       Transmission of IPv6 Packets over Overlay Multilink Network (OMNI)
                              Interfaces
                    draft-templin-atn-aero-interface-21

Abstract

   Mobile nodes (e.g., aircraft of various configurations, terrestrial
   vehicles, seagoing vessels, mobile enterprise 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 IPv6 packets over Overlay
   Multilink Network (OMNI) Interfaces.

Status of This Memo

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

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

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

   This Internet-Draft will expire on August 20, 2020.

Table of Contents

1. Introduction

   Mobile Nodes (MNs) (e.g., aircraft of various configurations,
   terrestrial vehicles, seagoing vessels, mobile enterprise devices,
   etc.) often have multiple data links 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 underlying
data link interfaces.

The MN configures a virtual interface (termed the "Overlay Multilink
Network (OMNI) interface") as a thin layer over the underlying access
network interfaces. The OMNI interface is therefore the only
interface abstraction exposed to the IPv6 layer and behaves according
to the Non-Broadcast, Multiple Access (NBMA) interface principle,
while underlying access network 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 a worldwide Internetwork 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 access network data links in
order to register per-link parameters (see Section 12).

The OMNI interface provides a multilink nexus for exchanging inbound
and outbound traffic via the correct underlying Access Network (ANET)
interface(s). The IPv6 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) from which OMNI link MNPs
are derived. If there are multiple OMNI links, the IPv6 layer will
see multiple OMNI interfaces.

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.

This document specifies the transmission of IPv6 packets [RFC8200]
and MN/MS control messaging over OMNI interfaces.

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.

Also, 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 defined in [RFC4291] (with Link-Local scope assumed).

The following terms are defined within the scope of this document:

Mobile Node (MN)
    an end system with multiple distinct upstream data link
    connections that are managed together as a single logical unit.
    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 IPv6 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 singluar or aggregate) that
    coordinates the mobility events of one or more MN.

Mobility Service Prefix (MSP)
    an aggregated IPv6 prefix (e.g., 2001:db8::/32) advertised to the
    rest of the Internetwork by the MS, and from which more-specific
    Mobile Network Prefixes (MNPs) are derived.

Mobile Network Prefix (MNP)
    a longer IPv6 prefix taken from the MSP (e.g.,
    2001:db8:1000:2000::/56) 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 between the MN and ANET are assumed.

Access Router (AR)

      a first-hop router in the ANET for connecting MNs to
      correspondents in outside Internetworks.

   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 for ANET MNs and INET
      correspondents. Examples include private enterprise networks,
      ground domain aviation service networks and the global public
      Internet itself.

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

   OMNI link
      a 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 ANET/INET interfaces.

   OMNI link local address (LLA)
      an IPv6 link-local address constructed as specified in Section 7,
      and assigned to an OMNI interface.

   OMNI Option
      an IPv6 Neighbor Discovery option providing multilink parameters
      for the OMNI interface as specified in Section 8.

   Multilink
      an OMNI interface's manner of managing diverse underlying data
      link interfaces as a single logical unit. The OMNI interface
      provides a single unified interface to upper layers, while
      underlying data link 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.

   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", "IPv6 layer", etc.

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.

   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 ANET 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 OMNI 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
   [RFC7847], and augmented as shown in Figure 1. The IP layer (L3)
   therefore sees the OMNI interface as a single network layer interface
   with multiple underlying ANET 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-      +---->|        (OMNI LLA)          |
            Physical          |     +----------------------------+
            Interface        +---->|  L2  |  L2  |        |  L2  |
            Binding               |(IF#1)|(IF#2)| ..... |(IF#n)|
                              +------+------+        +------+
                              |  L1  |  L1  |        |  L1  |
                              |      |      |        |      |
                              +------+------+        +------+
```
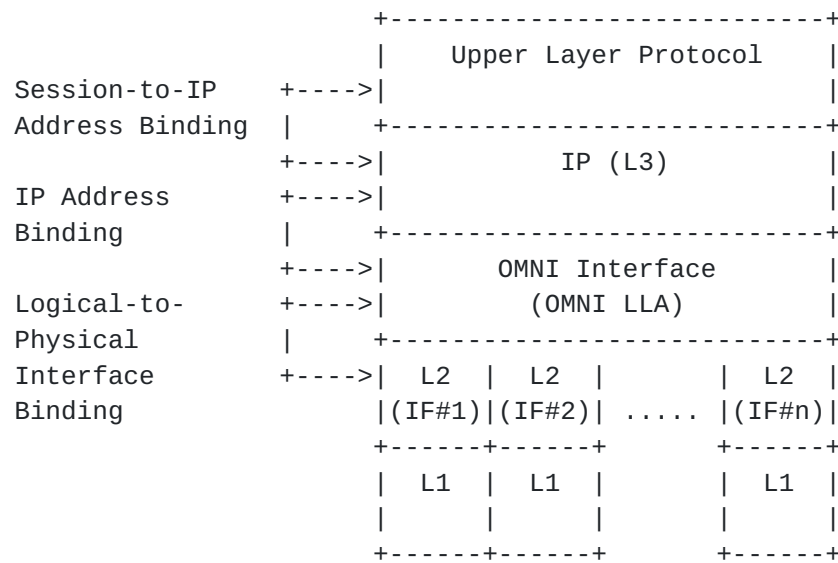
          Figure 1: OMNI Interface Architectural Layering Model

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

   o since OMNI LLAs are uniquely derived from an MNP, no Duplicate
      Address Detection (DAD) messaging is necessary over the OMNI
      interface.

   o ANET interfaces 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 OMNI LLA to all ANET interfaces.)

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

   o coordinating ANET interfaces in this way allows them to be
      represented in a unified MS profile with provisions for mobility
      and multilink operations.

   o 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.

   o 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.

   Other opportunities are discussed in [RFC7847].

   Figure 2 depicts the architectural model for a MN connecting to the
   MS via multiple independent ANETs. When an ANET interface becomes
   active, the MN's OMNI interface sends native (i.e., unencapsulated)
   IPv6 ND messages via the underlying ANET interface. IPv6 ND messages
   traverse the ground domain ANETs until they reach an Access Router
   (AR#1, AR#2, .., AR#n). The AR then coordinates with a Mobility
   Service Endpoint (MSE#1, MSE#2, ..., MSE#m) in the INET and returns
   an IPv6 ND message response to the MN. IPv6 ND messages traverse the
   ANET at layer 2; hence, the Hop Limit is not decremented.

```
                      +--------------+
                      |     MN       |
                      +--------------+
                      |OMNI interface|
                      +----+----+----+
              +--------|IF#1|IF#2|IF#n|------ +
             /         +----+----+----+        \
            /               |                   \
           /    <---- Native |  IP ---->          \
          v                  v                     v
       (:::)-.            (:::)-.               (:::)-.
    .-(::ANET:::)       .-(::ANET:::)         .-(::ANET:::)
      `-(:::::)-'         `-(:::::)-'           `-(:::::)-'
        +----+             +----+               +----+
   ...  |AR#1|  .........  |AR#2|   .........   |AR#n|  ...
   .    +-|--+             +-|--+               +-|--+      .
   .      |                  |                    |         .
   .      v                  v                    v         .
   .          <-----  Encapsulation ----->                  .
   .                                                        .
   .     +-----+             (:::)-.                        .
   .     |MSE#2|          .-(:::::::::)           +-----+   .
   .     +-----+        .-(:::   INET  ::::)-.     |MSE#m|   .
   .                   (:::::    Routing  ::::)   +-----+   .
   .                    `-(::: System :::)-'                .
   . +-----+              `-(:::::::::-'                    .
   . |MSE#1|          +-----+                 +-----+       .
   . +-----+          |MSE#3|                 |MSE#4|       .
   .                  +-----+                 +-----+       .
   .                                                        .
   .                                                        .
   .       <----- Worldwide Connected Internetwork ---->    .
    ........................................................
```
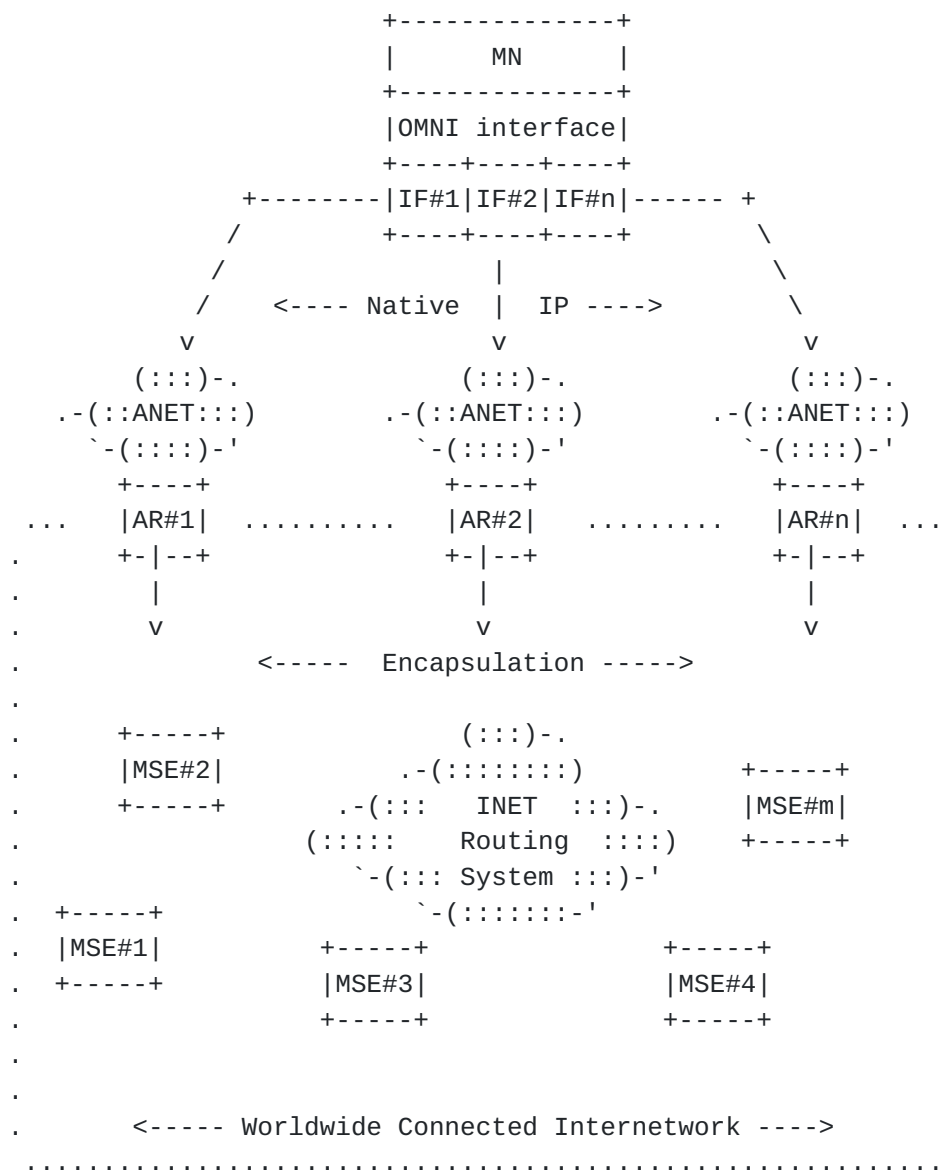
                Figure 2: MN/MS Coordination via Multiple ANETs

   After the initial IPv6 ND message exchange, the MN can send and
   receive unencapsulated IPv6 data packets over the OMNI interface.
   OMNI interface multilink services will forward the packets via ARs in
   the correct underlying ANETs. 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.

5. Maximum Transmission Unit (MTU) and Fragmentation

   All IPv6 interfaces are REQUIRED to configure a minimum Maximum
   Transmission Unit (MTU) of 1280 bytes [RFC8200]. 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].

   The OMNI interface configures an MTU of 9180 bytes [RFC2492]; the
   size is therefore not a reflection of the underlying ANET interface
   MTUs, but rather determines the largest packet the OMNI interface can
   forward or reassemble.

   The OMNI interface can employ link-layer IPv6 encapsulation and
   fragmentation/reassembly per [RFC2473], but its use is OPTIONAL since
   correct operation will result in either case. Implementations that
   omit link-layer IPv6 fragmentation/reassembly may be more prone to
   dropping large packets and returning a PTB, while those that include
   it may see performance increases at the expense of including
   additional code. In both cases, OMNI interface neighbors are
   responsible for advertising their willingness to reassemble.

   The OMNI interface returns internally-generated PTB messages for
   packets admitted into the interface that it deems too large for the
   outbound underlying ANET interface (e.g., according to ANET
   performance characteristics, MTU, etc). For all other packets, the
   OMNI interface performs PMTUD even if the destination appears to be
   on the same link since a proxy on the path could return a PTB
   message. This ensures that the path MTU is adaptive and reflects the
   current path used for a given data flow.

   The MN's OMNI interface forwards packets that are no larger than the
   MTU of the selected underlying ANET interface according to the ANET
   L2 frame format. When the OMNI interface forwards a packet that is
   larger than the ANET interface MTU, it drops the packet and returns a
   PTB if the AR is not willing to reassemble.

   Otherwise, the OMNI interface encapsulates the packet in an IPv6
   header with source address set to the MN's link-local address and
   destination address set to the link-local address of the MSE (see:
   Section 7). The OMNI interface then uses IPv6 fragmentation to break
   the encapsulated packet into fragments that are no larger than the
   ANET interface MTU and sends the fragments over the ANET where they
   will be intercepted by the AR. The AR then reassembles and conveys
   the packet toward the final destination.

   When an AR receives a fragmented or whole packet from the INET
   destined to an ANET MN, it first determines whether to forward or

drop and return a PTB. If the AR deems the packet to be of
acceptable size, it first reassembles locally (if necessary) then
forwards the packet to the MN. If the (reassembled) packet is no
larger than the ANET MTU, the AR forwards according to the ANET L2
frame format. If the packet is larger than the ANET MTU, the AR
instead uses link-layer IPv6 encapsulation and fragmentation as above
if the MN accepts fragments or drops and returns a PTB otherwise.
The MN then reassembles and discards the encapsulation header, then
forwards the whole packet to the final destination.

Applications that cannot tolerate loss due to MTU restrictions SHOULD
avoid sending packets larger than 1280 bytes, since dynamic path
changes can reduce the path MTU at any time. Applications that may
benefit from sending larger packets even though the path MTU may
change dynamically MAY use larger sizes (i.e., up to the OMNI
interface MTU).

Note that when the AR forwards a fragmented packet received from the
INET, it is imperative that the AR reassembles locally first instead
of blindly forwarding fragments directly to the MN to avoid attacks
such as tiny fragments, overlapping fragments, etc.

6. Frame Format

The OMNI interface transmits IPv6 packets according to the native
frame format of each underlying ANET 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

OMNI interfaces assign IPv6 Link-Local Addresses (i.e., "OMNI LLAs")
using the following constructs:

o   IPv6 MN OMNI LLAs encode the most-significant 64 bits of a MNP
    within the least-significant 64 bits (i.e., the interface ID) of a
    Link-Local IPv6 Unicast Address (see: [RFC4291], Section 2.5.6).
    For example, for the MNP 2001:db8:1000:2000::/56 the corresponding
    LLA is fe80::2001:db8:1000:2000.

o   IPv4-compatible MN OMNI LLAs are assigned as fe80::ffff:[v4addr],
    i.e., the most significant 10 bits of the prefix fe80::/10,
    followed by 70 '0' bits, followed by 16 '1' bits, followed by a
    32bit IPv4 address. For example, the IPv4-Compatible MN OMNI LLA

        for 192.0.2.1 is fe80::ffff:192.0.2.1 (also written as
        fe80::ffff:c000:0201).

   o MSE OMNI LLAs are assigned from the range fe80::/96, and MUST be
      managed for uniqueness. The lower 32 bits of the LLA includes a
      unique integer value between '1' and 'fffffffe', e.g., as in
      fe80::1, fe80::2, fe80::3, etc., fe80::ffff:fffe. The address
      fe80:: is the link-local Subnet-Router anycast address [RFC4291]
      and the address fe80::ffff:ffff is reserved. (Note that distinct
      OMNI link segments can avoid overlap by assigning MSE OMNI LLAs
      from unique fe80::/96 sub-prefixes. For example, a first segment
      could assign from fe80::1000/116, a second from fe80::2000/116, a
      third from fe80::3000/116, etc.)

   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 above OMNI LLA constructs.

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

8. 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.
   IPv6 Neighbor Discovery (ND) [RFC4861] messages on MN OMNI interfaces
   observe the native Source/Target Link-Layer Address Option (S/TLLAO)
   formats of the underlying ANET interfaces (e.g., for Ethernet the S/
   TLLAO is specified in [RFC2464]).

   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 ANET interfaces
   in a single IPv6 ND message exchange.

   OMNI interfaces use an IPv6 ND option called the "OMNI option"
   formatted as shown in Figure 3:

```
     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    | Prefix Length |R|P|A| Reserved|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                          Sub-Options                          ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                     Figure 3: OMNI Option Format

   In this format:

   o Type is set to TBD.

   o Length is set to the number of 8 octet blocks in the option.

   o Prefix Length is set according to the IPv6 source address type.
      For MN OMNI LLAs, the value is set to the length of the embedded
      MNP. For IPv4-compatible MN OMNI LLAs, the value is set to 96
      plus the length of the embedded IPv4 prefix. For MSE OMNI LLAs,
      the value is set to 128.

   o R (the "Register/Release" bit) is set to '1' to register an MNP or
      set to '0' to release a registration.

   o P (the "Primary" bit) is set to '1' in a MN RS message to request
      an AR to serve as primary, and set to '1' in the AR's RA message
      to accept the primary role. Set to '0' in all other RS/RA
      messages, and ignored in all other ND messages.

   o A (the "Accepts Fragments" bit) is set to '1' in an RS/RA message
      to indicate whether the sender is willing to accept OMNI interface
      link-local fragments (see: Section 5). Nodes that are willing to
      perform link-local reassembly set A to '1' (otherwise '0').

   o Reserved is set to the value '0' on transmission and ignored on
      reception.

   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 options, as described in Section 8.1.

8.1. Sub-Options

   The OMNI option includes zero or more Sub-Options, some of which may
   appear multiple times in the same message. Each consecutive Sub-
   Option is concatenated immediately after its predecessor. All Sub-
   Options except Pad1 (see below) are type-length-value (TLV) encoded
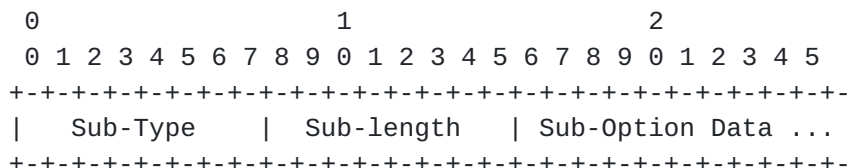   in the following format:

```
    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 4: Sub-Option Format

   o Sub-Type is a 1-byte field that encodes the Sub-Option type. Sub-
      Options defined in this document are:

```
      Option Name            Sub-Type
      Pad1                      0
      PadN                      1
      ifIndex-tuple (Type 1)    2
      ifIndex-tuple (Type 2)    3
      Notification ID           4
```

                              Figure 5

      Sub-Types 253 and 254 are reserved for experimentation, as
      recommended in[RFC3692]].

   o Sub-Length is a 1-byte field that encodes the length of the Sub-
      Body, in bytes

   o Sub-Body is a byte string with format determined by Sub-Type

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

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

8.1.1. Pad1

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

                         Figure 6: Pad1

   o Sub-Type is set to 0.

   o No Sub-Length or Sub-Body follows (i.e., the "Sub-Option" consists
      of a single zero octet).

8.1.2. 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
 |   Sub-Type=1  |Sub-length=N-2 | N-2 padding bytes ...
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

                         Figure 7: PadN

   o Sub-Type is set to 1.

   o Sub-Length is set to N-2 being the number of padding bytes that
      follow.

   o Sub-Body consists of N-2 zero-valued octets.

8.1.3. ifIndex-tuple (Type 1)

```
  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=4+N|    ifIndex    |    ifType     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Provider ID  | Link  |S|I|RSV| 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|  ...
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

                   Figure 8: ifIndex-tuple (Type 1)

o Sub-Type is set to 2.

o Sub-Length is set to 4+N (the number of Sub-Body bytes that
   follow).

o Sub-Body contains an "ifIndex-tuple" (Type 1) encoded as follows
   (note that the first four bytes must be present):

   * ifIndex is set to an 8-bit integer value corresponding to a
      specific underlying ANET interface. OMNI options MAY include
      multiple ifIndex-tuples, and MUST number each with an ifIndex
      value between '1' and '255' that represents a MN-specific 8-bit
      mapping for the actual ifIndex value assigned to the ANET
      interface by network management [RFC2863] (the ifIndex value
      '0' is reserved for use by the MS). Multiple ifIndex-tuples
      with the same ifIndex value MAY appear in the same OMNI option.

   * ifType is set to an 8-bit integer value corresponding to the
      underlying ANET interface identified by ifIndex. 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 ifIndex.

   * 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").

   * S is set to '1' if this ifIndex-tuple corresponds to the
      underlying ANET interface that is the source of the ND message.
      Set to '0' otherwise.

   * I is set to '0' ("Simplex") if the index for each singleton
      Bitmap byte in the Sub-Body is inferred from its sequential
      position (i.e., 0, 1, 2, ...), or set to '1' ("Indexed") if
      each Bitmap is preceded by an Index byte. Figure 8 shows the
      simplex case for I set to '0'. For I set to '1', each Bitmap
      is instead preceded by an Index that encodes a value "i" = (0 -
      255) as the index for its companion Bitmap as follows:

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

                          Figure 9

   * RSV is set to the value 0 on transmission and ignored on
      reception.

   * The remainder of the Sub-Body contains N = (0 - 251) bytes of
      traffic classifier preferences consisting of a first (indexed)
      Bitmap (i.e., "Bitmap(i)") followed by 0-8 1-byte 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 showin in
      Figure 8). The next Bitmap(i) is then consulted with its bits
      indicating which P[*] blocks follow, etc. out to the end of the
      Sub-Option. The first 16 P[*] blocks correspond to the 64
      Differentiated Service Code Point (DSCP) values P[00] - P[63]
      [RFC2474]. If additional P[*] blocks follow, their values
      correspond to "pseudo-DSCP" traffic classifier values P[64],
      P[65], P[66], etc. See Appendix A for further discussion and
      examples.

   * Each 2-bit P[*] field is set to the value '0' ("disabled"), '1'
      ("low"), '2' ("medium") or '3' ("high") to indicate a QoS
      preference level for ANET interface selection purposes. Not
      all P[*] values need to be included in all OMNI option
      instances of a given ifIndex-tuple. Any P[*] values
      represented in an earlier OMNI option but ommitted in the
      current OMNI option remain unchanged. Any P[*] values not yet
      represented in any OMNI option default to "medium".
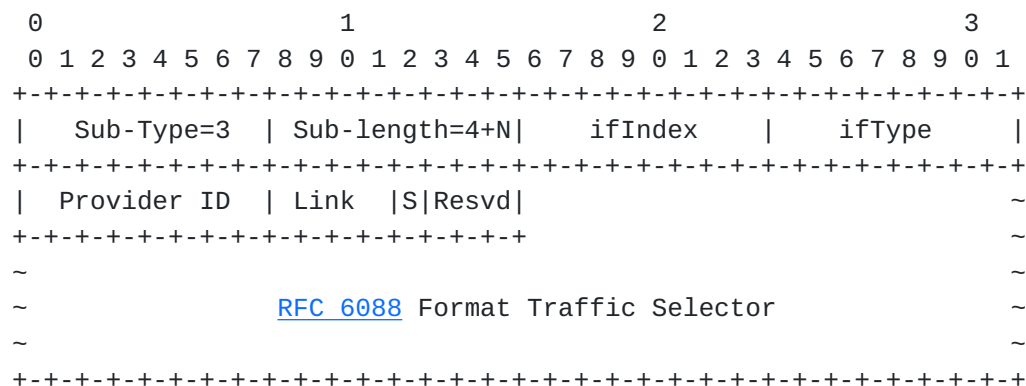
8.1.4. ifIndex-tuple (Type 2)

```
    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=4+N|    ifIndex    |    ifType     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Provider ID  | Link  |S|Resvd|                               ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               ~
   ~                                                               ~
   ~                 RFC 6088 Format Traffic Selector              ~
   ~                                                               ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                     Figure 10: ifIndex-tuple (Type 2)

   o Sub-Type is set to 3.

   o Sub-Length is set to 4+N (the number of Sub-Body bytes that
      follow).

   o Sub-Body contains an "ifIndex-tuple" (Type 2) encoded as follows
      (note that the first four bytes must be present):

      * ifIndex, ifType, Provider ID, Link and S are set exactly as for
         Type 1 ifIndex-tuples as specified in Section 8.1.3.

      * the remainder of the Sub-Option body encodes a variable-length
         traffic selector formatted per [RFC6088].

8.1.5. Notification ID

```
    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=4  | Sub-length=4  | Notification ID (bits 0 - 15) |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Notification ID (bits 16 - 32)|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
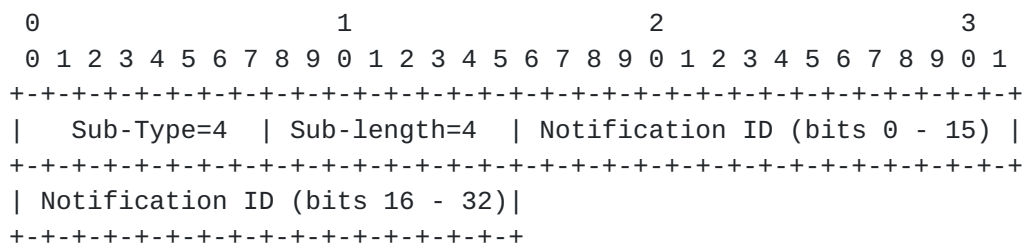
                      Figure 11: Notification ID

   o Sub-Type is set to 4.

   o Sub-Length is set to 4.

   o Notification ID contains the least-significant 32 bits of an MSE
      OMNI LLA to notify (e.g., for the LLA fe80::face:cafe the field
      contains 0xfacecafe). Valid only in MN RS messages, and ignored
      in all other ND messages. OMNI options contain zero or more
      Notification IDs.

9. Address Mapping - Multicast

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

10. Address Mapping for IPv6 Neighbor Discovery Messages

   Per [RFC4861], IPv6 ND messages may be sent to either a multicast or
   unicast link-scoped IPv6 destination address. However, IPv6 ND
   messaging is coordinated between the MN and MS only without invoking
   other nodes on the ANET.

   For this reason, ANET links maintain unicast L2 addresses ("MSADDR")
   for the purpose of supporting MN/MS IPv6 ND messaging. For Ethernet-
   compatible ANETs, this specification reserves one Ethernet unicast
   address TBD2. For non-Ethernet statically-addressed ANETs, MSADDR is
   reserved per the assigned numbers authority for the ANET addressing
   space. For still other ANETs, 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 an 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 MS devices that are configured to
   accept packets destined to MSADDR. Note that multiple MS devices 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].

11. Conceptual Sending Algorithm

   The MN's IPv6 layer selects the outbound OMNI interface according to
   standard IPv6 requirements when forwarding data packets from local or
   EUN applications to external correspondents. The OMNI interface
   maintains default routes and neighbor cache entries for MSEs, and may
   also include additional neighbor cache entries created through other
   means (e.g., Address Resolution, static configuration, etc.).

   After a packet enters the OMNI interface, an outbound ANET interface
   is selected based on multilink parameters such as DSCP, application
   port number, cost, performance, message size, etc. OMNI interface
   multilink selections could also be configured to perform replication
   across multiple ANET interfaces for increased reliability at the
   expense of packet duplication.

   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 ANET interfaces having diverse properties.

11.1. Multiple OMNI Interfaces

   MNs may associate with multiple MS instances concurrently. Each MS
   instance represents a distinct OMNI link distinguished by its
   associated MSPs. 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.

   Depending on local policy and configuration, an MN may choose between
   alternative active OMNI interfaces using a packet's DSCP, routing
   information or static configuration. Interface selection based on
   per-packet source addresses is also enabled when the MSPs for each
   OMNI interface are known (e.g., discovered through Prefix Information
   Options (PIOs) and/or Route Information Options (RIOs)).

   Each OMNI interface can be configured over the same or different sets
   of ANET interfaces. Each ANET distinguishes between the different
   OMNI links based on the MSPs represented in per-packet IPv6
   addresses.

   Multiple distinct OMNI links can therefore be used to support fault
   tolerance, load balancing, reliability, etc. The architectural model
   parallels Layer 2 Virtual Local Area Networks (VLANs), where the MSPs
   serve as (virtual) VLAN tags.

12. Router Discovery and Prefix Registration

   ARs process IPv6 ND messages destined to All-Routers multicast
   (ff02::2), Subnet-Router anycast (fe80::) and unicast IPv6 LLAs
   [RFC4291]. ARs configure the L2 address MSADDR (see: Section 10) and
   act as a proxy for MSE OMNI LLAs.

   MNs interface with the MS by sending RS messages with OMNI options.
   For each ANET interface, the MN sends an RS message with an OMNI
   option, with L2 destination address set to MSADDR and with L3
   destination address set to either a specific MSE OMNI LLA, link-local
   Subnet-Router anycast, or All-Routers multicast. The MN discovers
   MSE OMNI LLAs either through an RA message response to an initial
   anycast/multicast RS or before sending an initial RS message.
   [RFC5214] provides example MSE address discovery methods, including
   information conveyed during data link login, name service lookups,
   static configuration, etc.

   The AR receives the RS messages and coordinates with the
   corresponding MSE in a manner outside the scope of this document.
   The AR returns an RA message with source address set to the MSE OMNI
   LLA, with an OMNI option and with any information for the link that
   would normally be delivered in a solicited RA message. (Note that if

all MSEs share common state, the AR can instead return an RA with
source address set to link-local Subnet-Router anycast.)

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 initial RS
messages to register its MNP and an initial set of underlying ANET
interfaces that are also UP. The MN sends additional RS messages to
refresh lifetimes and to register/deregister underlying ANET
interfaces as they transition to UP or DOWN.

ARs return RA messages with configuration information in response to
a MN's RS messages. The AR sets the RA Cur Hop Limit, M and O flags,
Router Lifetime, Reachable Time and Retrans Timer values as directed
by the MSE, 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 ANET interface.

The AR coordinates with the MSE and sends immediate unicast RA
responses without delay; therefore, the IPv6 ND MAX_RA_DELAY_TIME and
MIN_DELAY_BETWEEN_RAS constants for multicast RAs do not apply. The
AR MAY send periodic and/or event-driven unsolicited RA messages, but
is not required to do so for unicast advertisements [RFC4861].

The MN sends RS messages from within the OMNI interface while using
an UP underlying ANET interface as the outbound interface. Each RS
message is formatted as though it originated from the IPv6 layer, but
the process is coordinated wholly from within the OMNI interface and
is therefore opaque to the IPv6 layer. The MN sends initial RS
messages over an UP underlying interface with its OMNI LLA as the
source and with destination set as discussed above. The RS messages
include an OMNI option per Section 8 with a valid Prefix Length,
(R,P,A) flags, and with ifIndex-tuples appropriate for underlying

ANET interfaces. The AR processes RS message and conveys the OMNI
option information to the MSE.

When the MSE processes the OMNI information, it first validates the
prefix registration information. If the prefix registration was
valid, the MSE injects the MNP into the routing/mapping system then
caches the new Prefix Length, MNP and ifIndex-tuples. If the MN's
OMNI option included one or more Notification IDs, the new MSE also
notifies the former MSE(s). The MSE then directs the AR to return an
RA message to the MN with an OMNI option per Section 8 and with a
non-zero Router Lifetime if the prefix registration was successful;
otherwise, with a zero Router Lifetime.

When the MN receives the RA message, it creates a default route with
L3 next hop address set to the address found in the RA source address
and with L2 address set to MSADDR. The AR will then forward packets
between the MN and the MS.

The MN then manages its underlying ANET interfaces according to their
states as follows:

o When an underlying ANET interface transitions to UP, the MN sends
   an RS over the ANET interface with an OMNI option. The OMNI
   option contains at least one ifIndex-tuple with values specific to
   this ANET interface, and may contain additional ifIndex-tuples
   specific to this and/or other ANET interfaces.

o When an underlying ANET interface transitions to DOWN, the MN
   sends an RS or unsolicited NA message over any UP ANET interface
   with an OMNI option containing an ifIndex-tuple for the DOWN ANET
   interface with Link(i) 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 ANET interfaces).

o When a MN wishes to release from a current MSE, it sends an RS or
   unsolicited NA message over any UP ANET interfaces with an OMNI
   option with R set to 0. The corresponding MSE then withdraws the
   MNP from the routing/mapping system and (for RS responses) directs
   the AR to return an RA message with an OMNI option and with Router
   Lifetime set to 0.

o When a MN wishes to transition to a new MSE, it sends an RS or
   unsolicited NA message over any UP ANET interfaces with an OMNI
   option with R set to 1, with the new MSE OMNI LLA set in the
   destination address, and (optionally) with a Notification ID
   included for the former MSE.

   o When all of a MNs underlying interfaces have transitioned to DOWN
      (or if the prefix registration lifetime expires) the MSE withdraws
      the MNP the same as if it had received a message with an OMNI
      option with R set to 0.

   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 multiple
   UP ANET interfaces, the MN declares this MSE unreachable and tries a
   different MSE.

   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.

   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). For this
   reason, the MN should select a primary AR, which is responsible for
   keeping the MS prefix registration alive on the MN's behalf. If the
   MN does not select a primary, then it must perform more frequent RS/
   RA exchanges on its own behalf to refresh the MS prefix registration
   lifetime.

13. AR and MSE Resilience

   ANETs 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.

[14](). 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 aeronautical radio links) and can therefore be tuned for rapid
response.

ARs perform proactive NUD for MSEs for which there are currently
active ANET MNs. If an MSE fails, ARs can quickly inform MNs of the
outage by sending multicast RA messages on the ANET interface. The
AR sends RA messages to the MN via the ANET interface with source
address set to the MSEs OMNI LLA, destination address set to All-
Nodes multicast (ff02::1) [RFC4291], and Router Lifetime set to 0.

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

[15](). IANA Considerations

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

The OMNI option also defines an 8-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         ifIndex-tuple (Type 1)    [RFCXXXX]
   3         ifIndex-tuple (Type 2)    [RFCXXXX]
   4         Notification ID           [RFCXXXX]
   5-252     Unassigned
   253-254   Experimental              [RFCXXXX]
   255       Reserved                  [RFCXXXX]
```

                  Figure 12: OMNI Option Sub-Type 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".

16. Security Considerations

   Security considerations for IPv6 [RFC8200] and IPv6 Neighbor
   Discovery [RFC4861] apply. OMNI interface IPv6 ND messages SHOULD
   include Nonce and Timestamp options [RFC3971] when synchronized
   transaction confirmation is needed.

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

17. 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:
   Michael Matyas, Madhu Niraula, Greg Saccone, Stephane Tamalet, Eric
   Vyncke. Pavel Drasil, Zdenek Jaron and Michal Skorepa are recognized
   for their many helpful ideas and suggestions.

18. References

18.1. Normative References

[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>.

[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>.

[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>.

   [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>.

18.2. Informative References

   [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>.

   [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>.

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

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

   [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>.

   [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>.

   [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>.

   [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>.

   [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>.

   [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>.

Appendix A. Type 1 ifIndex-tuple Traffic Classifier Preference Encoding

   Adaptation of the OMNI option Type 1 ifIndex-tuple's traffic
   classifier bitmap 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 ifIndex-tuple
   the same format must be used for the entire Sub-Option. Different
   Sub-Options may use different formats.

   The following examples 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=2  | Sub-length=4+N|    ifIndex    |    ifType     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Provider ID  |   Link   |S|0|RSV| 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 13: 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=2  | Sub-length=4+N|    ifIndex    |    ifType     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Provider ID  |   Link   |S|0|RSV| 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 14: 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=2  | Sub-length=4+N|    ifIndex    |    ifType     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Provider ID  |  Link  |S|1|RSV|  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 15: Example 3: Indexed Encoding

Appendix B. Prefix Length Considerations

   The 64-bit boundary in IPv6 addresses [RFC7421] determines the MN
   OMNI LLA format for encoding the most-significant 64 MNP bits into
   the least-significant 64 bits of the prefix fe80::/64 as discussed in
   Section 7.

   [RFC4291] defines the link-local address format as the most
   significant 10 bits of the prefix fe80::/10, followed by 54 unused
   bits, followed by the least-significant 64 bits of the address. If
   the 64-bit boundary is relaxed through future standards activity,
   then the 54 unused bits can be employed for extended coding of MNPs
   of length /65 up to /118.

   The extended coding format would continue to encode MNP bits 0-63 in
   bits 64-127 of the OMNI LLA, while including MNP bits 64-117 in bits
   10-63. For example, the OMNI LLA corresponding to the MNP
   2001:db8:1111:2222:3333:4444:5555::/112 would be
   fe8c:ccd1:1115:5540:2001:db8:1111:2222, and would still be a valid
   IPv6 LLA per [RFC4291].

Appendix C. 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 D. Change Log

<< RFC Editor - remove prior to publication >>

Differences from draft-templin-atn-aero-interface-20 to draft-
templin-atn-aero-interface-21:

o OMNI option format

o MTU

Differences from draft-templin-atn-aero-interface-19 to draft-
templin-atn-aero-interface-20:

o MTU

Differences from draft-templin-atn-aero-interface-18 to draft-
templin-atn-aero-interface-19:

o MTU

Differences from draft-templin-atn-aero-interface-17 to draft-
templin-atn-aero-interface-18:

o MTU and RA configuration information updated.

   Differences from [draft-templin-atn-aero-interface-16](#) to [draft-templin-atn-aero-interface-17](#):

   o New "Primary" flag in OMNI option.

   Differences from [draft-templin-atn-aero-interface-15](#) to [draft-templin-atn-aero-interface-16](#):

   o New note on MSE OMNI LLA uniqueness assurance.

   o General cleanup.

   Differences from [draft-templin-atn-aero-interface-14](#) to [draft-templin-atn-aero-interface-15](#):

   o General cleanup.

   Differences from [draft-templin-atn-aero-interface-13](#) to [draft-templin-atn-aero-interface-14](#):

   o General cleanup.

   Differences from [draft-templin-atn-aero-interface-12](#) to [draft-templin-atn-aero-interface-13](#):

   o Minor re-work on "Notify-MSE" (changed to Notification ID).

   Differences from [draft-templin-atn-aero-interface-11](#) to [draft-templin-atn-aero-interface-12](#):

   o Removed "Request/Response" OMNI option formats. Now, there is
      only one OMNI option format that applies to all ND messages.

   o Added new OMNI option field and supporting text for "Notify-MSE".

   Differences from [draft-templin-atn-aero-interface-10](#) to [draft-templin-atn-aero-interface-11](#):

   o Changed name from "aero" to "OMNI"

   o Resolved AD review comments from Eric Vyncke (posted to atn list)

   Differences from [draft-templin-atn-aero-interface-09](#) to [draft-templin-atn-aero-interface-10](#):

   o Renamed ARO option to AERO option

   o Re-worked [Section 13](#) text to discuss proactive NUD.

Differences from draft-templin-atn-aero-interface-08 to draft-templin-atn-aero-interface-09:

o Version and reference update

Differences from draft-templin-atn-aero-interface-07 to draft-templin-atn-aero-interface-08:

o Removed "Classic" and "MS-enabled" link model discussion

o Added new figure for MN/AR/MSE model.

o New Section on "Detecting and responding to MSE failure".

Differences from draft-templin-atn-aero-interface-06 to draft-templin-atn-aero-interface-07:

o Removed "nonce" field from AR option format. Applications that
   require a nonce can include a standard nonce option if they want
   to.

o Various editorial cleanups.

Differences from draft-templin-atn-aero-interface-05 to draft-templin-atn-aero-interface-06:

o New Appendix C on "VDL Mode 2 Considerations"

o New Appendix D on "RS/RA Messaging as a Single Standard API"

o Various significant updates in Section 5, 10 and 12.

Differences from draft-templin-atn-aero-interface-04 to draft-templin-atn-aero-interface-05:

o Introduced RFC6543 precedent for focusing IPv6 ND messaging to a
   reserved unicast link-layer address

o Introduced new IPv6 ND option for Aero Registration

o Specification of MN-to-MSE message exchanges via the ANET access
   router as a proxy

o IANA Considerations updated to include registration requests and
   set interim RFC4727 option type value.

Differences from draft-templin-atn-aero-interface-03 to draft-templin-atn-aero-interface-04:

o Removed MNP from aero option format - we already have RIOs and
   PIOs, and so do not need another option type to include a Prefix.

o Clarified that the RA message response must include an aero option
   to indicate to the MN that the ANET provides a MS.

o MTU interactions with link adaptation clarified.

Differences from draft-templin-atn-aero-interface-02 to draft-
templin-atn-aero-interface-03:

o Sections re-arranged to match RFC4861 structure.

o Multiple aero interfaces

o Conceptual sending algorithm

Differences from draft-templin-atn-aero-interface-01 to draft-
templin-atn-aero-interface-02:

o Removed discussion of encapsulation (out of scope)

o Simplified MTU section

o Changed to use a new IPv6 ND option (the "aero option") instead of
   S/TLLAO

o Explained the nature of the interaction between the mobility
   management service and the air interface

Differences from draft-templin-atn-aero-interface-00 to draft-
templin-atn-aero-interface-01:

o Updates based on list review comments on IETF 'atn' list from
   4/29/2019 through 5/7/2019 (issue tracker established)

o added list of opportunities afforded by the single virtual link
   model

o added discussion of encapsulation considerations to Section 6

o noted that DupAddrDetectTransmits is set to 0

o removed discussion of IPv6 ND options for prefix assertions. The
   aero address already includes the MNP, and there are many good
   reasons for it to continue to do so. Therefore, also including
   the MNP in an IPv6 ND option would be redundant.

   o Significant re-work of "Router Discovery" section.

   o New Appendix B on Prefix Length considerations

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

   o 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