to accept carrier packets destined to MSADDR, e.g., as a basis for supporting redundancy.

Therefore, Proxy/Servers must accept and process carrier packets destined to MSADDR, while all other devices must not process carrier packets destined to MSADDR. This model has well-established operational experience in Proxy Mobile IPv6 (PMIP) [RFC5213][RFC6543].

Appendix C. IPv4 as an OAL Encapsulation Service

Throughout the document, IPv6 encapsulation per [RFC2473] is assumed as the OMNI Adaptation Layer (OAL) encapsulation service. At first glance, the minimum 40 octets needed for encapsulation may seem excessive however the full OAL encapsulation headers rarely appear over the wire due to effective header compression.

Still, the question may arise as to whether IPv4 encapsulation per [RFC2003] could be applied instead with OMNI encapsulation Type OMNI-IP4. After all, the IPv4 header is significantly smaller than even the smallest IPv6 header plus extensions. However, IPv4 provides only 32-bit addresses useful for OAL addressing whereas IPv6 provides 128-bits allowing for full MLA addresses along with their security and uniqueness properties.

IPv4 as an OAL encapsulation service may therefore be suitable for small networks where adaptation layer routers operate based on 32-bit router IDs deployed through well-managed assignments. However, IPv4 does not honor the Flow Label and IPv4 links could configure MTUs as small as 68 octets. An OAL IPv4 header plus extensions would also not be as compressible as for IPv6, therefore resulting in extra cost for carrying uncompressible IPv4 header information in the data plane.

Appendix D. Change Log

<< RFC Editor - remove prior to publication >>

Differences from earlier versions:

Draft -75 to -80

* Final version; future updates to appear in amendments draft.

* MLA routes now routable via the OMNI link.

* MLA routes now discovered within the OMNI interface.

* ICMPv6 error messages now appear as standalone control messages and not appended to an IPv6 ND message.

* Specified AFV state management procedures.

Draft -74 to -75

* OMNI option now includes "OMNI Length" and "Auth Offset".

Draft -70 to -74

* Support marking non IPv6 ND messages as control.

* OMNI interface LLA clarifications.

Draft -69 to -70

* Eliminated distinction between "gen/sec blocks" in OMNI option.

* Restored the DHCPv6 message sub-option.

* Included rules for processing IPv6 ND message checksums.

* Corrections to HMAC/SEND authentication and OMNI Checksum.

* Further clarification on Segment Routing.

Author's Address

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