Network Working Group F. Templin, Ed.
Internet-Draft The Boeing Company
Updates: rfc1191, rfc4443, rfc8201 (if A. Whyman
approved) MWA Ltd c/o Inmarsat Global Ltd
Intended status: Standards Track February 3, 2021
Expires: August 7, 2021

Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces
draft-templin-6man-omni-interface-75

Abstract

Mobile nodes (e.g., aircraft of various configurations, terrestrial vehicles, seagoing vessels, enterprise wireless devices, etc.) communicate with networked correspondents over multiple access network data links and configure mobile routers to connect end user networks. A multilink interface specification is therefore needed for coordination with the network-based mobility service. This document specifies the transmission of IP packets over Overlay Multilink Network (OMNI) Interfaces.

Table of Contents

1. Introduction

Mobile Nodes (MNs) (e.g., aircraft of various configurations, terrestrial vehicles, seagoing vessels, enterprise wireless devices, pedestrians with cellphones, etc.) often have multiple interface connections to wireless and/or wired-link data links used for communicating with networked correspondents. These data links may have diverse performance, cost and availability properties that can change dynamically according to mobility patterns, flight phases, proximity to infrastructure, etc. MNs coordinate their data links in a discipline known as "multilink", in which a single virtual interface is configured over the node's underlying interface connections to the data links.

The MN configures a virtual interface (termed the "Overlay Multilink Network (OMNI) interface") as a thin layer over the underlying interfaces. The OMNI interface is therefore the only interface abstraction exposed to the IP layer and behaves according to the Non-Broadcast, Multiple Access (NBMA) interface principle, while underlying interfaces appear as link layer communication channels in the architecture. The OMNI interface connects to a virtual overlay service known as the "OMNI link". The OMNI link spans one or more Internetworks that may include private-use infrastructures and/or the global public Internet itself.

Each MN receives a Mobile Network Prefix (MNP) for numbering downstream-attached End User Networks (EUNs) independently of the access network data links selected for data transport. The MN performs router discovery over the OMNI interface (i.e., similar to IPv6 customer edge routers [RFC7084]) and acts as a mobile router on behalf of its EUNs. The router discovery process is iterated over

each of the OMNI interface's underlying interfaces in order to register per-link parameters (see Section 14).

The OMNI interface provides a multilink nexus for exchanging inbound and outbound traffic via the correct underlying interface(s). The IP layer sees the OMNI interface as a point of connection to the OMNI link. Each OMNI link has one or more associated Mobility Service Prefixes (MSPs), which are typically IP Global Unicast Address (GUA) prefixes from which OMNI link MNPs are derived. If there are multiple OMNI links, the IPv6 layer will see multiple OMNI interfaces.

MNs may connect to multiple distinct OMNI links by configuring multiple OMNI interfaces, e.g., omni0, omni1, omni2, etc. Each OMNI interface is configured over a set of underlying interfaces and provides a nexus for Safety-Based Multilink (SBM) operation. Each OMNI SBM topology configures a common ULA prefix [ULA]::/48, and each OMNI link within the topology configures a unique 16-bit Subnet ID '*' to construct the sub-prefix [ULA*]::/64 (see: Section 8). The IP layer applies SBM routing to select an OMNI interface, which then applies Performance-Based Multilink (PBM) to select the correct underlying interface. Applications can apply Segment Routing [RFC8402] to select independent SBM topologies for fault tolerance.

The OMNI interface interacts with a network-based Mobility Service (MS) through IPv6 Neighbor Discovery (ND) control message exchanges [RFC4861]. The MS provides Mobility Service Endpoints (MSEs) that track MN movements and represent their MNPs in a global routing or mapping system.

Many OMNI use cases are currently under active consideration. In particular, 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 a document now in AD evaluation for RFC publication [I-D.ietf-ipwave-vehicular-networking]. 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 devices represent another large class of potential OMNI users.

This document specifies the transmission of IP packets and MN/MS control messages over OMNI interfaces. The OMNI interface supports either IP protocol version (i.e., IPv4 [RFC0791] or IPv6 [RFC8200])

as the network layer in the data plane, while using IPv6 ND messaging as the control plane independently of the data plane IP protocol(s). The OMNI Adaptation Layer (OAL) which operates as a mid-layer between L3 and L2 is based on IP-in-IPv6 encapsulation per [RFC2473] as discussed in the following sections. OMNI interfaces enable multilink, mobility, multihop and multicast services, with provisions for both Vehicle-to-Infrastructure (V2I) communications and Vehicle-to-Vehicle (V2V) communications outside the context of infrastructure.

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] specifications. Additionally, this document assumes the following IPv6 ND message types: Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS), Neighbor Advertisement (NA) and Redirect.

The Protocol Constants defined in Section 10 of [RFC4861] are used in their same format and meaning in this document. The terms "All-Routers multicast", "All-Nodes multicast" and "Subnet-Router anycast" are the same as defined in [RFC4291] (with Link-Local scope assumed).

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 following terms are defined within the scope of this document:

Mobile Node (MN)
: an end system with a mobile router having multiple distinct upstream data link connections that are grouped together in one or more logical units. The MN's data link connection parameters can change over time due to, e.g., node mobility, link quality, etc. The MN further connects a downstream-attached End User Network (EUN). The term MN used here is distinct from uses in other documents, and does not imply a particular mobility protocol.

End User Network (EUN)
: a simple or complex downstream-attached mobile network that travels with the MN as a single logical unit. The IP addresses assigned to EUN devices remain stable even if the MN's upstream data link connections change.

Mobility Service (MS)

a mobile routing service that tracks MN movements and ensures that MNs remain continuously reachable even across mobility events. Specific MS details are out of scope for this document.

Mobility Service Endpoint (MSE)
an entity in the MS (either singular or aggregate) that coordinates the mobility events of one or more MN.

Mobility Service Prefix (MSP)
an aggregated IP Global Unicast Address (GUA) prefix (e.g., 2001:db8::/32, 192.0.2.0/24, etc.) assigned to the OMNI link and from which more-specific Mobile Network Prefixes (MNPs) are delegated. OMNI link administrators typically obtain MSPs from an Internet address registry, however private-use prefixes can alternatively be used subject to certain limitations (see: Section 9). OMNI links that connect to the global Internet advertise their MSPs to their interdomain routing peers.

Mobile Network Prefix (MNP)
a longer IP prefix delegated from an MSP (e.g., 2001:db8:1000:2000::/56, 192.0.2.8/30, etc.) and assigned to a MN. MNs sub-delegate the MNP to devices located in EUNs.

Access Network (ANET)
a data link service network (e.g., an aviation radio access network, satellite service provider network, cellular operator network, wifi network, etc.) that connects MNs. Physical and/or data link level security is assumed, and sometimes referred to as "protected spectrum". Private enterprise networks and ground domain aviation service networks may provide multiple secured IP hops between the MN's point of connection and the nearest Access Router.

Access Router (AR)
a router in the ANET for connecting MNs to correspondents in outside Internetworks. The AR may be located on the same physical link as the MN, or may be located multiple IP hops away. In the latter case, the MN uses encapsulation to communicate with the AR as though it were on the same physical link.

ANET interface
a MN's attachment to a link in an ANET.

Internetwork (INET)
a connected network region with a coherent IP addressing plan that provides transit forwarding services between ANETs and nodes that connect directly to the open INET via unprotected media. No physical and/or data link level security is assumed, therefore

security must be applied by upper layers. The global public Internet itself is an example.

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

*NET
a "wildcard" term used when a given specification applies equally to both ANET and INET cases.

OMNI link
a Non-Broadcast, Multiple Access (NBMA) virtual overlay configured over one or more INETs and their connected ANETs. An OMNI link can comprise multiple INET 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.

OMNI interface
a node's attachment to an OMNI link, and configured over one or more underlying *NET interfaces. If there are multiple OMNI links in an OMNI domain, a separate OMNI interface is configured for each link.

OMNI Adaptation Layer (OAL)
an OMNI interface process whereby packets admitted into the interface are wrapped in a mid-layer IPv6 header and fragmented/reassembled if necessary to support the OMNI link Maximum Transmission Unit (MTU). The OAL is also responsible for generating MTU-related control messages as necessary, and for providing addressing context for spanning multiple segments of a bridged OMNI link.

OMNI Option
an IPv6 Neighbor Discovery option providing multilink parameters for the OMNI interface as specified in Section 11.

Mobile Network Prefix Link Local Address (MNP-LLA)
an IPv6 Link Local Address that embeds the most significant 64 bits of an MNP in the lower 64 bits of fe80::/64, as specified in Section 7.

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

Administrative Link Local Address (ADM-LLA)
an IPv6 Link Local Address that embeds a 32-bit administratively-assigned identification value in the lower 32 bits of fe80::/96, as specified in Section 7.

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

Multilink
an OMNI interface's manner of managing diverse underlying interface connections to data links as a single logical unit. The OMNI interface provides a single unified interface to upper layers, while underlying interface selections are performed on a per-packet basis considering factors such as DSCP, flow label, application policy, signal quality, cost, etc. Multilinking decisions are coordinated in both the outbound (i.e. MN to correspondent) and inbound (i.e., correspondent to MN) directions.

Multihop
an iterative relaying of IP packets between MNs over an OMNI underlying interface technology (such as omnidirectional wireless) without support of fixed infrastructure. Multihop services entail node-to-node relaying within a Mobile/Vehicular Ad-hoc Network (MANET/VANET) for MN-to-MN communications and/or for "range extension" where MNs within range of communications infrastructure elements provide forwarding services for other MNs.

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

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

underlying interface
a *NET interface over which an OMNI interface is configured. The OMNI interface is seen as a L3 interface by the IP layer, and each underlying interface is seen as a L2 interface by the OMNI interface. The underlying interface either connects directly to the physical communications media or coordinates with another node where the physical media is hosted.

Mobility Service Identification (MSID)
Each MSE and AR is assigned a unique 32-bit Identification (MSID) (see: Section 7). IDs are assigned according to MS-specific guidelines (e.g., see: [I-D.templin-intarea-6706bis]).

Safety-Based Multilink (SBM)
A means for ensuring fault tolerance through redundancy by connecting multiple affiliated OMNI interfaces to independent routing topologies (i.e., multiple independent OMNI links).

Performance Based Multilink (PBM)
A means for selecting underlying interface(s) for 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. Each OMNI domain consists of a set of affiliated OMNI links that all configure the same ::/48 ULA prefix with a unique 16-bit Subnet ID as discussed in Section 8.

3. Requirements

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.

OMNI links maintain a constant value "MAX_MSID" selected to provide MNs with an acceptable level of MSE redundancy while minimizing control message amplification. It is RECOMMENDED that MAX_MSID be set to the default value 5; if a different value is chosen, it should be set uniformly by all nodes on the OMNI link.

An implementation is not required to internally use the architectural constructs described here so long as its external behavior is consistent with that described in this document.

4. Overlay Multilink Network (OMNI) Interface Model

An OMNI interface is a MN virtual interface configured over one or more underlying interfaces, which may be physical (e.g., an aeronautical radio link) or virtual (e.g., an Internet or higher-layer "tunnel"). The MN receives a MNP from the MS, and coordinates with the MS through IPv6 ND message exchanges. The MN uses the MNP to construct a unique Link-Local Address (MNP-LLA) through the algorithmic derivation specified in Section 7 and assigns the LLA to the OMNI interface.

The OMNI interface architectural layering model is the same as in [RFC5558][RFC7847], and augmented as shown in Figure 1. The IP layer therefore sees the OMNI interface as a single L3 interface with multiple underlying interfaces that appear as L2 communication channels in the architecture.

```
                                    +----------------------------+
                                    |    Upper Layer Protocol    |
              Session-to-IP    +---->|                            |
              Address Binding  |     +----------------------------+
                               +---->|           IP (L3)          |
              IP Address       +---->|                            |
              Binding          |     +----------------------------+
                               +---->|       OMNI Interface       |
              Logical-to-      +---->|           (LLA)            |
              Physical         |     +----------------------------+
              Interface        +---->|  L2  |  L2  |       |  L2  |
              Binding                |(IF#1)|(IF#2)| ..... |(IF#n)|
                                     +------+------+       +------+
                                     |  L1  |  L1  |       |  L1  |
                                     |      |      |       |      |
                                     +------+------+       +------+
```

Figure 1: OMNI Interface Architectural Layering Model

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

o INET interfaces connect to an INET either natively or through one or several IPv4 Network Address Translators (NATs). Native INET interfaces have global IP addresses that are reachable from any INET correspondent. NATed INET interfaces typically have private IP addresses and connect to a private network behind one or more NATs that provide INET access.

o ANET interfaces connect to a protected ANET that is separated from the open INET by an AR acting as a proxy. The ANET interface may be either on the same L2 link segment as the AR, or separated from the AR by multiple IP hops.

o VPNed interfaces use security encapsulation over a *NET to a Virtual Private Network (VPN) gateway. Other than the link-layer encapsulation format, VPNed interfaces behave the same as for Direct interfaces.

o Direct (aka "point-to-point") interfaces connect directly to a peer without crossing any *NET paths. An example is a line-of-sight link between a remote pilot and an unmanned aircraft.

The OMNI virtual interface model gives rise to a number of opportunities:

- since MNP-LLAs are uniquely derived from an MNP, no Duplicate Address Detection (DAD) or Multicast Listener Discovery (MLD) messaging is necessary.

- since Temporary LLAs are statistically unique, they can be used without DAD for short-term purposes, e.g. until an MNP-LLA is obtained.

- *NET interfaces on the same L2 link segment as an AR do not require any L3 addresses (i.e., not even link-local) in environments where communications are coordinated entirely over the OMNI interface. (An alternative would be to also assign the same LLA to all *NET interfaces.)

- as underlying 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 underlying interfaces in this way allows them to be represented in a unified MS profile with provisions for mobility and multilink operations.

- exposing a single virtual interface abstraction to the IPv6 layer allows for multilink operation (including QoS based link selection, packet replication, load balancing, etc.) at L2 while still permitting L3 traffic shaping based on, e.g., DSCP, flow label, etc.

- the OMNI interface allows inter-INET traversal when nodes located in different INETs need to communicate with one another. This mode of operation would not be possible via direct communications over the underlying interfaces themselves.

- the OMNI Adaptation Layer (OAL) within the OMNI interface supports lossless and adaptive path MTU mitigations not available for communications directly over the underlying interfaces themselves. The OAL supports "packing" of multiple IP payload packets within a single OAL packet.

- L3 sees the OMNI interface as a point of connection to the OMNI link; if there are multiple OMNI links (i.e., multiple MS's), L3 will see 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.

Other opportunities are discussed in [RFC7847]. Note that even when the OMNI virtual interface is present, applications can still access underlying interfaces either through the network protocol stack using an Internet socket or directly using a raw socket. This allows for intra-network (or point-to-point) communications without invoking the OMNI interface and/or OAL. For example, when an IPv6 OMNI interface is configured over an underlying IPv4 interface, applications can still invoke IPv4 intra-network communications as long as the communicating endpoints are not subject to mobility dynamics. However, the opportunities discussed above are not available when the architectural layering is bypassed in this way.

Figure 2 depicts the architectural model for a MN with an attached EUN connecting to the MS via multiple independent *NETs. When an underlying interface becomes active, the MN's OMNI interface sends IPv6 ND messages without encapsulation if the first-hop Access Router (AR) is on the same underlying link; otherwise, the interface uses IP-in-IP encapsulation. The IPv6 ND messages traverse the ground domain *NETs until they reach an AR (AR#1, AR#2, ..., AR#n), which then coordinates with an INET Mobility Service Endpoint (MSE#1, MSE#2, ..., MSE#m) and returns an IPv6 ND message response to the MN. The Hop Limit in IPv6 ND messages is not decremented due to encapsulation; hence, the OMNI interface appears to be attached to an ordinary link.

```
                           +--------------+        (:::)-.
                           |      MN      |<-->.-(::EUN:::)
                           +--------------+      `-(::::)-'
                           |OMNI interface|
                           +----+----+----+
                  +--------|IF#1|IF#2|IF#n|------ +
                 /         +----+----+----+        \
                /                 |                 \
               /                  |                  \
              v                   v                   v
           (:::)-.              (:::)-.              (:::)-.
      .-(::*NET:::)        .-(::*NET:::)        .-(::*NET:::)
        `-(::::)-'           `-(::::)-'           `-(::::)-'
          +----+               +----+               +----+
    ...   |AR#1|  ..........   |AR#2|   .........   |AR#n|  ...
   .      +-|--+               +-|--+               +-|--+     .
   .        |                    |                    |
   .        v                    v                    v        .
   .             <-----  INET Encapsulation ----->             .
   .                                                           .
   .      +-----+               (:::)-.                        .
   .      |MSE#2|           .-(::::::::)          +-----+      .
   .      +-----+       .-(:::   INET  :::)-.     |MSE#m|      .
   .                  (:::::    Routing  ::::)    +-----+      .
   .                     `-(::: System :::)-'                  .
   .  +-----+                `-(:::::::-'                      .
   .  |MSE#1|          +-----+               +-----+           .
   .  +-----+          |MSE#3|               |MSE#4|           .
   .                   +-----+               +-----+           .
   .                                                           .
   .                                                           .
   .       <----- Worldwide Connected Internetwork ---->       .
    ...........................................................
```

Figure 2: MN/MS Coordination via Multiple *NETs

After the initial IPv6 ND message exchange, the MN (and/or any nodes on its attached EUNs) can send and receive IP data packets over the OMNI interface. OMNI interface multilink services will forward the packets via ARs in the correct underlying *NETs. The AR encapsulates the packets according to the capabilities provided by the MS and forwards them to the next hop within the worldwide connected Internetwork via optimal routes.

OMNI links span one or more underlying Internetwork via the OMNI Adaptation Layer (OAL) which is based on a mid-layer overlay encapsulation using [RFC2473]. Each OMNI link corresponds to a different overlay (differentiated by an address codepoint) which may

be carried over a completely separate underlying topology. Each MN can facilitate SBM by connecting to multiple OMNI links using a distinct OMNI interface for each link.

Note: OMNI interface underlying interfaces often connect directly to physical media on the local platform (e.g., 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 underlying interface) to the node hosting the physical media. The OMNI interface may also apply encapsulation at a layer above the underlying interface such that 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 underlying interface must monitor the node hosting the physical media (e.g., through periodic keepalives) so that it can convey up/down/status information to the OMNI interface.

5. The OMNI Adaptation Layer (OAL)

The OMNI interface observes the link nature of tunnels, including the Maximum Transmission Unit (MTU), Maximum Reassembly Unit (MRU) and the role of fragmentation and reassembly [I-D.ietf-intarea-tunnels]. The OMNI interface is configured over one or more underlying interfaces that may have diverse MTUs. OMNI interfaces accommodate MTU diversity through the use of the OMNI Adaptation Layer (OAL) as discussed in this section.

IPv6 underlying interfaces are REQUIRED to configure a minimum MTU of 1280 bytes and a minimum MRU of 1500 bytes [RFC8200]. Therefore, the minimum IPv6 path MTU is 1280 bytes 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 packets of at least 1280 bytes without generating an IPv6 Path MTU Discovery (PMTUD) Packet Too Big (PTB) message [RFC8201]. (Note: the source can apply "source fragmentation" for locally-generated IPv6 packets up to 1500 bytes and larger still if it if has a way to determine that the destination configures a larger MRU, but this does not affect the minimum IPv6 path MTU.)

IPv4 underlying interfaces are REQUIRED to configure a minimum MTU of 68 bytes [RFC0791] and a minimum MRU of 576 bytes [RFC0791][RFC1122]. Therefore, when the Don't Fragment (DF) bit in the IPv4 header is set to 0 the minimum IPv4 path MTU is 576 bytes since routers on the path support network fragmentation and the destination is required to reassemble at least that much. The "Don't Fragment" (DF) bit in the IPv4 encapsulation headers of packets sent over IPv4 underlying

interfaces therefore MUST be set to 0. (Note: even if the encapsulation source has a way to determine that the encapsulation destination configures an MRU larger than 576 bytes, it should not assume a larger minimum IPv4 path MTU without careful consideration of the issues discussed in Section 5.1.)

In network paths where IPv6/IPv4 protocol translation or IPv6-in-IPv4 encapsulation may be prevalent, it may be prudent for the OAL to always assume the IPv4 minimum path MTU (i.e., 576 bytes) regardless of the underlying interface IP protocol version. By always assuming the IPv4 minimum path MTU even for IPv6 underlying interfaces, the OAL may produce smaller fragments and additional header overhead but will always interoperate and never run the risk of presenting a destination interface with a packet that exceeds its MRU.

The OMNI interface configures both an MTU and MRU of 9180 bytes [RFC2492]; the size is therefore not a reflection of the underlying interface MTUs, but rather determines the largest packet the OMNI interface can forward or reassemble. The OMNI interface uses the OMNI Adaptation Layer (OAL) to admit packets from the network layer that are no larger than the OMNI interface MTU while generating ICMPv4 Fragmentation Needed [RFC1191] or ICMPv6 Path MTU Discovery (PMTUD) Packet Too Big (PTB) [RFC8201] messages as necessary. This document refers to both of these ICMPv4/ICMPv6 message types simply as "PTBs", and introduces a distinction between PTB "hard" and "soft" errors as discussed below.

For IPv4 packets with DF=0, the network layer performs IPv4 fragmentation if necessary then admits the packets/fragments into the OMNI interface; these fragments will be reassembled by the final destination. For IPv4 packets with DF=1 and IPv6 packets, the network layer admits the packet if it is no larger than the OMNI interface MTU; otherwise, it drops the packet and returns a PTB hard error message to the source.

For each admitted IP packet/fragment, the OMNI interface internally employs the OAL when necessary by encapsulating the inner IP packet/fragment in a mid-layer IPv6 header per [RFC2473] before adding any outer IP encapsulations. (The OAL does not decrement the inner IP Hop Limit/TTL during encapsulation since the insertion occurs at a layer below IP forwarding.) If the OAL packet will itself require fragmentation, the OMNI interface then calculates the 32-bit CRC over the entire mid-layer packet and writes the value in a trailing field at the end of the packet. Next, the OAL fragments this mid-layer IPv6 packet, forwards the fragments (using *NET encapsulation if necessary), and returns an internally-generated PTB soft error message (subject to rate limiting) if it deems the packet too large according to factors such as link performance characteristics,

reassembly congestion, etc. This ensures that the path MTU is adaptive and reflects the current path used for a given data flow.

The OAL operates with respect to both the minimum IPv6 and IPv4 path MTUs as follows:

o When an OMNI interface sends a packet toward a final destination via an ANET peer, it sends without OAL encapsulation if the packet (including any outer-layer ANET encapsulations) is no larger than the underlying interface MTU for on-link ANET peers or the minimum ANET path MTU for peers separated by multiple IP hops. Otherwise, the OAL inserts an IPv6 header per [RFC2473] with source address set to the node's own Unique-Local Address (ULA) (see: Section 8) and destination set to either the Administrative ULA (ADM-ULA) of the ANET peer or the Mobile Network Prefix ULA (MNP-ULA) corresponding to the final destination (see below). The OAL then calculates and appends the trailing 32-bit CRC, then uses IPv6 fragmentation to break the packet into a minimum number of non-overlapping fragments where the size of each non-final fragment (including both the OMNI and any outer-layer ANET encapsulations) is determined by the underlying interface MTU for on-link ANET peers or the minimum ANET path MTU for peers separated by multiple IP hops. The OAL then encapsulates the fragments in any ANET headers and sends them to the ANET peer, which either reassembles before forwarding if the OAL destination is its own ADM-ULA or forwards the fragments toward the final destination without first reassembling otherwise.

o When an OMNI interface sends a packet toward a final destination via an INET interface, it sends packets (including any outer-layer INET encapsulations) no larger than the minimum INET path MTU without OAL encapsulation if the destination is reached via an INET address within the same OMNI link segment. Otherwise, the OAL inserts an IPv6 header per [RFC2473] with source address set to the node's ULA, destination set to the ULA of the next hop OMNI node toward the final destination and (if necessary) with an OMNI Routing Header (ORH) (see: [I-D.templin-intarea-6706bis]) with final segment addressing information. The OAL next calculates and appends the trailing 32-bit CRC, then uses IPv6 fragmentation to break the packet into a minimum number of non-overlapping fragments where the size of each non-final fragment (including both the OMNI and outer-layer INET encapsulations) is determined by the minimum INET path MTU. The OAL then encapsulates the fragments in any INET headers and sends them to the OMNI link neighbor, which reassembles before forwarding toward the final destination.

In light of the above considerations, the OAL should assume a minimum path MTU of 576 bytes for the purpose of generating OAL fragments. Each OAL fragment will undergo *NET encapsulation including either a 20 byte IPv4 or 40 byte IPv6 header (plus an 8 byte UDP header for INETs), leaving a minimum of 528 bytes for each fragment. Each OAL fragment must accommodate 40 bytes for the OAL IPv6 header plus 8 bytes for the fragment header (while reserving 40 additional bytes in case a maximum-length ORH is inserted during re-encapsulation), leaving 440 bytes to accommodate the actual inner IP packet fragment. OAL fragmentation algorithms therefore MUST produce non-final fragments with the OAL IPv6 header Payload Length set to no less than 448 bytes (8 bytes for the fragment header plus 440 bytes for the inner packet fragment), while the Payload Length of the final fragment may be smaller. OAL reassembly algorithms MUST drop any non-final fragments with Payload Length less than 448 bytes.

Note that OAL fragmentation algorithms MAY produce larger non-final OAL fragments if better path MTU information is available. For example, for ANETs in which no UDP encapsulation header is needed the algorithm can increase the minimum non-final fragment length by 8 bytes. In a second example, for ANETs in which no IPv6 hops will be traversed over the path the algorithm can increase the minimum length by 20 bytes. In a third example, if there is assurance that no ORH will be inserted in the path the algorithm can increase the minimum length by 40 bytes. In a final example, when two ANET peers share a common physical or virtual link with a larger MTU (e.g., 1280 bytes or larger), the OAL can base the minimum non-final fragment length on this larger MTU size as long as the receiving ANET peer reassembles (and possibly also refragments) before forwarding. (Other examples are possible, and dependent on actual ANET/INET deployment scenarios.)

In all of the above examples, optimizing the minimum OAL fragment size may be important for accommodating links where performance is dependent on maximum use of the available link MTU, e.g. for wireless aviation data links. Additionally, in order to set the correct context for reassembly, the OMNI interface that inserts the OAL header MUST also be the one that inserts the IPv6 fragment header Identification value. While not strictly required, sending all fragments of the same fragmented OAL packet consecutively over the same underlying interface with minimal inter-fragment delay may increase the likelihood of successful reassembly.

Ordinary PTB messages with ICMPv4 header "unused" field or ICMPv6 header Code field value 0 are hard errors that always indicate that a packet has been dropped due to a real MTU restriction. However, the OAL can also forward large packets via encapsulation and fragmentation while at the same time returning PTB soft error

messages (subject to rate limiting) indicating that a forwarded packet was uncomfortably large. The OMNI interface can therefore continuously forward large packets without loss while returning PTB soft error messages recommending a smaller size. Original sources that receive the soft errors in turn reduce the size of the packets they send, i.e., the same as for hard errors.

The OAL sets the ICMPv4 header "unused" field or ICMPv6 header Code field to the value 1 in PTB soft error messages. The OAL sets the PTB destination address to the source address of the original packet, and sets the source address to the MNP Subnet Router Anycast address of the MN. The OAL then sets the MTU field to a value no smaller than 576 for ICMPv4 or 1280 for ICMPv6, and returns the PTB soft error to the original source.

When the original source receives the PTB, it reduces its path MTU estimate the same as for hard errors but does not regard the message as a loss indication. (If the original source does not recognize the soft error code, it regards the PTB the same as a hard error but should heed the retransmission advice given in [RFC8201] suggesting retransmission based on normal packetization layer retransmission timers.) This document therefore updates [RFC1191][RFC4443] and [RFC8201]. Furthermore, implementations of [RFC4821] must be aware that PTB hard or soft errors may arrive at any time even if after a successful MTU probe (this is the same consideration as for an ordinary path fluctuation following a successful probe).

In summary, the OAL 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 loss, original sources that receive soft errors can quickly scan for path MTU increases without waiting for the minimum 10 minutes specified for loss-oriented PTB hard errors [RFC1191][RFC8201]. The OAL therefore provides a lossless and adaptive service that accommodates MTU diversity especially well-suited for dynamic multilink environments.

Note: An OMNI interface that reassembles OAL fragments may experience congestion-oriented loss in its reassembly cache and can optionally send PTB soft errors to the original source and/or ICMP "Time Exceeded" messages to the source of the OAL fragments. In environments where the messages may contribute to unacceptable additional congestion, however, the OMNI interface can refrain from sending PTB soft errors and simply regard the loss as an ordinary unreported congestion event for which the original source will eventually compensate.

Note: When the network layer forwards an IPv4 packet/fragment with DF=0 into the OMNI interface, the interface can optionally perform (further) IPv4 fragmentation before invoking the OAL so that the fragments will be reassembled by the final destination. When the network layer performs IPv6 fragmentation for locally-generated IPv6 packets, the OMNI interface typically invokes the OAL without first applying (further) IPv6 fragmentation; the network layer should therefore fragment to the minimum IPv6 path MTU (or smaller still) to push the reassembly burden to the final destination and avoid receiving PTB soft errors from the OMNI interface. Aside from these non-normative guidelines, the manner in which any IP fragmentation is invoked prior to OAL encapsulation/fragmentation is an implementation matter.

Note: The source OAL includes a trailing 32-bit CRC only for OAL packets that require fragmentation, and the destination OAL discards any OAL packets with incorrect CRC values following reassembly. (The source OAL calculates the CRC over the entire packet, then appends the CRC to the end of the packet and adds the CRC length to the OAL Payload Length prior to fragmentation. The destination OAL subtracts the CRC length from the OAL Payload Length and verifies the CRC following reassembly.) A 32-bit CRC is sufficient for detecting reassembly misassociations for packet sizes no larger than the OMNI interface MTU but may not be sufficient to detect errors for larger sizes [CRC].

Note: Some underlying interface types (e.g., VPNs) may already provide their own robust fragmentation and reassembly services even without OAL encapsulation. In those cases, the OAL can invoke the inherent underlying interface schemes instead while employing PTB soft errors in the same fashion as described above. Other underlying interface facilities such as header/message compression can also be harnessed in a similar fashion.

Note: Applications can dynamically tune the size of the packets they to send to produce the best possible throughput and latency, with the understanding that these parameters may change over time due to factors such as congestion, mobility, network path changes, etc. The receipt or absence of soft errors should be seen as hints of when increasing or decreasing packet sizes may be beneficial.

5.1. Fragmentation Security Implications

As discussed in Section 3.7 of [RFC8900], there are four 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 underlying interface such that congestion experienced over a first underlying interface does not cause discard of incomplete reassemblies for uncongested underlying interfaces.

3. Attacks based on predictable fragment identification values - this threat is mitigated by selecting a suitably random ID value per [RFC7739].

4. Evasion of Network Intrusion Detection Systems (NIDS) - this threat is mitigated by disallowing "tiny fragments" per the OAL fragmentation procedures specified above.

Additionally, IPv4 fragmentation includes a 16-bit Identification (IP ID) field with only 65535 unique values such that at high data rates the field could wrap and apply to new packets while the fragments of old packets using the same ID are still alive in the network [RFC4963]. However, since the largest OAL fragment that will be sent via an IPv4 *NET path is 576 bytes any IPv4 fragmentation would occur only on links with an IPv4 MTU smaller than this size, and [RFC3819] recommendations suggest that such links will have low data rates. Since IPv6 provides a 32-bit Identification value, IP ID wraparound at high data rates is not a concern for IPv6 fragmentation.

Finally, [RFC6980] documents fragmentation security concerns for large IPv6 ND messages. These concerns are addressed when the OMNI interface employs the OAL instead of directly fragmenting the IPv6 ND message itself. For this reason, OMNI interfaces MUST NOT admit IPv6 ND messages larger than the OMNI interface MTU, and MUST employ the OAL for IPv6 ND messages admitted into the OMNI interface the same as discussed above.

5.2. OAL "Super-Packet" Packing

By default, the source OAL includes a 40-byte IPv6 encapsulation header for each inner IP payload packet during OAL encapsulation. When fragmentation is needed, the source OAL also calculates and

includes a 32-bit trailing CRC for the entire packet then performs fragmentation such that a copy of the 40-byte IPv6 header plus an 8-byte IPv6 Fragment Header is included in each fragment. However, these encapsulations may represent excessive overhead in some environments. A technique known as "packing" discussed in [I-D.ietf-intarea-tunnels] is therefore supported so that multiple inner IP payload packets can be included within a single OAL packet known as a "super-packet".

When the source OAL has multiple inner IP payload packets with total length no larger than the OMNI interface MTU to send to the same destination, it can optionally concatenate them into a super-packet encapsulated in a single OAL header. Within the super-packet, the IP header of the first inner packet (iHa) followed by its data (iDa) is concatenated immediately following the OAL header, then the inner IP header of the next packet (iHb) followed by its data (iDb) is concatenated immediately following the first packet, etc. The super-packet format is transposed from [I-D.ietf-intarea-tunnels] and shown in Figure 3:

```
   <-- Multiple inner IP payload packets to be "packed" -->
                +-----+-----+
                | iHa | iDa |
                +-----+-----+
                      |
                      |     +-----+-----+
                      |     | iHb | iDb |
                      |     +-----+-----+
                      |           |
                      |           |     +-----+-----+
                      |           |     | iHc | iDc |
                      |           |     +-----+-----+
                      |           |           |
                      v           v           v
     +----------+-----+-----+-----+-----+-----+-----+
     |  OAL Hdr | iHa | iDa | iHb | iDb | iHc | iDc |
     +----------+-----+-----+-----+-----+-----+-----+
     <-- OMNI "Super-Packet" with single OAL hdr  -->
```

Figure 3: OAL Super-Packet Format

When the source OAL sends a super-packet, it calculates a CRC and applies OAL fragmentation if necessary then sends the packet or fragments to the destination OAL. When the destination OAL receives the super-packet as a whole packet or as fragments, it reassembles and verifies the CRC if necessary then regards the OAL header Payload Length (after subtracting the CRC length) as the sum of the lengths of all payload packets. The destination OAL then selectively

extracts each individual payload packet (e.g., by setting pointers into the buffer containing the super-packet and maintaining a reference count, by copying each payload packet into its own buffer, etc.) and forwards each payload packet or processes it locally as appropriate. During extraction, the OAL determines the IP protocol version of each successive inner payload packet 'j' by examining the first four bits of iH(j), and determines the length of the inner packet by examining the rest of iH(j) according to the IP protocol version.

Note: OMNI interfaces must take care to avoid processing super-packet payload elements that would subvert security. Specifically, if a super-packet contains a mix of data and control payload packets (which could include critical security codes), the node MUST NOT process the data packets before processing the control packets.

6. Frame Format

The OMNI interface transmits IP packets according to the native frame format of each underlying 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 tunnels over IPv6 the frame format is specified in [RFC2473], etc.

7. Link-Local Addresses (LLAs)

OMNI nodes are assigned OMNI interface IPv6 Link-Local Addresses (LLAs) through pre-service administrative actions. "MNP-LLAs" embed the MNP assigned to the mobile node, while "ADM-LLAs" include an administratively-unique ID that is guaranteed to be unique on the link. LLAs are configured as follows:

- o IPv6 MNP-LLAs encode the most-significant 64 bits of a MNP within the least-significant 64 bits of the IPv6 link-local prefix fe80::/64, i.e., in the LLA "interface identifier" portion. The prefix length for the LLA is determined by adding 64 to the MNP prefix length. For example, for the MNP 2001:db8:1000:2000::/56 the corresponding MNP-LLA is fe80::2001:db8:1000:2000/120.

- o IPv4-compatible MNP-LLAs are constructed as fe80::ffff:[IPv4], i.e., the interface identifier consists of 16 '0' bits, followed by 16 '1' bits, followed by a 32bit IPv4 address/prefix. The prefix length for the LLA is determined by adding 96 to the MNP prefix length. For example, the IPv4-Compatible MN OMNI LLA for 192.0.2.0/24 is fe80::ffff:192.0.2.0/120 (also written as fe80::ffff:c000:0200/120).

o ADM-LLAs are assigned to ARs and MSEs and MUST be managed for uniqueness. The lower 32 bits of the LLA includes a unique integer "MSID" value between 0x00000001 and 0xfeffffff, e.g., as in fe80::1, fe80::2, fe80::3, etc., fe80::feffffff. The ADM-LLA prefix length is determined by adding 96 to the MSID prefix length. For example, if the prefix length for MSID 0x10012001 is 16 then the ADM-LLA prefix length is set to 112 and the LLA is written as fe80::1001:2001/112. The "zero" address for each ADM-LLA prefix is the Subnet-Router anycast address for that prefix [RFC4291]; for example, the Subnet-Router anycast address for fe80::1001:2001/112 is simply fe80::1001:2000. The MSID range 0xff000000 through 0xffffffff is reserved for future use.

o Temporary LLAs are constructed per [I-D.ietf-6man-rfc4941bis] and used by MNs for the short-term purpose of procuring an actual MNP-LLA upon startup or (re)connecting to the network. MNs may use Temporary LLAs as the IPv6 source address of an RS message in order to request a MNP-LLA from the MS.

Since the prefix 0000::/8 is "Reserved by the IETF" [RFC4291], no MNPs can be allocated from that block ensuring that there is no possibility for overlap between the MNP- and ADM-LLA constructs discussed above.

Since MNP-LLAs are based on the distribution of administratively assured unique MNPs, and since ADM-LLAs are guaranteed unique through administrative assignment, OMNI interfaces set the autoconfiguration variable DupAddrDetectTransmits to 0 [RFC4862].

Temporary LLAs are distinguished from MNP- and ADM-LLAs by examining the OMNI option T field (see: Section 11), and employ optimistic DAD principles [RFC4429] since they are probabilistically unique and their use is short-duration in nature.

Note: If future protocol extensions relax the 64-bit boundary in IPv6 addressing, the additional prefix bits of an MNP could be encoded in bits 16 through 63 of the MNP-LLA. (The most-significant 64 bits would therefore still be in bits 64-127, and the remaining bits would appear in bits 16 through 48.) However, the analysis provided in [RFC7421] suggests that the 64-bit boundary will remain in the IPv6 architecture for the foreseeable future.

Note: Even though this document honors the 64-bit boundary in IPv6 addressing per [RFC7421], it suggests prefix lengths longer than /64 for routing purposes. This effectively extends IPv6 routing determination into the interface identifier portion of the IPv6 address, but it does not redefine the 64-bit boundary.

8. Unique-Local Addresses (ULAs)

OMNI domains use IPv6 Unique-Local Addresses (ULAs) as the source and destination addresses in OAL IPv6 encapsulation headers. ULAs are only routable within the scope of a an OMNI domain, and are derived from the IPv6 Unique Local Address prefix fc00::/7 followed by the L bit set to 1 (i.e., as fd00::/8) followed by a 40-bit pseudo-random Global ID to produce the prefix [ULA]::/48, which is then followed by a 16-bit Subnet ID then finally followed by a 64 bit Interface ID as specified in Section 3 of [RFC4193]. All nodes in the same OMNI domain configure the same 40-bit Global ID as the OMNI domain identifier. The statistic uniqueness of the 40-bit pseudo-random Global ID allows different OMNI domains to be joined together in the future without requiring renumbering.

Each OMNI link instance is identified by a value between 0x0000 and 0xfeff in bits 48-63 of [ULA]::/48 (the values 0xff00 through 0xffff are reserved for future use). For example, OMNI ULAs associated with instance 0 are configured from the prefix [ULA]:0000::/64, instance 1 from [ULA]:0001::/64, instance 2 from [ULA]:0002::/64, etc. ULAs and their associated prefix lengths are configured in correspondence with LLAs through stateless prefix translation where "MNP-ULAs" are assigned in correspondence to MNP-LLAs and "ADM-ULAs" are assigned in correspondence to ADM-LLAs. For example, for OMNI link instance [ULA]:1010::/64:

- the MNP-ULA corresponding to the MNP-LLA fe80::2001:db8:1:2 with a 56-bit MNP length is derived by copying the lower 64 bits of the LLA into the lower 64 bits of the ULA as [ULA]:1010:2001:db8:1:2/120 (where, the ULA prefix length becomes 64 plus the IPv6 MNP length).

- the MNP-ULA corresponding to fe80::ffff:192.0.2.0 with a 28-bit MNP length is derived by simply writing the LLA interface ID into the lower 64 bits as [ULA]:1010:0:ffff:192.0.2.0/124 (where, the ULA prefix length is 64 plus 32 plus the IPv4 MNP length).

- the ADM-ULA corresponding to fe80::1000/112 is simply [ULA]:1010::1000/112.

- the ADM-ULA corresponding to fe80::/128 is simply [ULA]:1010::/128.

- the Temporary ULA corresponding to a Temporary LLA is simply [ULA]:1010:[64-bit Temporary Interface ID]/128.

- etc.

Each OMNI interface assigns the Anycast ADM-ULA specific to the OMNI link instance. For example, the OMNI interface connected to instance 3 assigns the Anycast address [ULA]:0003::/128. Routers that configure OMNI interfaces advertise the OMNI service prefix (e.g., [ULA]:0003::/64) into the local routing system so that applications can direct traffic according to SBM requirements.

The ULA presents an IPv6 address format that is routable within the OMNI routing system and can be used to convey link-scoped IPv6 ND messages across multiple hops using IPv6 encapsulation [RFC2473]. The OMNI link extends across one or more underling Internetworks to include all ARs and MSEs. All MNs are also considered to be connected to the OMNI link, however OAL encapsulation is omitted whenever possible to conserve bandwidth (see: Section 13).

Each OMNI link can be subdivided into "segments" that often correspond to different administrative domains or physical partitions. OMNI nodes can use IPv6 Segment Routing [RFC8402] when necessary to support efficient packet forwarding to destinations located in other OMNI link segments. A full discussion of Segment Routing over the OMNI link appears in [I-D.templin-intarea-6706bis].

Note: IPv6 ULAs taken from the prefix fc00::/7 followed by the L bit set to 0 (i.e., as fc00::/8) are never used for OMNI OAL addressing, however the range could be used for MSP and MNP addressing under certain limiting conditions (see: Section 9).

9. Global Unicast Addresses (GUAs)

OMNI domains use IP Global Unicast Address (GUA) prefixes [RFC4291] as Mobility Service Prefixes (MSPs) from which Mobile Network Prefixes (MNP) are delegated to Mobile Nodes (MNs).

For IPv6, GUA prefixes are assigned by IANA [IPV6-GUA] and/or an associated regional assigned numbers authority such that the OMNI domain can be interconnected to the global IPv6 Internet without causing inconsistencies in the routing system. An OMNI domain could instead use ULAs with the 'L' bit set to 0 (i.e., from the prefix fc00::/8)[RFC4193], however this would require IPv6 NAT if the domain were ever connected to the global IPv6 Internet.

For IPv4, GUA prefixes are assigned by IANA [IPV4-GUA] and/or an associated regional assigned numbers authority such that the OMNI domain can be interconnected to the global IPv4 Internet without causing routing inconsistencies. An OMNI domain could instead use private IPv4 prefixes (e.g., 10.0.0.0/8, etc.) [RFC3330], however this would require IPv4 NAT if the domain were ever connected to the global IPv4 Internet.

10. Node Identification

OMNI MNs and MSEs that connect over open Internetworks generate a Host Identity Tag (HIT) as specified in [RFC7401] and use the value as a robust general-purpose node identification value. Hierarchical HITs (HHITs) [I-D.ietf-drip-rid] may provide a useful alternative in certain domains such as the Unmanned (Air) Traffic Management (UTM) service for Unmanned Air Systems (UAS). MNs and MSEs can then use their (H)HITs in IPv6 ND control message exchanges.

When a MN is truly outside the context of any infrastructure, it may have no MNP information at all. In that case, the MN can use its (H)HIT as an IPv6 source/destination address for sustained communications in Vehicle-to-Vehicle (V2V) and (multihop) Vehicle-to-Infrastructure (V2I) scenarios. The MN can also propagate the (H)HIT into the multihop routing tables of (collective) Mobile/Vehicular Ad-hoc Networks (MANETs/VANETs) using only the vehicles themselves as communications relays.

When a MN connects to ARs over (non-multihop) protected-spectrum ANETs, an alternate form of node identification (e.g., MAC address, serial number, airframe identification value, VIN, etc.) may be sufficient. In that case, the MN should still generate a (H)HIT and maintain it in conjunction with any other node identifiers. The MN can then include OMNI "Node Identification" sub-options (see: Section 11.1.12) in IPv6 ND messages should the need to transmit identification information over the network arise.

11. Address Mapping - Unicast

OMNI interfaces maintain a neighbor cache for tracking per-neighbor state and use the link-local address format specified in Section 7. OMNI interface IPv6 Neighbor Discovery (ND) [RFC4861] messages sent over physical underlying interfaces without encapsulation observe the native underlying interface Source/Target Link-Layer Address Option (S/TLLAO) format (e.g., for Ethernet the S/TLLAO is specified in [RFC2464]). OMNI interface IPv6 ND messages sent over underlying interfaces via encapsulation do not include S/TLLAOs which were intended for encoding physical L2 media address formats and not encapsulation IP addresses. Furthermore, S/TLLAOs are not intended for encoding additional interface attributes needed for multilink coordination. Hence, this document does not define an S/TLLAO format but instead defines a new option type termed the "OMNI option" designed for these purposes.

MNs such as aircraft typically have many 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 underlying interfaces in a single IPv6 ND message exchange. OMNI interfaces use an IPv6 ND option called the OMNI option formatted as shown in Figure 4:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Type     |     Length    |T|   Preflen   | S/T-omIndex  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                          Sub-Options                          ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: OMNI Option Format

In this format:

- Type is set to TBD1.

- Length is set to the number of 8 octet blocks in the option. The value 0 is invalid, while the values 1 through 255 (i.e., 8 through 2040 octets, respectively) indicate the total length of the OMNI option.

- T is a 1 bit field set to 1 for Temporary LLAs (otherwise, set to 0) and Preflen is a 7 bit field that determines the length of prefix associated with an LLA. Values 1 through 127 specify a prefix length, while the value 0 indicates "unspecified". For IPv6 ND messages sent from a MN to the MS, T and Preflen apply to the IPv6 source LLA and provide the length that the MN is requesting or asserting to the MS. For IPv6 ND messages sent from the MS to the MN, T and Preflen apply to the IPv6 destination LLA and indicate the length that the MS is granting to the MN. For IPv6 ND messages sent between MS endpoints, T is set to 0 and Preflen provides the length associated with the source/target MN that is subject of the ND message.

- S/T-omIndex is an 8 bit field corresponds to the omIndex value for source or target underlying interface used to convey this IPv6 ND message. OMNI interfaces MUST number each distinct underlying interface with an omIndex value between '1' and '255' that represents a MN-specific 8-bit mapping for the actual ifIndex value assigned by network management [RFC2863] (the omIndex value '0' is reserved for use by the MS). For RS and NS messages, S/T-omIndex corresponds to the source underlying interface the

message originated from. For RA and NA messages, S/T-omIndex corresponds to the target underlying interface that the message is destined to. (For NS messages used for Neighbor Unreachability Detection (NUD), S/T-omIndex instead identifies the neighbor's underlying interface to be used as the target interface to return the NA.)

- o Sub-Options is a Variable-length field, of length such that the complete OMNI Option is an integer multiple of 8 octets long. Contains one or more Sub-Options, as described in Section 11.1.

The OMNI option may appear in any IPv6 ND message type; it is processed by interfaces that recognize the option and ignored by all other interfaces. If multiple OMNI option instances appear in the same IPv6 ND message, the interface processes the T, Preflen and S/T-omIndex fields in the first instance and ignores those fields in all other instances. The interface processes the Sub-Options of all OMNI option instances in the consecutive order in which they appear in the IPv6 ND message, beginning with the first instance and continuing consecutively through any additional instances to the end of the message.

The OMNI option(s) in each IPv6 ND message may include full or partial information for the neighbor. The union of the information in the most recently received OMNI options is therefore retained, and the information is aged/removed in conjunction with the corresponding neighbor cache entry.

11.1. Sub-Options

The OMNI option includes zero or more Sub-Options. Each consecutive Sub-Option is concatenated immediately after its predecessor. All Sub-Options except Pad1 (see below) are in type-length-value (TLV) encoded in the following format:

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Sub-Type|      Sub-length     | Sub-Option Data ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 5: Sub-Option Format

- o Sub-Type is a 5-bit field that encodes the Sub-Option type. Sub-Options defined in this document are:

Option Name	Sub-Type
Pad1	0
PadN	1
Interface Attributes (Type 1)	2
Interface Attributes (Type 2)	3
Traffic Selector	4
Origin Indication	5
MS-Register	6
MS-Release	7
Geo Coordinates	8
DHCPv6 Message	9
HIP Message	10
Node Identification	11

Figure 6

Sub-Types 12-29 are available for future assignment. Sub-Type 30 is reserved for experimentation, as recommended in [RFC3692]. Sub-Type 31 is reserved by IANA.

- o Sub-Length is an 11-bit field that encodes the length of the Sub-Option Data ranging from 0 to 2034 octets (the values 2035 through 2047 are invalid, since each OMNI option is limited to 2040 octets).

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

Note that Sub-Type and Sub-Length are coded together in network byte order in two consecutive octets. Note also that Sub-Option Data may be up to 2034 octets in length. This allows ample space for encoding large objects (e.g., ascii character strings, protocol messages, security codes, etc.), while a single OMNI option is limited to 2040 octets the same as for any IPv6 ND option. If the Sub-Options to be coded would cause an OMNI option to exceed 2040 octets, any remaining Sub-Options are encoded in additional OMNI options in the consecutive order of intended processing. Implementations must therefore be mindful of size limitations, and must refrain from sending IPv6 ND messages larger than the OMNI interface MTU.

During processing, unrecognized Sub-Options are ignored and the next Sub-Option processed until the end of the OMNI option is reached.

The following Sub-Option types and formats are defined in this document:

11.1.1. Pad1

```
         0
         0 1 2 3 4 5 6 7
        +-+-+-+-+-+-+-+-+
        | S-Type=0|x|x|x|
        +-+-+-+-+-+-+-+-+
```

Figure 7: Pad1

o Sub-Type is set to 0. If multiple instances appear in OMNI options of the same message all are processed.

o Sub-Type is followed by three 'x' bits, set randomly on transmission and ignored on receipt. Pad1 therefore consists of a whole single octet with the most significant 5 bits set to 0, and with no Sub-Length or Sub-Option Data fields following.

11.1.2. PadN

```
         0                   1                   2
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
        | S-Type=1|    Sub-length=N     | N padding octets ...
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 8: PadN

o Sub-Type is set to 1. If multiple instances appear in OMNI options of the same message all are processed.

o Sub-Length is set to N (from 0 to 2034 being the number of padding octets that follow.

o Sub-Option Data consists of N zero-valued octets.

11.1.3. Interface Attributes (Type 1)

The Interface Attributes (Type 1) sub-option provides a basic set of attributes for underlying interfaces. Interface Attributes (Type 1) is deprecated throughout the rest of this specification, and Interface Attributes (Type 2) (see: Section 11.1.4) are indicated wherever the term "Interface Attributes" appears without an associated Type designation.

Nodes SHOULD NOT include Interface Attributes (Type 1) sub-options in IPv6 ND messages they send, and MUST ignore any in IPv6 ND messages

they receive. If an Interface Attributes (Type 1) is included, it must have the following format:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Sub-Type=2|   Sub-length=N    |    omIndex    |    omType     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Provider ID  | Link  | Resvd |P00|P01|P02|P03|P04|P05|P06|P07|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |P08|P09|P10|P11|P12|P13|P14|P15|P16|P17|P18|P19|P20|P21|P22|P23|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |P24|P25|P26|P27|P28|P29|P30|P31|P32|P33|P34|P35|P36|P37|P38|P39|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |P40|P41|P42|P43|P44|P45|P46|P47|P48|P49|P50|P51|P52|P53|P54|P55|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |P56|P57|P58|P59|P60|P61|P62|P63|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Interface Attributes (Type 1)

- o Sub-Type is set to 2. If multiple instances with different omIndex values appear in OMNI option of the same message all are processed; if multiple instances with the same omIndex value appear, the first is processed and all others are ignored
- o Sub-Length is set to N (from 4 to 2034 that encodes the number of Sub-Option Data octets that follow.
- o omIndex is a 1-octet field containing a value from 0 to 255 identifying the underlying interface for which the attributes apply.
- o omType is a 1-octet field containing a value from 0 to 255 corresponding to the underlying interface identified by omIndex.
- o Provider ID is a 1-octet field containing a value from 0 to 255 corresponding to the underlying interface identified by omIndex.
- o Link encodes a 4-bit link metric. The value '0' means the link is DOWN, and the remaining values mean the link is UP with metric ranging from '1' ("lowest") to '15' ("highest").
- o Resvd is reserved for future use.
- o A 16-octet ""Preferences" field immediately follows 'Resvd', with values P[00] through P[63] corresponding to the 64 Differentiated Service Code Point (DSCP) values [RFC2474]. Each 2-bit P[*] field

is set to the value '0' ("disabled"), '1' ("low"), '2' ("medium") or '3' ("high") to indicate a QoS preference for underlying interface selection purposes.

11.1.4. Interface Attributes (Type 2)

The Interface Attributes (Type 2) sub-option provides L2 forwarding information for the multilink conceptual sending algorithm discussed in Section 13. The L2 information is used for selecting among potentially multiple candidate underlying interfaces that can be used to forward packets to the neighbor based on factors such as DSCP preferences and link quality. Interface Attributes (Type 2) further includes link-layer address information to be used for either OAL encapsulation or direct UDP/IP encapsulation (when OAL encapsulation can be avoided).

Interface Attributes (Type 2) are the sole Interface Attributes format in this specification that all OMNI nodes must honor. Wherever the term "Interface Attributes" occurs throughout this specification without a "Type" designation, the format given below is indicated:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| S-Type=3|    Sub-length=N     |    omIndex     |    omType     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Provider ID  | Link  |R| API |    SRT    | FMT |   LHS (0 - 7) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                LHS (bits 8 - 31)               |               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               ~
~                                                                ~
~                    Link Layer Address (L2ADDR)                 ~
~                                                                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Bitmap(0)=0xff|P00|P01|P02|P03|P04|P05|P06|P07|P08|P09|P10|P11|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P12|P13|P14|P15|P16|P17|P18|P19|P20|P21|P22|P23|P24|P25|P26|P27|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P28|P29|P30|P31| Bitmap(1)=0xff|P32|P33|P34|P35|P36| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 10: Interface Attributes (Type 2)

- o Sub-Type is set to 3. If multiple instances with different omIndex values appear in OMNI options of the same message all are processed; if multiple instances with the same omIndex value appear, the first is processed and all others are ignored.

- Sub-Length is set to N (from 4 to 2034) that encodes the number of Sub-Option Data octets that follow.

- Sub-Option Data contains an "Interface Attributes (Type 2)" option encoded as follows (note that the first four octets must be present):

  * omIndex is set to an 8-bit integer value corresponding to a specific underlying interface the same as specified above for the OMNI option S/T-omIndex field. The OMNI options of a same message may include multiple Interface Attributes Sub-Options, with each distinct omIndex value pertaining to a different underlying interface. The OMNI option will often include an Interface Attributes Sub-Option with the same omIndex value that appears in the S/T-omIndex. In that case, the actual encapsulation address of the received IPv6 ND message should be compared with the L2ADDR encoded in the Sub-Option (see below); if the addresses are different (or, if L2ADDR is absent) the presence of a NAT is assumed.

  * omType is set to an 8-bit integer value corresponding to the underlying interface identified by omIndex. The value represents an OMNI interface-specific 8-bit mapping for the actual IANA ifType value registered in the 'IANAifType-MIB' registry [http://www.iana.org].

  * Provider ID is set to an OMNI interface-specific 8-bit ID value for the network service provider associated with this omIndex.

  * Link encodes a 4-bit link metric. The value '0' means the link is DOWN, and the remaining values mean the link is UP with metric ranging from '1' ("lowest") to '15' ("highest").

  * R is reserved for future use.

  * API - a 3-bit "Address/Preferences/Indexed" code that determines the contents of the remainder of the sub-option as follows:

    + When the most significant bit (i.e., "Address") is set to 1, the SRT, FMT, LHS and L2ADDR fields are included immediately following the API code; else, they are omitted.

    + When the next most significant bit (i.e., "Preferences") is set to 1, a preferences block is included next; else, it is omitted. (Note that if "Address" is set the preferences block immediately follows L2ADDR; else, it immediately follows the API code.)

+ When a preferences block is present and the least significant bit (i.e., "Indexed") is set to 0, the block is encoded in "Simplex" form as shown in Figure 9; else it is encoded in "Indexed" form as discussed below.

* When API indicates that an "Address" is included, the following fields appear in consecutive order (else, they are omitted):

  + SRT - a 5-bit Segment Routing Topology prefix length value that (when added to 96) determines the prefix length to apply to the ULA formed from concatenating [ULA*]::/96 with the 32 bit LHS MSID value that follows. For example, the value 16 corresponds to the prefix length 112.

  + FMT - a 3-bit "Framework/Mode/Type" code corresponding to the included Link Layer Address as follows:

    - When the most significant bit (i.e., "Framework") is set to 0, L2ADDR is the INET encapsulation address of a Proxy/Server; otherwise, it is the address for the Source/Target itself

    - When the next most significant bit (i.e., "Mode") is set to 0, the Source/Target L2ADDR is on the open INET; otherwise, it is (likely) located behind one or more NATs.

    - When the least significant bit (i.e., "Type") is set to 0, L2ADDR includes a UDP Port Number followed by an IPv4 address; else, a UDP Port Number followed by an IPv6 address.

  + LHS - the 32 bit MSID of the Last Hop Server/Proxy on the path to the target. When SRT and LHS are both set to 0, the LHS is considered unspecified in this IPv6 ND message. When SRT is set to 0 and LHS is non-zero, the prefix length is set to 128. SRT and LHS together provide guidance to the OMNI interface forwarding algorithm. Specifically, if SRT/LHS is located in the local OMNI link segment then the OMNI interface can encapsulate according to FMT/L2ADDR (following any necessary NAT traversal messaging); else, it must forward according to the OMNI link spanning tree. See [I-D.templin-intarea-6706bis] for further discussion.

  + Link Layer Address (L2ADDR) - Formatted according to FMT, and identifies the link-layer address (i.e., the encapsulation address) of the source/target. The UDP Port Number appears in the first two octets and the IP address

appears in the next 4 octets for IPv4 or 16 octets for IPv6. The Port Number and IP address are recorded in ones-compliment "obfuscated" form per [RFC4380]. The OMNI interface forwarding algorithm uses FMT/L2ADDR to determine the encapsulation address for forwarding when SRT/LHS is located in the local OMNI link segment. Note that if the target is behind a NAT, L2ADDR will contain the mapped INET address stored in the NAT; otherwise, L2ADDR will contain the native INET information of the target itself.

* When API indicates that "Preferences" are included, a preferences block appears as the remainder of the Sub-Option as a series of Bitmaps and P[*] values. In "Simplex" form, the index for each singleton Bitmap octet is inferred from its sequential position (i.e., 0, 1, 2, ...) as shown in Figure 9. In "Indexed" form, each Bitmap is preceded by an Index octet that encodes a value "i" = (0 - 255) as the index for its companion Bitmap as follows:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|   Index=i     |   Bitmap(i)   |P[*] values ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 11

* The preferences consist of a first (simplex/indexed) Bitmap (i.e., "Bitmap(i)") followed by 0-8 single-octet blocks of 2-bit P[*] values, followed by a second Bitmap (i), followed by 0-8 blocks of P[*] values, etc. Reading from bit 0 to bit 7, the bits of each Bitmap(i) that are set to '1'' indicate the P[*] blocks from the range P[(i*32)] through P[(i*32) + 31] that follow; if any Bitmap(i) bits are '0', then the corresponding P[*] block is instead omitted. For example, if Bitmap(0) contains 0xff then the block with P[00]-P[03], followed by the block with P[04]-P[07], etc., and ending with the block with P[28]-P[31] are included (as shown in Figure 9). The next Bitmap(i) is then consulted with its bits indicating which P[*] blocks follow, etc. out to the end of the Sub-Option.

* Each 2-bit P[*] field is set to the value '0' ("disabled"), '1' ("low"), '2' ("medium") or '3' ("high") to indicate a QoS preference for underlying interface selection purposes. Not all P[*] values need to be included in the OMNI option of each IPv6 ND message received. Any P[*] values represented in an earlier OMNI option but omitted in the current OMNI option remain unchanged. Any P[*] values not yet represented in any OMNI option default to "medium".

* The first 16 P[*] blocks correspond to the 64 Differentiated Service Code Point (DSCP) values P[00] - P[63] [RFC2474]. Any additional P[*] blocks that follow correspond to "pseudo-DSCP" traffic classifier values P[64], P[65], P[66], etc. See Appendix A for further discussion and examples.

11.1.5. Traffic Selector

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| S-Type=4|     Sub-length=N    |    omIndex    |               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               ~
~                                                               ~
~                 RFC 6088 Format Traffic Selector              ~
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Traffic Selector

o Sub-Type is set to 4. If multiple instances appear in OMNI options of the same message all are processed, i.e., even if the same omIndex value appears multiple times.

o Sub-Length is set to N (the number of Sub-Option Data octets that follow).

o Sub-Option Data contains a 1-octet omIndex encoded exactly as specified in Section 11.1.3, followed by an N-1 octet traffic selector formatted per [RFC6088] beginning with the "TS Format" field. The largest traffic selector for a given omIndex is therefore 2033 octets.

11.1.6. Origin Indication

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| S-Type=5|    Sub-length=6/18  |       Origin Port Number      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                      Origin IPv4/IPv6 Address                 ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: Origin Indication

o Sub-Type is set to 5. If multiple instances appear in OMNI options of the same message the first instance is processed and all others are ignored.

o Sub-Length is set to either 6 or 18; if Sub-Length encodes any other value, the Sub-Option is ignored.

o Sub-Option Data contains a 2-octet Port Number followed by a 4-octet IPv4 address if Sub-Length encodes 6 or a 16-octet IPv6 address if Sub-Length encodes 18.

11.1.7. MS-Register

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| S-Type=6|    Sub-length=4n    |      MSID[1] (bits 0 - 15)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     MSID [1] (bits 16 - 32)   |      MSID[2] (bits 0 - 15)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     MSID [2] (bits 16 - 32)   |      MSID[3] (bits 0 - 15)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ...        ...        ...        ...        ...        ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     MSID [n] (bits 16 - 32)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: MS-Register Sub-option

o Sub-Type is set to 6. If multiple instances appear in OMNI options of the same message all are processed. Only the first MAX_MSID values processed (whether in a single instance or multiple) are retained and all other MSIDs are ignored.

o Sub-Length is set to 4n, with 508 as the maximum value for n.

o A list of n 4-octet MSIDs is included in the following 4n octets. The Anycast MSID value '0' in an RS message MS-Register sub-option requests the recipient to return the MSID of a nearby MSE in a corresponding RA response.

11.1.8. MS-Release

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | S-Type=7|    Sub-length=4n    |      MSID[1] (bits 0 - 15)    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |     MSID [1] (bits 16 - 32)   |      MSID[2] (bits 0 - 15)    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |     MSID [2] (bits 16 - 32)   |      MSID[3] (bits 0 - 15)    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           ...        ...        ...        ...       ...        ...
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |     MSID [n] (bits 16 - 32)   |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: MS-Release Sub-option

- o Sub-Type is set to 7. If multiple instances appear in OMNI options of the same message all are processed. Only the first MAX_MSID values processed (whether in a single instance or multiple) are retained and all other MSIDs are ignored.

- o Sub-Length is set to 4n, with 508 as the maximum value for n.

- o A list of n 4 octet MSIDs is included in the following 4n octets. The Anycast MSID value '0' is ignored in MS-Release sub-options, i.e., only non-zero values are processed.

11.1.9. Geo Coordinates

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | S-Type=8|    Sub-length=N     |      Geo Coordinates
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 16: Geo Coordinates Sub-option

- o Sub-Type is set to 8. If multiple instances appear in OMNI options of the same message the first is processed and all others are ignored.

- o Sub-Length is set to N (i.e., the length of the encoded Geo Coordinates up to a maximum of 2034 octets).

- o A set of Geo Coordinates. Format(s) to be specified in future documents; should include Latitude/Longitude, plus any additional attributes such as altitude, heading, speed, etc.

11.1.10. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Message

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| S-Type=9|    Sub-length=N     |    msg-type   |  id (octet 0) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   transaction-id (octets 1-2) |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
.                        DHCPv6 options                         .
.                 (variable number and length)                  .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: DHCPv6 Message Sub-option

- Sub-Type is set to 9. If multiple instances appear in OMNI options of the same message the first is processed and all others are ignored.

- Sub-Length is set to N (i.e., the length of the DHCPv6 message beginning with 'msg-type' and continuing to the end of the DHCPv6 options). The length of the entire DHCPv6 message is therefore restricted to 2034 octets.

- 'msg-type' and 'transaction-id' are coded according to Section 8 of [RFC8415].

- A set of DHCPv6 options coded according to Section 21 of [RFC8415] follows.

11.1.11. Host Identity Protocol (HIP) Message

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S-Type=10|    Sub-length=N     |0| Packet Type |Version| RES.|1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Checksum             |           Controls            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Sender's Host Identity Tag (HIT)               |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Receiver's Host Identity Tag (HIT)              |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
/                        HIP Parameters                         /
/                                                               /
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: HIP Message Sub-option

- Sub-Type is set to 10. If multiple instances appear in OMNI options of the same message the first is processed and all others are ignored.

- Sub-Length is set to N, i.e., the length of the option in octets beginning immediately following the Sub-Length field and extending to the end of the HIP parameters. The length of the entire HIP message is therefore restricted to 2034 octets.

- The HIP message is coded exactly as specified in Section 5 of [RFC7401], with the exception that the OMNI "Sub-Type" and "Sub-Length" fields replace the first two header octets of the HIP message (i.e., the Next Header and Header Length fields).

11.1.12. Node Identification

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S-Type=11|    Sub-length=N     |     ID-Type    |              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+              ~
~            Node Identification Value (N-1 octets)             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19: Node Identification

- o Sub-Type is set to 11. If multiple instances appear in OMNI options of the same IPv6 ND message the first instance of a specific ID-Type is processed and all other instances of the same ID-Type are ignored. (Note therefore that it is possible for a single IPv6 ND message to convey multiple Node Identifications - each having a different ID-Type.)

- o Sub-Length is set to N, i.e., the combined length of the ID-Type and Node Identification Value fields. The maximum Node Identification Value length is therefore 2033 octets.

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

  * 0 - Universally Unique IDentifier (UUID) [RFC4122]. Indicates that Node Identification Value contains a 16 octet UUID.

  * 1 - Host Identity Tag (HIT) [RFC7401]. Indicates that Node Identification Value contains a 16 octet HIT.

  * 2 - Hierarchical HIT (HHIT) [I-D.ietf-drip-rid]. Indicates that Node Identification Value contains a 16 octet HHIT.

  * 3 - Network Access Identifier (NAI) [RFC7542]. Indicates that Node Identification Value contains an N-1 octet NAI.

  * 4 - Fully-Qualified Domain Name (FQDN) [RFC1035]. Indicates that Node Identification Value contains an N-1 octet FQDN.

  * 5 - 252 - Unassigned.

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

  * 255 - reserved by IANA.

o Node Identification Value is an (N - 1)-octet field encoded according to the appropriate the "ID-Type" reference above.

When a Node Identification Value is needed for DHCPv6 messaging purposes, it is encoded as a DHCP Unique IDentifier (DUID) using the "DUID-EN for OMNI" format with enterprise number 45282 (see: Section 22) as shown in Figure 20:

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         DUID-Type (2)         |      EN (high bits == 0)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     EN (low bits = 45282)     |    ID-Type    |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
.                   Node Identification Value                   .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20: DUID-EN for OMNI Format

In this format, the ID-Type and Node Identification Value fields are coded exactly as in Figure 19 following the 6 octet DUID-EN header, and the entire "DUID-EN for OMNI" is included in a DHCPv6 message per [RFC8415].

12. Address Mapping - Multicast

The multicast address mapping of the native underlying interface applies. The mobile router on board the MN also serves as an IGMP/MLD Proxy for its EUNs and/or hosted applications per [RFC4605] while using the L2 address of the AR as the L2 address for all multicast packets.

The MN uses Multicast Listener Discovery (MLDv2) [RFC3810] to coordinate with the AR, and *NET L2 elements use MLD snooping [RFC4541].

13. Multilink Conceptual Sending Algorithm

The MN's IPv6 layer selects the outbound OMNI interface according to SBM considerations when forwarding data packets from local or EUN applications to external correspondents. Each OMNI interface maintains a neighbor cache the same as for any IPv6 interface, but with additional state for multilink coordination. Each OMNI interface maintains default routes via ARs discovered as discussed in Section 14, and may configure more-specific routes discovered through means outside the scope of this specification.

After a packet enters the OMNI interface, one or more outbound underlying interfaces are selected based on PBM traffic attributes, and one or more neighbor underlying interfaces are selected based on the receipt of Interface Attributes sub-options in IPv6 ND messages (see: Figure 9). Underlying interface selection for the nodes own local interfaces are based on attributes such as DSCP, application port number, cost, performance, message size, etc. OMNI interface multilink selections could also be configured to perform replication across multiple underlying interfaces for increased reliability at the expense of packet duplication. The set of all Interface Attributes received in IPv6 ND messages determines the multilink forwarding profile for selecting the neighbor's underlying interfaces.

When the OMNI interface sends a packet over a selected outbound underlying interface, the OAL includes or omits a mid-layer encapsulation header as necessary as discussed in Section 5 and as determined by the L2 address information received in Interface Attributes. The OAL also performs encapsulation when the nearest AR is located multiple hops away as discussed in Section 14.1. (Note that the OAL MAY employ packing when multiple packets are available for forwarding to the same destination.)

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

13.1. Multiple OMNI Interfaces

MNs may connect to multiple independent OMNI links concurrently in support of SBM. Each OMNI interface is distinguished by its Anycast ULA (e.g., [ULA]:0002::, [ULA]:1000::, [ULA]:7345::, etc.). The MN configures a separate OMNI interface for each link so that multiple interfaces (e.g., omni0, omni1, omni2, etc.) are exposed to the IPv6 layer. A different Anycast ULA is assigned to each interface, and the MN injects the service prefixes for the OMNI link instances into the EUN routing system.

Applications in EUNs can use Segment Routing to select the desired OMNI interface based on SBM considerations. The Anycast ULA is written into the IPv6 destination address, and the actual destination (along with any additional intermediate hops) is written into the Segment Routing Header. Standard IP routing directs the packets to the MN's mobile router entity, and the Anycast ULA identifies the OMNI interface to be used for transmission to the next hop. When the MN receives the message, it replaces the IPv6 destination address

with the next hop found in the routing header and transmits the message over the OMNI interface identified by the Anycast ULA.

Multiple distinct OMNI links can therefore be used to support fault tolerance, load balancing, reliability, etc. The architectural model is similar to Layer 2 Virtual Local Area Networks (VLANs).

13.2. MN<->AR Traffic Loop Prevention

After an AR has registered an MNP for a MN (see: Section 14), the AR will forward packets destined to an address within the MNP to the MN. The MN will under normal circumstances then forward the packet to the correct destination within its internal networks.

If at some later time the MN loses state (e.g., after a reboot), it may begin returning packets destined to an MNP address to the AR as its default router. The AR therefore must drop any packets originating from the MN and destined to an address within the MN's registered MNP. To do so, the AR institutes the following check:

o if the IP destination address belongs to a neighbor on the same OMNI interface, and if the link-layer source address is the same as one of the neighbor's link-layer addresses, drop the packet.

14. Router Discovery and Prefix Registration

MNs interface with the MS by sending RS messages with OMNI options under the assumption that one or more AR on the *NET will process the message and respond. The MN then configures default routes for the OMNI interface via the discovered ARs as the next hop. The manner in which the *NET ensures AR coordination is link-specific and outside the scope of this document (however, considerations for *NETs that do not provide ARs that recognize the OMNI option are discussed in Section 19).

For each underlying interface, the MN sends an RS message with an OMNI option to coordinate with MSEs identified by MSID values. Example MSID discovery methods are given in [RFC5214] and include data link login parameters, name service lookups, static configuration, a static "hosts" file, etc. The MN can also send an RS with an MS-Register sub-option that includes the Anycast MSID value '0', i.e., instead of or in addition to any non-zero MSIDs. When the AR receives an RS with a MSID '0', it selects a nearby MSE (which may be itself) and returns an RA with the selected MSID in an MS-Register sub-option. The AR selects only a single wildcard MSE (i.e., even if the RS MS-Register sub-option included multiple '0' MSIDs) while also soliciting the MSEs corresponding to any non-zero MSIDs.

MNs configure OMNI interfaces that observe the properties discussed in the previous section. The OMNI interface and its underlying 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 or DOWN through administrative action and/or through state transitions of the underlying interfaces. When a first underlying interface transitions to UP, the OMNI interface also transitions to UP. When all underlying interfaces transition to DOWN, the OMNI interface also transitions to DOWN.

When an OMNI interface transitions to UP, the MN sends RS messages to register its MNP and an initial set of underlying interfaces that are also UP. The MN sends additional RS messages to refresh lifetimes and to register/deregister underlying interfaces as they transition to UP or DOWN. The MN sends initial RS messages over an UP underlying interface with its MNP-LLA as the source and with destination set to All-Routers multicast (ff02::2) [RFC4291]. The RS messages include an OMNI option per Section 11 with a Preflen assertion, Interface Attributes appropriate for underlying interfaces, MS-Register/Release sub-options containing MSID values, and with any other necessary OMNI sub-options (e.g., a DUID sub-option as an identity for the MN). The S/T-omIndex field is set to the index of the underlying interface over which the RS message is sent.

ARs process IPv6 ND messages with OMNI options and act as an MSE themselves and/or as a proxy for other MSEs. ARs receive RS messages and create a neighbor cache entry for the MN, then coordinate with any MSEs named in the Register/Release lists in a manner outside the scope of this document. When an MSE processes the OMNI information, it first validates the prefix registration information then injects/withdraws the MNP in the routing/mapping system and caches/discards the new Preflen, MNP and Interface Attributes. The MSE then informs the AR of registration success/failure, and the AR returns an RA message to the MN with an OMNI option per Section 11.

The AR returns the RA message via the same underlying interface of the MN over which the RS was received, and with destination address set to the MNP-LLA (i.e., unicast), with source address set to its own LLA, and with an OMNI option with S/T-omIndex set to the value included in the RS. The OMNI option also includes a Preflen confirmation, Interface Attributes, MS-Register/Release and any other necessary OMNI sub-options (e.g., a DUID sub-option as an identity for the AR). The RA also includes any information for the link, including RA Cur Hop Limit, M and O flags, Router Lifetime, Reachable Time and Retrans Timer values, and includes any necessary options such as:

- o PIOs with (A; L=0) that include MSPs for the link [RFC8028].

- o RIOs [RFC4191] with more-specific routes.

- o an MTU option that specifies the maximum acceptable packet size for this underlying interface.

The AR MAY also send periodic and/or event-driven unsolicited RA messages per [RFC4861]. In that case, the S/T-omIndex field in the OMNI option of the unsolicited RA message identifies the target underlying interface of the destination MN.

The AR can combine the information from multiple MSEs into one or more "aggregate" RAs sent to the MN in order conserve *NET bandwidth. Each aggregate RA includes an OMNI option with MS-Register/Release sub-options with the MSEs represented by the aggregate. If an aggregate is sent, the RA message contents must consistently represent the combined information advertised by all represented MSEs. Note that since the AR uses its own ADM-LLA as the RA source address, the MN determines the addresses of the represented MSEs by examining the MS-Register/Release OMNI sub-options.

When the MN receives the RA message, it creates an OMNI interface neighbor cache entry for each MSID that has confirmed MNP registration via the L2 address of this AR. If the MN connects to multiple *NETs, it records the additional L2 AR addresses in each MSID neighbor cache entry (i.e., as multilink neighbors). The MN then configures a default route via the MSE that returned the RA message, and assigns the Subnet Router Anycast address corresponding to the MNP (e.g., 2001:db8:1:2::) to the OMNI interface. The MN then manages its underlying interfaces according to their states as follows:

- o When an underlying interface transitions to UP, the MN sends an RS over the underlying interface with an OMNI option. The OMNI option contains at least one Interface Attribute sub-option with values specific to this underlying interface, and may contain additional Interface Attributes specific to other underlying interfaces. The option also includes any MS-Register/Release sub-options.

- o When an underlying interface transitions to DOWN, the MN sends an RS or unsolicited NA message over any UP underlying interface with an OMNI option containing an Interface Attribute sub-option for the DOWN underlying interface with Link set to '0'. The MN sends an RS when an acknowledgement is required, or an unsolicited NA when reliability is not thought to be a concern (e.g., if

redundant transmissions are sent on multiple underlying interfaces).

o When the Router Lifetime for a specific AR nears expiration, the MN sends an RS over the underlying interface to receive a fresh RA. If no RA is received, the MN can send RS messages to an alternate MSID in case the current MSID has failed. If no RS messages are received even after trying to contact alternate MSIDs, the MN marks the underlying interface as DOWN.

o When a MN wishes to release from one or more current MSIDs, it sends an RS or unsolicited NA message over any UP underlying interfaces with an OMNI option with a Release MSID. Each MSID then withdraws the MNP from the routing/mapping system and informs the AR that the release was successful.

o When all of a MNs underlying interfaces have transitioned to DOWN (or if the prefix registration lifetime expires), any associated MSEs withdraw the MNP the same as if they had received a message with a release indication.

The MN 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 underlying interface (i.e., even after attempting to contact alternate MSEs), the MN declares this underlying interface as DOWN.

The IPv6 layer sees the OMNI interface as an ordinary IPv6 interface. Therefore, when the IPv6 layer sends an RS message the OMNI interface returns an internally-generated RA message as though the message originated from an IPv6 router. The internally-generated RA message contains configuration information that is consistent with the information received from the RAs generated by the MS. Whether the OMNI interface IPv6 ND messaging process is initiated from the receipt of an RS message from the IPv6 layer is an implementation matter. Some implementations may elect to defer the IPv6 ND messaging process until an RS is received from the IPv6 layer, while others may elect to initiate the process proactively. Still other deployments may elect to administratively disable the ordinary RS/RA messaging used by the IPv6 layer over the OMNI interface, since they are not required to drive the internal RS/RA processing. (Note that this same logic applies to IPv4 implementations that employ ICMP-based Router Discovery per [RFC1256].)

Note: The Router Lifetime value in RA messages indicates the time before which the MN must send another RS message over this underlying 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., REACHABLETIME seconds). ARs are therefore responsible for keeping MS state alive on a shorter timescale than the MN is required to do on its own behalf.

Note: On multicast-capable underlying interfaces, MNs should send periodic unsolicited multicast NA messages and ARs 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 a unicast exchange to test reachability.

Note: if an AR acting as a proxy forwards a MN's RS message to another node acting as an MSE using UDP/IP encapsulation, it must use a distinct UDP source port number for each MN. This allows the MSE to distinguish different MNs behind the same AR at the link-layer, whereas the link-layer addresses would otherwise be indistinguishable.

Note: when an AR acting as an MSE returns an RA to an INET Client, it includes an OMNI option with an Interface Attributes sub-option with omIndex set to 0 and with SRT, FMT, LHS and L2ADDR information for its INET interface. This provides the Client with partition prefix context regarding the local OMNI link segment.

14.1. Router Discovery in IP Multihop and IPv4-Only Networks

On some *NETs, a MN may be located multiple IP hops away from the nearest AR. Forwarding through IP multihop *NETs is conducted through the application of a routing protocol (e.g., a MANET/VANET routing protocol over omni-directional wireless interfaces, an inter-domain routing protocol in an enterprise network, etc.). These *NETs could be either IPv6-enabled or IPv4-only, while IPv4-only *NETs could be either multicast-capable or unicast-only (note that for IPv4-only *NETs the following procedures apply for both single-hop and multihop cases).

A MN located potentially multiple *NET hops away from the nearest AR prepares an RS message with source address set to either its MNP-LLA or a Temporary LLA, and with destination set to link-scoped All-Routers multicast the same as discussed above. For IPv6-enabled *NETs, the MN then encapsulates the message in an IPv6 header with source address set to the ULA corresponding to the LLA source address and with destination set to either a unicast or anycast ADM-ULA. For IPv4-only *NETs, the MN instead encapsulates the RS message in an IPv4 header with source address set to the node's own IPv4 address and with destination address set to either the unicast IPv4 address of an AR [RFC5214] or an IPv4 anycast address reserved for OMNI. The

MN then sends the encapsulated RS message via the *NET interface, where it will be forwarded by zero or more intermediate *NET hops.

When an intermediate *NET hop that participates in the routing protocol receives the encapsulated RS, it forwards the message according to its routing tables (note that an intermediate node could be a fixed infrastructure element or another MN). This process repeats iteratively until the RS message is received by a penultimate *NET hop within single-hop communications range of an AR, which forwards the message to the AR.

When the AR receives the message, it decapsulates the RS and coordinates with the MS the same as for an ordinary link-local RS, since the inner Hop Limit will not have been decremented by the multihop forwarding process. The AR then prepares an RA message with source address set to its own ADM-LLA and destination address set to the LLA of the original MN, then encapsulates the message in an IPv4/IPv6 header with source address set to its own IPv4/ULA address and with destination set to the encapsulation source of the RS.

The AR then forwards the message to an *NET node within communications range, which forwards the message according to its routing tables to an intermediate node. The multihop forwarding process within the *NET continues repetitively until the message is delivered to the original MN, which decapsulates the message and performs autoconfiguration the same as if it had received the RA directly from the AR as an on-link neighbor.

Note: An alternate approach to multihop forwarding via IPv6 encapsulation would be to statelessly translate the IPv6 LLAs into ULAs and forward the messages without encapsulation. This would violate the [RFC4861] requirement that certain IPv6 ND messages must use link-local addresses and must not be accepted if received with Hop Limit less than 255. This document therefore advocates encapsulation since the overhead is nominal considering the infrequent nature and small size of IPv6 ND messages. Future documents may consider encapsulation avoidance through translation while updating [RFC4861].

Note: An alternate approach to multihop forwarding via IPv4 encapsulation would be to employ IPv6/IPv4 protocol translation. However, for IPv6 ND messages the LLAs would be truncated due to translation and the OMNI Router and Prefix Discovery services would not be able to function. The use of IPv4 encapsulation is therefore indicated.

Note: An IPv4 anycast address for OMNI in IPv4 networks could be part of a new IPv4 /24 prefix allocation, but this may be difficult to

obtain given IPv4 address exhaustion. An alternative would be to re-purpose the prefix 192.88.99.0 which has been set aside from its former use by [RFC7526].

14.2. MS-Register and MS-Release List Processing

When a MN sends an RS message with an OMNI option via an underlying interface to an AR, the MN must convey its knowledge of its currently-associated MSEs. Initially, the MN will have no associated MSEs and should therefore include an MS-Register sub-option with the single MSID value 0 which requests the AR to select and assign an MSE. The AR will then return an RA message with source address set to the ADM-LLA of the selected MSE.

As the MN activates additional underlying interfaces, it can optionally include an MS-Register sub-option with MSID value 0, or with non-zero MSIDs for MSEs discovered from previous RS/RA exchanges. The MN will thus eventually begin to learn and manage its currently active set of MSEs, and can register with new MSEs or release from former MSEs with each successive RS/RA exchange. As the MN's MSE constituency grows, it alone is responsible for including or omitting MSIDs in the MS-Register/Release lists it sends in RS messages. The inclusion or omission of MSIDs determines the MN's interface to the MS and defines the manner in which MSEs will respond. The only limiting factor is that the MN should include no more than MAX_MSID values in each list per each IPv6 ND message, and should avoid duplication of entries in each list unless it wants to increase likelihood of control message delivery.

When an AR receives an RS message sent by a MN with an OMNI option, the option will contain zero or more MS-Register and MS-Release sub-options containing MSIDs. After processing the OMNI option, the AR will have a list of zero or more MS-Register MSIDs and a list of zero or more of MS-Release MSIDs. The AR then processes the lists as follows:

- o For each list, retain the first MAX_MSID values in the list and discard any additional MSIDs (i.e., even if there are duplicates within a list).

- o Next, for each MSID in the MS-Register list, remove all matching MSIDs from the MS-Release list.

- o Next, proceed according to whether the AR's own MSID or the value 0 appears in the MS-Register list as follows:

  * If yes, send an RA message directly back to the MN and send a proxy copy of the RS message to each additional MSID in the MS-

Register list with the MS-Register/Release lists omitted. Then, send an unsolicited NA (uNA) message to each MSID in the MS-Release list with the MS-Register/Release lists omitted and with an OMNI option with S/T-omIndex set to 0.

* If no, send a proxy copy of the RS message to each additional MSID in the MS-Register list with the MS-Register list omitted. For the first MSID, include the original MS-Release list; for all other MSIDs, omit the MS-Release list.

Each proxy copy of the RS message will include an OMNI option and encapsulation header with the ADM-ULA of the AR as the source and the ADM-ULA of the Register MSE as the destination. When the Register MSE receives the proxy RS message, if the message includes an MS-Release list the MSE sends a uNA message to each additional MSID in the Release list. The Register MSE then sends an RA message back to the (Proxy) AR wrapped in an OMNI encapsulation header with source and destination addresses reversed, and with RA destination set to the MNP-LLA of the MN. When the AR receives this RA message, it sends a proxy copy of the RA to the MN.

Each uNA message (whether send by the first-hop AR or by a Register MSE) will include an OMNI option and an encapsulation header with the ADM-ULA of the Register MSE as the source and the ADM-ULA of the Release ME as the destination. The uNA informs the Release MSE that its previous relationship with the MN has been released and that the source of the uNA message is now registered. The Release MSE must then note that the subject MN of the uNA message is now "departed", and forward any subsequent packets destined to the MN to the Register MSE.

Note that it is not an error for the MS-Register/Release lists to include duplicate entries. If duplicates occur within a list, the AR will generate multiple proxy RS and/or uNA messages - one for each copy of the duplicate entries.

14.3. DHCPv6-based Prefix Registration

When a MN is not pre-provisioned with an MNP-LLA (or, when multiple MNPs are needed), it will require the AR to select MNPs on its behalf and set up the correct routing state within the MS. The DHCPv6 service [RFC8415] supports this requirement.

When an MN needs to have the AR select MNPs, it sends an RS message with a Temporary LLA as the source and with DHCPv6 Message sub-option containing a Client Identifier, one or more IA_PD options and a Rapid Commit option. The MN also sets the 'msg-type' field to "Solicit", and includes a 3-octet 'transaction-id'.

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

When the AR receives the DHCPv6 Reply, it adds routes to the routing system and creates MNP-LLAs based on the delegated MNPs. The AR then sends an RA back to the MN with the DHCPv6 Reply message included in an OMNI DHCPv6 message sub-option. If the RS message source address was a Temporary address, the AR includes one of the (newly-created) MNP-LLAs as the RA destination address. The MN then creates a default route, assigns Subnet Router Anycast addresses and uses the RA destination address as its primary MNP-LLA. The MN will then use this primary MNP-LLA as the source address of any IPv6 ND messages it sends as long as it retains ownership of the MNP.

Note: After a MN performs a DHCPv6-based prefix registration exchange with a first AR, it would need to repeat the exchange with each additional MSE it registers with. In that case, the MN supplies the MNP delegations received from the first AR in the IA_PD fields of a DHCPv6 message when it engages the additional MSEs.

15. Secure Redirection

If the *NET link model is multiple access, the AR is responsible for assuring that address duplication cannot corrupt the neighbor caches of other nodes on the link. When the MN sends an RS message on a multiple access *NET link, the AR verifies that the MN is authorized to use the address and returns an RA with a non-zero Router Lifetime only if the MN is authorized.

After verifying MN authorization and returning an RA, the AR MAY return IPv6 ND Redirect messages to direct MNs located on the same *NET link to exchange packets directly without transiting the AR. In that case, the MNs can exchange packets according to their unicast L2 addresses discovered from the Redirect message instead of using the dogleg path through the AR. In some *NET links, however, such direct communications may be undesirable and continued use of the dogleg path through the AR may provide better performance. In that case, the AR can refrain from sending Redirects, and/or MNs can ignore them.

16. AR and MSE Resilience

*NETs SHOULD deploy ARs in Virtual Router Redundancy Protocol (VRRP) [RFC5798] configurations so that service continuity is maintained even if one or more ARs fail. Using VRRP, the MN is unaware which of the (redundant) ARs 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.

MSEs 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 MSE Failures

In environments where fast recovery from MSE failure is required, ARs SHOULD use proactive Neighbor Unreachability Detection (NUD) in a manner that parallels Bidirectional Forwarding Detection (BFD) [RFC5880] to track MSE reachability. ARs 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 *NET links such as aeronautical radios) and can therefore be tuned for rapid response.

ARs perform proactive NUD for MSEs for which there are currently active MNs on the *NET. If an MSE fails, ARs can quickly inform MNs of the outage by sending multicast RA messages on the *NET interface. The AR sends RA messages to MNs via the *NET interface with an OMNI option with a Release ID for the failed MSE, and with destination address set to All-Nodes multicast (ff02::1) [RFC4291].

The AR SHOULD send MAX_FINAL_RTR_ADVERTISEMENTS RA messages separated by small delays [RFC4861]. Any MNs on the *NET interface that have been using the (now defunct) MSE will receive the RA messages and associate with a new MSE.

18. Transition Considerations

When a MN connects to an *NET link for the first time, it sends an RS message with an OMNI option. If the first hop AR recognizes the option, it returns an RA with its ADM-LLA as the source, the MNP-LLA as the destination and with an OMNI option included. The MN then engages the AR according to the OMNI link model specified above. If

the first hop AR is a legacy IPv6 router, however, it instead returns an RA message with no OMNI option and with a non-OMNI unicast source LLA as specified in [RFC4861]. In that case, the MN engages the *NET according to the legacy IPv6 link model and without the OMNI extensions specified in this document.

If the *NET link model is multiple access, there must be assurance that address duplication cannot corrupt the neighbor caches of other nodes on the link. When the MN sends an RS message on a multiple access *NET link with an LLA source address and an OMNI option, ARs that recognize the option ensure that the MN is authorized to use the address and return an RA with a non-zero Router Lifetime only if the MN is authorized. ARs that do not recognize the option instead return an RA that makes no statement about the MN's authorization to use the source address. In that case, the MN should perform Duplicate Address Detection to ensure that it does not interfere with other nodes on the link.

An alternative approach for multiple access *NET links to ensure isolation for MN / AR communications is through L2 address mappings as discussed in Appendix C. This arrangement imparts a (virtual) point-to-point link model over the (physical) multiple access link.

19. OMNI Interfaces on Open Internetworks

OMNI interfaces configured over IPv6-enabled underlying interfaces on an open Internetwork without an OMNI-aware first-hop AR receive RA messages that do not include an OMNI option, while OMNI interfaces configured over IPv4-only underlying interfaces do not receive any (IPv6) RA messages at all. OMNI interfaces that receive RA messages without an OMNI option configure addresses, on-link prefixes, etc. on the underlying interface that received the RA according to standard IPv6 ND and address resolution conventions [RFC4861] [RFC4862]. OMNI interfaces configured over IPv4-only underlying interfaces configure IPv4 address information on the underlying interfaces using mechanisms such as DHCPv4 [RFC2131].

OMNI interfaces configured over underlying interfaces that connect to an open Internetwork can apply security services such as VPNs to connect to an MSE, or can establish a direct link to an MSE through some other means (see Section 4). In environments where an explicit VPN or direct link may be impractical, OMNI interfaces can instead use UDP/IP encapsulation per [RFC6081][RFC4380] and HIP-based message authentication per [RFC7401].

For "Vehicle-to-Infrastructure (V2I)" coordination, the MN codes a HIP "Initiator" message in an OMNI option of an IPv6 RS message and the MSE responds with a HIP "Responder" message coded in an OMNI

option of an IPv6 RA message. HIP security services are applied per [RFC7401], using the RS/RA messages as simple "shipping containers" to convey the HIP parameters. In that case, a "two-message HIP exchange" through a single RS/RA exchange may be sufficient for mutual authentication. For "Vehicle-to-Vehicle (V2V)" coordination, two MNs can coordinate directly with one another with HIP "Initiator/Responder" messages coded in OMNI options of IPv6 NS/NA messages. In that case, a four-message HIP exchange (i.e., two back-to-back NS/NA exchanges) may be necessary for the two MNs to attain mutual authentication.

After establishing a VPN or preparing for UDP/IP encapsulation, OMNI interfaces send control plane messages to interface with the MS, including RS/RA messages used according to Section 14 and NS/NA messages used for route optimization and mobility (see: [I-D.templin-intarea-6706bis]). The control plane messages must be authenticated while data plane messages are delivered the same as for ordinary best-effort traffic with basic source address-based data origin verification. Data plane communications via OMNI interfaces that connect over open Internetworks without an explicit VPN should therefore employ transport- or higher-layer security to ensure integrity and/or confidentiality.

OMNI interfaces configured over open Internetworks are often located behind NATs. The OMNI interface accommodates NAT traversal using UDP/IP encapsulation and the mechanisms discussed in [I-D.templin-intarea-6706bis].

Note: Following the initial HIP Initiator/Responder exchange, OMNI interfaces configured over open Internetworks maintain HIP associations through the transmission of IPv6 ND messages that include OMNI options with HIP "Update" and "Notify" messages. OMNI interfaces use the HIP "Update" message when an acknowledgement is required, and use the "Notify" message in unacknowledged isolated IPv6 ND messages (e.g., unsolicited NAs).

20. Time-Varying MNPs

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

The prefix delegation services discussed in Section 14.3 allows OMNI MNs that desire time-varying MNPs to obtain short-lived prefixes to

use a Temporary LLA as the source address of an RS message with an OMNI option with DHCPv6 Option sub-options. The MN would then be obligated to renumber its internal networks whenever its MNP (and therefore also its OMNI address) changes. This should not present a challenge for MNs with automated network renumbering services, however presents limits for the durations of ongoing sessions that would prefer to use a constant address.

21. Using (H)HITs Instead of Temporary Addresses

MNs that generate (H)HITs but do not have pre-assigned MNPs can request MNP delegations by issuing IPv6 ND messages that use the (H)HIT instead of a Temporary address. In particular, when a MN creates an RS message it can set the source address to the unspecified address (::) and encapsulate the message in an IPv6 header with the (H)HIT as the source address. The MN sets the T field in the OMNI option to 0 (since the address is not Temporary) and sends the message to the AR as specified in Section 14.

When the AR receives the message, it examines the RS encapsulation source address to determine that the source is a (H)HIT and not a ULA. The AR next invokes the DHCPv6 protocol to request an MNP prefix delegation, then prepares an RA message with source address set to its own ADM-LLA and destination set to the MNP-LLA corresponding to the delegated MNP. The AR finally encapsulates the RA in an IPv6 header with source address set to its own ADM-ULA and destination set to the (H)HIT from the RS encapsulation source address, then returns the encapsulated RA to the MN.

22. IANA Considerations

The IANA has assigned a 4-octet Private Enterprise Number (PEN) code "45282" in the "enterprise-numbers" registry. This document is the normative reference for using this code in DHCP Unique IDentifiers based on Enterprise Numbers (DUID-EN) for OMNI Interfaces (see: Section 10).

The IANA is instructed to allocate an official Type number TBD1 from the registry "IPv6 Neighbor Discovery Option Formats" for the OMNI option. Implementations set Type to 253 as an interim value [RFC4727].

The IANA is instructed to assign a new Code value "1" in the "ICMPv6 Code Fields: Type 2 - Packet Too Big" registry. The registry should read as follows:

Code	Name	Reference
0	Diagnostic Packet Too Big	[RFC4443]
1	Advisory Packet Too Big	[RFCXXXX]

Figure 21: ICMPv6 Code Fields: Type 2 - Packet Too Big Values

The IANA is instructed to allocate one Ethernet unicast address TBD2 (suggest 00-00-5E-00-52-14 [RFC5214]) in the registry "IANA Ethernet Address Block - Unicast Use".

The OMNI option defines a 5-bit Sub-Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI option Sub-Type values". Initial values for the OMNI option Sub-Type values registry are given below; future assignments are to be made through Expert Review [RFC8126].

Value	Sub-Type name	Reference
0	Pad1	[RFCXXXX]
1	PadN	[RFCXXXX]
2	Interface Attributes (Type 1)	[RFCXXXX]
3	Interface Attributes (Type 2)	[RFCXXXX]
4	Traffic Selector	[RFCXXXX]
5	Origin Indication	[RFCXXXX]
6	MS-Register	[RFCXXXX]
7	MS-Release	[RFCXXXX]
8	Geo Coordinates	[RFCXXXX]
9	DHCPv6 Message	[RFCXXXX]
10	HIP Message	[RFCXXXX]
11	Node Identification	[RFCXXXX]
12-29	Unassigned	
30	Experimental	[RFCXXXX]
31	Reserved	[RFCXXXX]

Figure 22: OMNI Option Sub-Type Values

The OMNI Node Identification Sub-Option (see: Section 11.1.12) contains an 8-bit ID-Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI Node Identification Sub-Option ID-Type values". Initial values for the OMNI Node Identification Sub-Option ID Type values registry are given below; future assignments are to be made through Expert Review [RFC8126].

Value	Sub-Type name	Reference
0	UUID	[RFCXXXX]
1	HIT	[RFCXXXX]
2	HHIT	[RFCXXXX]
3	Network Access Identifier	[RFCXXXX]
4	FQDN	[RFCXXXX]
5-252	Unassigned	[RFCXXXX]
253-254	Experimental	[RFCXXXX]
255	Reserved	[RFCXXXX]

Figure 23: OMNI Node Identification Sub-Option ID-Type Values

23. Security Considerations

Security considerations for IPv4 [RFC0791], IPv6 [RFC8200] and IPv6 Neighbor Discovery [RFC4861] apply. OMNI interface IPv6 ND messages SHOULD include Nonce and Timestamp options [RFC3971] when transaction confirmation and/or time synchronization is needed.

MN OMNI interfaces configured over secured ANET interfaces inherit the physical and/or link-layer security properties (i.e., "protected spectrum") of the connected ANETs. MN OMNI interfaces configured over open INET interfaces can use symmetric securing services such as VPNs or can by some other means establish a direct link. When a VPN or direct link may be impractical, however, the security services specified in [RFC7401] can be employed. While the OMNI link protects control plane messaging, applications must still employ end-to-end transport- or higher-layer security services to protect the data plane.

Strong network layer security for control plane messages and forwarding path integrity for data plane messages between MSEs MUST be supported. In one example, the AERO service [I-D.templin-intarea-6706bis] constructs a spanning tree between MSEs and secures the links in the spanning tree with network layer security mechanisms such as IPsec [RFC4301] or Wireguard. Control plane messages are then constrained to travel only over the secured spanning tree paths and are therefore protected from attack or eavesdropping. Since data plane messages can travel over route optimized paths that do not strictly follow the spanning tree, however, end-to-end transport- or higher-layer security services are still required.

Identity-based key verification infrastructure services such as iPSK may be necessary for verifying the identities claimed by MNs. This requirement should be harmonized with the manner in which (H)HITs are attested 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 5.1.

24. Implementation Status

Draft -29 is implemented in the recently tagged AERO/OMNI 3.0.0 internal release, and Draft -30 is now tagged as the AERO/OMNI 3.0.1. Newer specification versions will be tagged in upcoming releases. First public release expected before the end of 2020.

25. 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: Stuart Card, Michael Matyas, Robert Moskowitz, Madhu Niraula, Greg Saccone, Stephane Tamalet, Eric Vyncke. Pavel Drasil, Zdenek Jaron and Michal Skorepa are especially recognized for their many helpful ideas and suggestions. Madhuri Madhava Badgandi, Sean Dickson, Don Dillenburg, Joe Dudkowski, Vijayasarathy Rajagopalan, Ron Sackman 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.

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.

26. References

26.1. Normative References

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

[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>.

[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>.

[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>.

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

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

[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>.

[RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <https://www.rfc-editor.org/info/rfc4727>.

[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>.

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

[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>.

26.2. 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/", March 2020.

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

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

[I-D.ietf-6man-rfc4941bis] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", draft-ietf-6man-rfc4941bis-12 (work in progress), November 2020.

[I-D.ietf-drip-rid] Moskowitz, R., Card, S., Wiethuechter, A., and A. Gurtov, "UAS Remote ID", draft-ietf-drip-rid-06 (work in progress), December 2020.

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

[I-D.ietf-ipwave-vehicular-networking] Jeong, J., "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases", draft-ietf-ipwave-vehicular-networking-19 (work in progress), July 2020.

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

[I-D.templin-6man-lla-type]
Templin, F., "The IPv6 Link-Local Address Type Field", draft-templin-6man-lla-type-02 (work in progress), November 2020.

[I-D.templin-intarea-6706bis]
Templin, F., "Asymmetric Extended Route Optimization (AERO)", draft-templin-intarea-6706bis-87 (work in progress), January 2021.

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

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

[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>.

[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>.

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

[RFC2225] Laubach, M. and J. Halpern, "Classical IP and ARP over ATM", RFC 2225, DOI 10.17487/RFC2225, April 1998, <https://www.rfc-editor.org/info/rfc2225>.

[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>.

[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>.

[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>.

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

[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>.

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

[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>.

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

[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>.

[RFC3879] Huitema, C. and B. Carpenter, "Deprecating Site Local Addresses", RFC 3879, DOI 10.17487/RFC3879, September 2004, <https://www.rfc-editor.org/info/rfc3879>.

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

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

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

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

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

[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>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/info/rfc4821>.

[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>.

[RFC5175] Haberman, B., Ed. and R. Hinden, "IPv6 Router Advertisement Flags Option", RFC 5175, DOI 10.17487/RFC5175, March 2008, <https://www.rfc-editor.org/info/rfc5175>.

[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>.

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

[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>.

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

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

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

[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>.

[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>.

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <https://www.rfc-editor.org/info/rfc7084>.

[RFC7421] Carpenter, B., Ed., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, DOI 10.17487/RFC7421, January 2015, <https://www.rfc-editor.org/info/rfc7421>.

[RFC7526] Troan, O. and B. Carpenter, Ed., "Deprecating the Anycast Prefix for 6to4 Relay Routers", BCP 196, RFC 7526, DOI 10.17487/RFC7526, May 2015, <https://www.rfc-editor.org/info/rfc7526>.

[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>.

[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>.

[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>.

[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>.

Appendix A. Interface Attribute Preferences Bitmap Encoding

Adaptation of the OMNI option Interface Attributes Preferences Bitmap encoding to specific Internetworks such as the Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS) may include link selection preferences based on other traffic classifiers (e.g., transport port numbers, etc.) in addition to the existing DSCP-based preferences. Nodes on specific Internetworks maintain a map of traffic classifiers to additional P[*] preference fields beyond the first 64. For example, TCP port 22 maps to P[67], TCP port 443 maps to P[70], UDP port 8060 maps to P[76], etc.

Implementations use Simplex or Indexed encoding formats for P[*] encoding in order to encode a given set of traffic classifiers in the most efficient way. Some use cases may be more efficiently coded using Simplex form, while others may be more efficient using Indexed. Once a format is selected for preparation of a single Interface Attribute the same format must be used for the entire Interface Attribute sub-option. Different sub-options may use different formats.

The following figures show coding examples for various Simplex and Indexed formats:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-Type=3|    Sub-length=N   |    omIndex    |    omType     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Provider ID  | Link  |R| API | Bitmap(0)=0xff|P00|P01|P02|P03|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P04|P05|P06|P07|P08|P09|P10|P11|P12|P13|P14|P15|P16|P17|P18|P19|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P20|P21|P22|P23|P24|P25|P26|P27|P28|P29|P30|P31| Bitmap(1)=0xff|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P32|P33|P34|P35|P36|P37|P38|P39|P40|P41|P42|P43|P44|P45|P46|P47|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P48|P49|P50|P51|P52|P53|P54|P55|P56|P57|P58|P59|P60|P61|P62|P63|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Bitmap(2)=0xff|P64|P65|P67|P68| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 24: Example 1: Dense Simplex Encoding

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-Type=3|    Sub-length=N   |    omIndex     |    omType     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Provider ID  | Link  |R| API | Bitmap(0)=0x00| Bitmap(1)=0x0f|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P48|P49|P50|P51|P52|P53|P54|P55|P56|P57|P58|P59|P60|P61|P62|P63|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Bitmap(2)=0x00| Bitmap(3)=0x00| Bitmap(4)=0x00| Bitmap(5)=0x00|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Bitmap(6)=0xf0|192|193|194|195|196|197|198|199|200|201|202|203|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|204|205|206|207| Bitmap(7)=0x00| Bitmap(8)=0x0f|272|273|274|275|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|276|277|278|279|280|281|282|283|284|285|286|287| Bitmap(9)=0x00|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Bitmap(10)=0x00| ...
+-+-+-+-+-+-+-+-+-+-+-
```

Figure 25: Example 2: Sparse Simplex Encoding

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-Type=3|    Sub-length=N   |    omIndex     |    omType     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Provider ID  | Link  |R| API |  Index = 0x00 | Bitmap = 0x80 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P00|P01|P02|P03|  Index = 0x01 | Bitmap = 0x01 |P60|P61|P62|P63|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Index = 0x10 | Bitmap = 0x80 |512|513|514|515|  Index = 0x18 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Bitmap = 0x01 |796|797|798|799| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 26: Example 3: Indexed Encoding

Appendix B. 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 layer 2 "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 C. MN / AR Isolation Through L2 Address Mapping

Per [RFC4861], IPv6 ND messages may be sent to either a multicast or unicast link-scoped IPv6 destination address. However, IPv6 ND messaging should be coordinated between the MN and AR only without invoking other nodes on the *NET. This implies that MN / AR control messaging should be isolated and not overheard by other nodes on the link.

To support MN / AR isolation on some *NET links, ARs can maintain an OMNI-specific unicast L2 address ("MSADDR"). For Ethernet-compatible *NETs, this specification reserves one Ethernet unicast address TBD2 (see: Section 22). For non-Ethernet statically-addressed *NETs, MSADDR is reserved per the assigned numbers authority for the *NET addressing space. For still other *NETs, MSADDR may be dynamically discovered through other means, e.g., L2 beacons.

MNs 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 L2 address. In this way, all of the MN's IPv6 ND messages will be received by ARs that are configured to accept packets destined to MSADDR. Note that multiple ARs on the link could be configured to accept packets destined to MSADDR, e.g., as a basis for supporting redundancy.

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

Appendix D. Change Log

<< RFC Editor - remove prior to publication >>

Differences from draft-templin-6man-omni-interface-35 to draft-templin-6man-omni-interface-36:

- Major clarifications on aspects such as "hard/soft" PTB error messages
- Made generic so that either IP protocol version (IPv4 or IPv6) can be used in the data plane.

Differences from draft-templin-6man-omni-interface-31 to draft-templin-6man-omni-interface-32:

- MTU
- Support for multi-hop ANETS such as ISATAP.

Differences from draft-templin-6man-omni-interface-29 to draft-templin-6man-omni-interface-30:

- Moved link-layer addressing information into the OMNI option on a per-ifIndex basis
- Renamed "ifIndex-tuple" to "Interface Attributes"

Differences from draft-templin-6man-omni-interface-27 to draft-templin-6man-omni-interface-28:

- Updates based on implementation experience.

Differences from draft-templin-6man-omni-interface-25 to draft-templin-6man-omni-interface-26:

- Further clarification on "aggregate" RA messages.
- Expanded Security Considerations to discuss expectations for security in the Mobility Service.

Differences from draft-templin-6man-omni-interface-20 to draft-templin-6man-omni-interface-21:

- Safety-Based Multilink (SBM) and Performance-Based Multilink (PBM).

Differences from draft-templin-6man-omni-interface-18 to draft-templin-6man-omni-interface-19:

- SEND/CGA.

Differences from draft-templin-6man-omni-interface-17 to draft-templin-6man-omni-interface-18:

- Teredo

Differences from draft-templin-6man-omni-interface-14 to draft-templin-6man-omni-interface-15:

- Prefix length discussions removed.

Differences from draft-templin-6man-omni-interface-12 to draft-templin-6man-omni-interface-13:

- Teredo

Differences from draft-templin-6man-omni-interface-11 to draft-templin-6man-omni-interface-12:

- Major simplifications and clarifications on MTU and fragmentation.

- Document now updates RFC4443 and RFC8201.

Differences from draft-templin-6man-omni-interface-10 to draft-templin-6man-omni-interface-11:

- Removed /64 assumption, resulting in new OMNI address format.

Differences from draft-templin-6man-omni-interface-07 to draft-templin-6man-omni-interface-08:

- OMNI MNs in the open Internet

Differences from draft-templin-6man-omni-interface-06 to draft-templin-6man-omni-interface-07:

- Brought back L2 MSADDR mapping text for MN / AR isolation based on L2 addressing.

- Expanded "Transition Considerations".

Differences from draft-templin-6man-omni-interface-05 to draft-templin-6man-omni-interface-06:

- Brought back OMNI option "R" flag, and discussed its use.

Differences from draft-templin-6man-omni-interface-04 to draft-templin-6man-omni-interface-05:

- Transition considerations, and overhaul of RS/RA addressing with the inclusion of MSE addresses within the OMNI option instead of as RS/RA addresses (developed under FAA SE2025 contract number DTFAWA-15-D-00030).

Differences from draft-templin-6man-omni-interface-02 to draft-templin-6man-omni-interface-03:

- Added "advisory PTB messages" under FAA SE2025 contract number DTFAWA-15-D-00030.

Differences from draft-templin-6man-omni-interface-01 to draft-templin-6man-omni-interface-02:

- Removed "Primary" flag and supporting text.

- Clarified that "Router Lifetime" applies to each ANET interface independently, and that the union of all ANET interface Router Lifetimes determines MSE lifetime.

Differences from draft-templin-6man-omni-interface-00 to draft-templin-6man-omni-interface-01:

- "All-MSEs" OMNI LLA defined. Also reserved fe80::ff00:0000/104 for future use (most likely as "pseudo-multicast").

- Non-normative discussion of alternate OMNI LLA construction form made possible if the 64-bit assumption were relaxed.

First draft version (draft-templin-atn-aero-interface-00):

- Draft based on consensus decision of ICAO Working Group I Mobility Subgroup March 22, 2019.

Authors' Addresses

Fred L. Templin (editor)
The Boeing Company
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org

Tony Whyman
MWA Ltd c/o Inmarsat Global Ltd
99 City Road
London EC1Y 1AX
England

Email: tony.whyman@mccallumwhyman.com