Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces
draft-templin-6man-omni-interface-46

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.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Mobile Nodes (MNs) (e.g., aircraft of various configurations,
   terrestrial vehicles, seagoing vessels, enterprise wireless 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 links.

   The MN configures a virtual interface (termed the "Overlay Multilink
   Network (OMNI) interface") as a thin layer over the underlying Access
   Network (ANET) 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 12).

   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) 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. The IP
layer selects an OMNI interface based on SBM routing considerations,
then the selected interface 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.

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.

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 prefix (e.g., 2001:db8::/32, 192.0.2.0/24, etc.)
      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 IP prefix taken 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 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 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 ANET/INET interfaces.

   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 Link-Local Address (LLA)
      a link local IPv6 address per [RFC4291] constructed as specified
      in Section 7.

   OMNI Segment-Local Address (SLA)
      an IPv6 address from the prefix fec0::/10 (see: Section 2.5.7 of
      [RFC4291]) constructed as specified in Section 8. OMNI SLAs are
      statelessly derived from OMNI LLAs, and vice-versa.

   OMNI Option
      an IPv6 Neighbor Discovery option providing multilink parameters
      for the OMNI interface as specified in Section 9.

   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", "IP layer", etc.

underlying interface
   an ANET/INET 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.

Mobility Service Identification (MSID)
   Each MSE and AR is assigned a unique 32-bit Identification (MSID)
   as specified in Section 7.

Safety-Based Multilink (SBM)
   A means for ensuring fault tolerance through redundancy by
   connecting multiple independent 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
   trasnmission and reception within a single OMNI interface.

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 amplificaiton. 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 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
[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-     +---->|         (OMNI LLA)         |
        Physical        |     +----------------------------+
        Interface       +---->|  L2  |  L2  |       |  L2  |
        Binding              |(IF#1)|(IF#2)|  .....  |(IF#n)|
                             +------+------+       +------+
                             |  L1  |  L1  |       |  L1  |
                             |      |      |       |      |
                             +------+------+       +------+
```
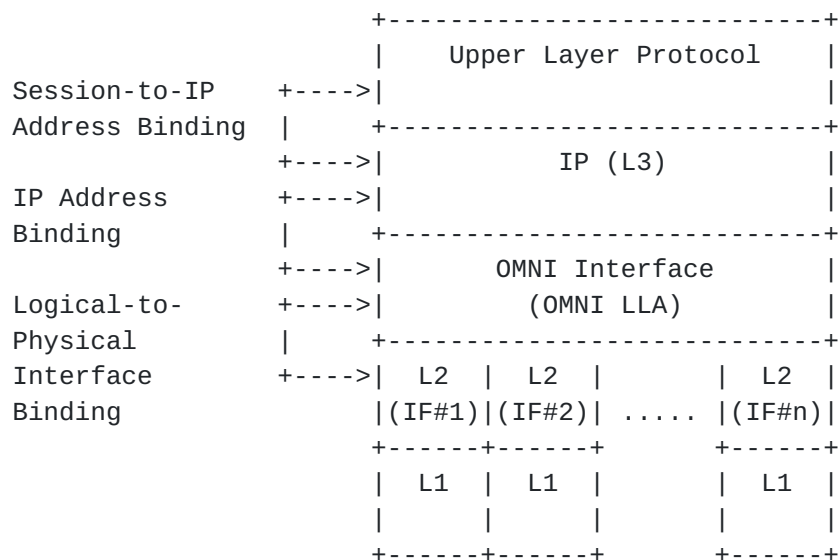
Figure 1: OMNI Interface Architectural Layering Model

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 an INET/ANET 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 ANET/INET 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:

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

   o ANET 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 OMNI LLA to all ANET interfaces.)

   o 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.

   o coordinating underlying 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 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.

o 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.

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.

o 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 connecting to the
MS via multiple independent ANETs. When an underlying interface
becomes active, the MN's OMNI interface sends native (i.e.,
unencapsulated) IPv6 ND messages via the underlying 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 ---->      .
      ..........................................................
```
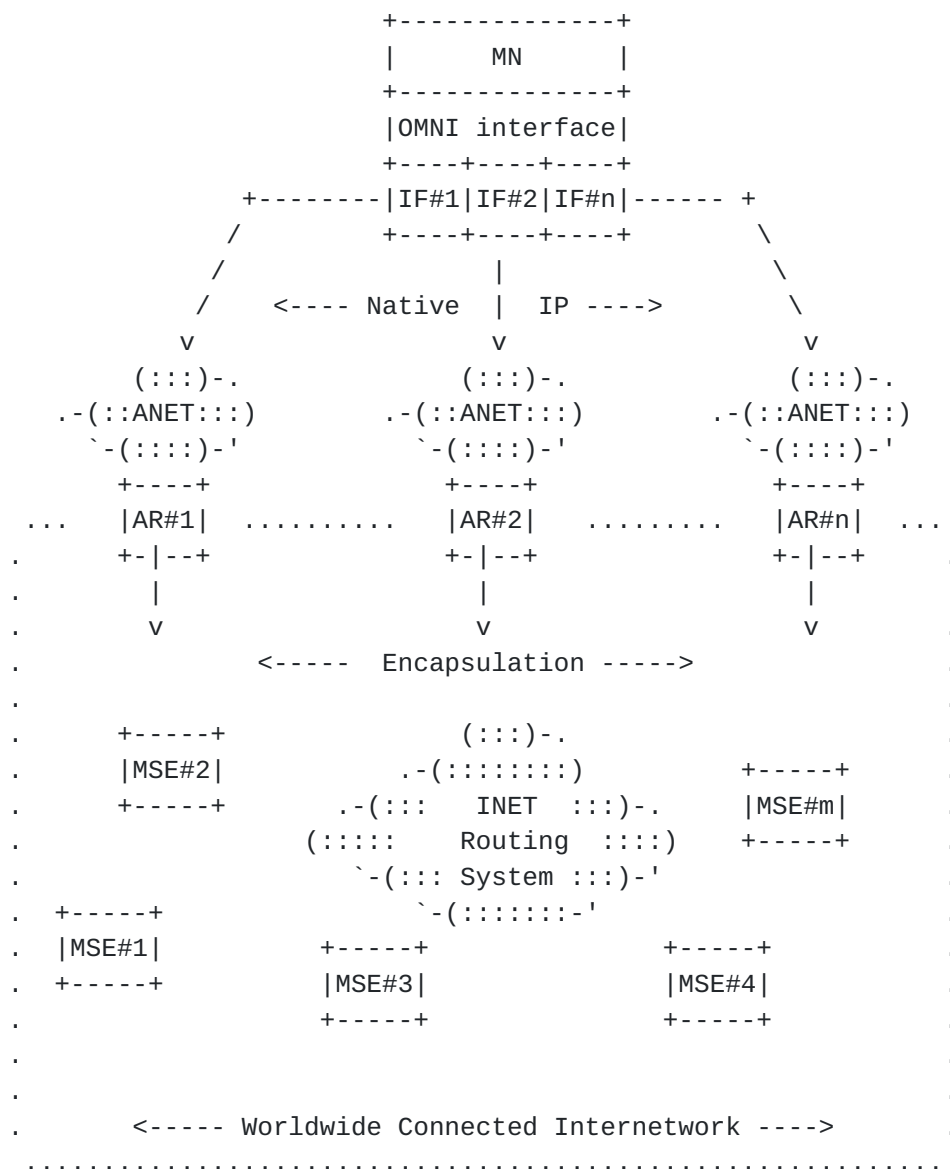
               Figure 2: MN/MS Coordination via Multiple ANETs

   After the initial IPv6 ND message exchange, the MN can send and
   receive unencapsulated IP 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.

   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.

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 Adpatation 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], meaning that
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 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 consderation of the issues discussed in
Section 5.1.)

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 inserting a mid-layer IPv6 header
between the inner IP packet/fragment and any outer IP encapsulation
headers per [RFC2473] (but without decrementing the inner IP Hop
Limit/TTL since the insertion occurs at a layer below IP forwarding).
The OAL then calculates the 32-bit CRC over the entire mid-layer
packet and writes the value in a trailing 4-octet field at the end of
the packet. Next, the OAL fragments this mid-layer IPv6 packet,
forwards the fragments (using outer IP encapsulation if necessary),
and returns an internally-generated PTB soft error message (subject
to rate limitiing) 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 OMNI Segment-Local Address (SLA) (see:
   Section 8) and destination set to the OMNI SLA of the ANET peer.
   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 largest fragment size
   (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 fragements in any ANET
   headers and sends them to the ANET peer, which reassembles before
   forwarding toward the final destination.

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 OMNI SLA, destination set to the SLA of the next hop
   OMNI node toward the final destination and (if necessary) with a
   Segment Routing Header with the remaining Segment IDs on the path
   to the final destination. 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 largest fragment size (including both the OMNI and outer-layer
   INET encapsulations) is the minimum INET path MTU, and the
   smallest fragment size is no smaller than 256 bytes (i.e.,
   slightly less than half the minimum IPv4 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.

The OAL unconditionally drops all OAL fragments received from an INET
peer that are smaller than 256 bytes (note that no minimum fragment
size is specified for ANET peers since the underlying ANET is secured
against tiny fragment attacks). In order to set the correct context
for reassembly, the OAL of the OMNI interface that inserts the IPv6
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 (i.e., whether the MN was the source or target of the

original packet). 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: 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. Always
assuming the IPv4 minimum path MTU even for IPv6 underlying
interfaces may produce more fragments and additional header overhead,
but will always interoperate and never run the risk of presenting an
IPv4 node with a packet that exceeds its MRU.

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 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: Inclusion of the 32-bit CRC prior to fragmentation assumes that the receiving OAL will discard any packets with incorrect CRC values following reassembly. The 32-bit CRC is sufficient to detect reassembly misassociations for packet sizes up to the OMNI interface MTU 9180 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 properties such as header/message compression can also be harnessed in a similar fashion.

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 INET 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.

6. Frame Format

    The OMNI interface transmits IPv6 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 interfaces construct IPv6 Link-Local Addresses (i.e., "OMNI
    LLAs") as follows:

    o IPv6 MN OMNI LLAs encode the most-significant 112 bits of a MNP
       within the least-significant 112 bits of the IPv6 link-local
       prefix fe80::/16. The Prefix Length is determined by adding 16 to
       the length of the embedded MNP. For example, for the MNP
       2001:db8:1000:2000::/56 the corresponding MN OMNI LLA is
       fe80:2001:db8:1000:2000::/72. This specification updates the IPv6
       link-local address format specified in Section 2.5.6 of [RFC4291]
       by defining a use for bits 11 - 63.

    o IPv4-compatible MN OMNI LLAs are constructed as fe80::ffff:[IPv4],
       i.e., the most significant 16 bits of the prefix fe80::/16,
       followed by 64 '0' bits, followed by 16 '1' bits, followed by a
       32bit IPv4 address/prefix. The Prefix Length is determined by
       adding 96 to the length of the embedded IPv4 address/prefix. 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 MS OMNI LLAs are assigned to ARs and MSEs from the range
   fe80::/96, 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::feff:ffff. The MS OMNI LLA Prefix Length is
   determined by adding 96 to the MSID prefix length. For example,
   if the MSID '0x10002000' prefix length is 16 then the MS OMNI LLA
   Prefix Length is set to 112 and the LLA is written as
   fe80::1000:2000/112. Finally, the MSID 0x00000000 is the
   "Anycast" MSID and corresponds to the link-local Subnet-Router
   anycast address (fe80::) [RFC4291]; the MSID range 0xff000000
   through 0xffffffff is reserved for future use.

o The OMNI LLA range fe80::/32 is used as the service prefix for the
   address format specified in Section 4 of [RFC4380] (see Section 17
   for further discussion).

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 MS OMNI LLAs are guaranteed unique
through administrative assignment, OMNI interfaces set the
autoconfiguration variable DupAddrDetectTransmits to 0 [RFC4862].

8. Segment-Local Addresses (SLAs)

OMNI links use IPv6 Segment-Local Addresses (i.e., "OMNI SLAs") as
the source and destination addresses in OAL IPv6 encapsulation
headers. This document uses the prefix fec0::/10 for mapping OMNI
LLAs to routable OMNI SLAs, and therefore updates Section 2.5.7 of
[RFC4291] by re-purposing the (deprecated) IPv6 "site-local" address
block.

Each OMNI link instance is identified by bits 10-15 of the OMNI
service prefix fec0::/10. For example, OMNI SLAs associated with
instance 0 are configured from the prefix fec0::/16, instance 1 from
fec1::/16, etc., up to instance 63 from feff::/16. OMNI SLAs and
their associated prefix lengths are configured in one-to-one
correspondence with OMNI LLAs through stateless prefix translation.
For example, for OMNI link instance fec0::/16:

o the OMNI SLA corresponding to fe80:2001:db8:1:2::/80 is simply
   fec0:2001:db8:1:2::/80

o the OMNI SLA corresponding to fe80::ffff:192.0.2.0/120 is simply
   fec0::ffff:192.0.2.0/120

o   the OMNI SLA corresponding to fe80::1000/112 is simply
    fec0::1000/112

o   the OMNI SLA corresponding to fe80::/128 is simply fec0::/128.

Each OMNI interface assigns the Anycast OMNI SLA specific to the OMNI
link instance, e.g., the OMNI interface connected to instance 3
assigns the Anycast OMNI SLA fec3:. Routers that configure OMNI
interfaces advertise the OMNI service prefix (e.g., fec3::/16) into
the local routing system so that applications can direct traffic
according to SBM requirements.

The OMNI SLA presents an IPv6 address format that is routable within
the OMNI link routing system and can be used to convey link-scoped
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 over
ANET links when possible to conserve bandwidth (see: Section 11).

The 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: Re-purposing the (deprectated) IPv6 "site-local" address block
as OMNI SLAs is a natural fit since both LLA and SLA prefix lengths
are ::/10, the prefixes fe80:: and fec0:: differ only in a single bit
setting, and LLAs and SLAs can be unambiguously allocated in one-to-
one correspondence with one another. This would also make good use
of an otherwise-wasted address range that has been "parked" since the
2004 deprecation. However, re-purposing the block will require an
IETF standards action acknowledging this document as obsoleting
[RFC3879] and updating [RFC4291].

9. 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. 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 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 | S/T-ifIndex   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                         Sub-Options                           ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
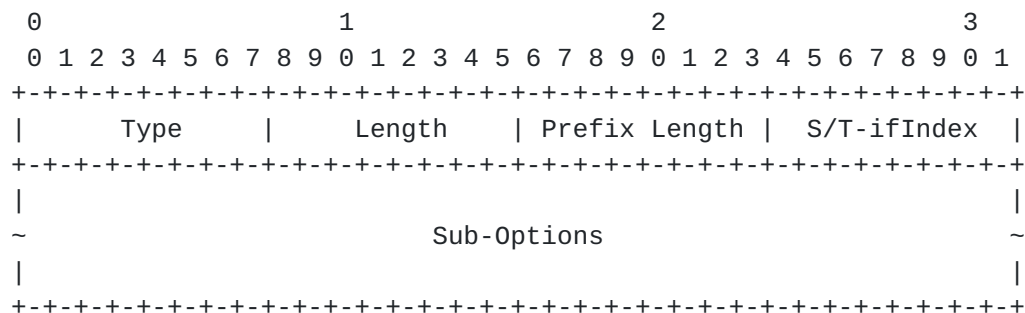
                     Figure 3: OMNI Option Format

In this format:

o Type is set to TBD. If multiple OMNI option instances appear in
   the same IPv6 ND message, the first instance is processed and all
   other instances are ignored.

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

o Prefix Length is determines the length of prefix to be applied to
   an OMNI MN LLA/SLA. For IPv6 ND messages sent from a MN to the
   MS, Prefix Length is the length that the MN is requesting or
   asserting to the MS. For IPv6 ND messages sent from the MS to the
   MN, Prefix Length indicates the length that the MS is granting to
   the MN. For IPv6 ND messages sent between MS endpoints, Prefix
   Length indicates the length associated with the target MN that is
   subject of the ND message.

o S/T-ifIndex corresponds to the ifIndex value for source or target
   underlying interface used to convey this IPv6 ND message. OMNI
   interfaces MUST number each distinct underlying interface with an
   ifIndex value between '1' and '255' that represents a MN-specific
   8-bit mapping for the actual ifIndex value assigned by network

management [RFC2863] (the ifIndex value '0' is reserved for use by
the MS). For RS and NS messages,S/T-ifIndex corresponds to the
source underlying interface the message originated from. For RA
and NA messages, S/T-ifIndex corresponds to the target underlying
interface that the message is destined to.

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 9.1.

9.1. 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 ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```
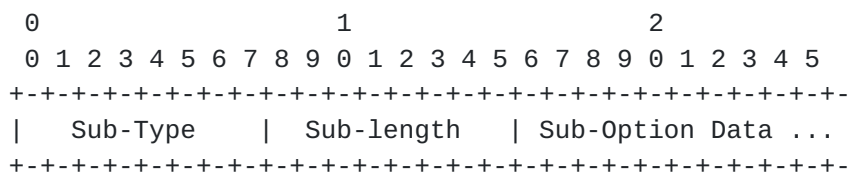
                    Figure 4: Sub-Option Format

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

```
     Option Name                 Sub-Type
     Pad1                            0
     PadN                            1
     Interface Attributes            2
     Traffic Selector                3
     MS-Register                     4
     MS-Release                      5
     Network Access Identifier       6
     Geo Coordinates                 7
     DHCP Unique Identifier (DUID)   8
```

                              Figure 5

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

o Sub-Length is a 1-octet field that encodes the length of the Sub-
   Option Data (i.e., ranging from 0 to 255 octets).

o Sub-Option Data is a block of data 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 is reached.

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

9.1.1. Pad1

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

                     Figure 6: Pad1

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

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

9.1.2. 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  | N padding octets ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

                     Figure 7: PadN

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

o Sub-Length is set to N (from 0 to 255) being the number of padding
   octets that follow.

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

9.1.3. Interface Attributes

```
 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  |    ifIndex    |    ifType     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Provider ID  | Link  |R| APS  |    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 8: Interface Attributes

o Sub-Type is set to 2. If multiple instances with different
   ifIndex values appear in the same OMNI option all are processed;
   if multiple instances with the same ifIndex value appear, the
   first is processed and all others are ignored.

o Sub-Length is set to N (from 4 to 255) that encodes the number of
   Sub-Option Data octets that follow.

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

   * ifIndex is set to an 8-bit integer value corresponding to a
      specific underlying interface the same as specified above for
      the OMNI option header S/T-ifIndex. An OMNI option may include
      multiple Interface Attributes Sub-Options, with each distinct
      ifIndex value pertaining to a different underlying interface.
      The OMNI option will often include an Interface Attributes Sub-
      Option with the same ifIndex value that appears in the S/
      T-ifIndex. 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 absent) the presence of a Network
      Address Translator (NAT) is indicated.

   * ifType is set to an 8-bit integer value corresponding to the
      underlying 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").

*   R is reserved for future use.

*   APS - a 3-bit "Address/Preferences/Simplex" 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 APS 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 APS code.)

    + When a preferences block is present and the least
       significant bit (i.e., "Simplex") is set to 1, the block is
       encoded in "Simplex" form as shown in Figure 8; else it is
       encoded in "Indexed" form as discussed below.

*   When APS 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 SLA formed from concatenating fe*::/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 addresss 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 a Network
   Address Translator (NAT).

- 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 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; 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 algoritherm uses FMT/L2ADDR to
   determine the encapsulation address for forwarding when SRT/
   LHS is located in the local OMNI link segment.

* When APS 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 8.
   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 9

* 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 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.

* 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.

9.1.4. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Sub-Type=3  |  Sub-length=N |    ifIndex    |             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+             ~
~                                                             ~
~                    RFC 6088 Format Traffic Selector         ~
~                                                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

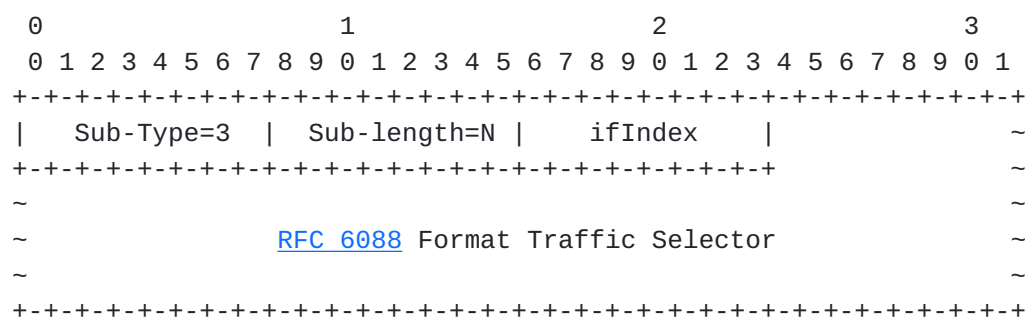   Figure 10: Traffic Selector

o Sub-Type is set to 3. If multiple instances appear in the same
   OMNI option all are processed, i.e., even if the same ifIndex
   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 ifIndex encoded exactly as
      specified in Section 9.1.3, followed by an N-1 octet traffic
      selector formatted per [RFC6088] beginning with the "TS Format"
      field. The largest traffic selector for a given ifIndex is
      therefore 254 octets.

9.1.5. 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Sub-Type=4  | 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)   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

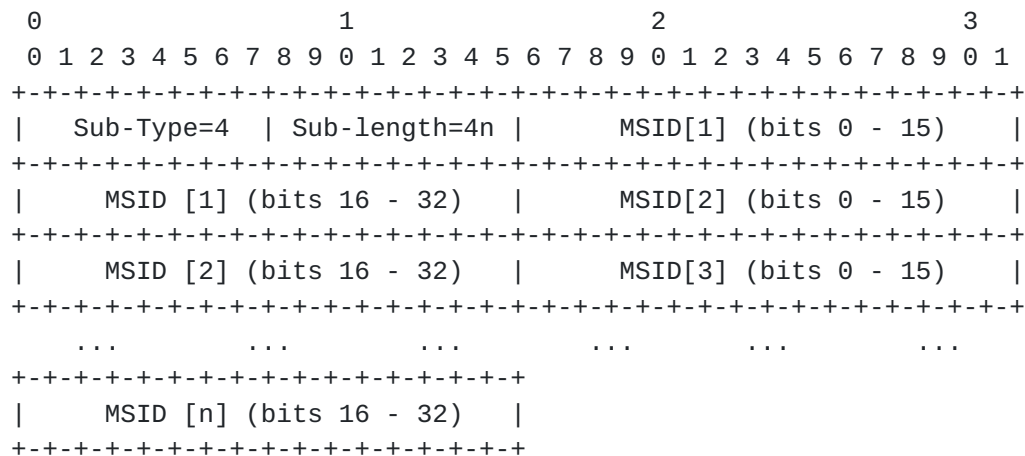   Figure 11: MS-Register Sub-option

   o Sub-Type is set to 4. If multiple instances appear in the same
      OMNI option 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.

   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.
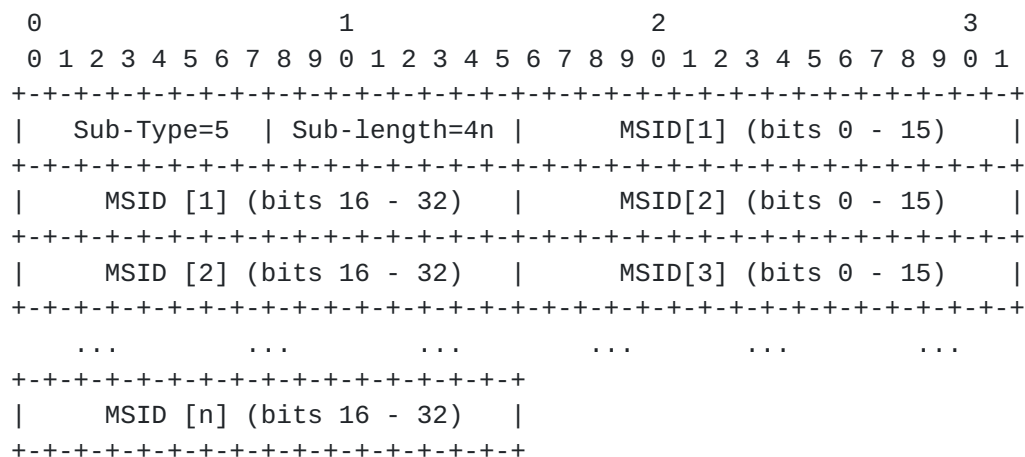
9.1.6. MS-Release

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Sub-Type=5  | 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 12: MS-Release Sub-option

o Sub-Type is set to 5. If multiple instances appear in the same
   IPv6 OMNI option 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.

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.

9.1.7. Network Access Identifier (NAI)

```
     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=6  | Sub-length=N  |Network Access Identifier (NAI)
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```
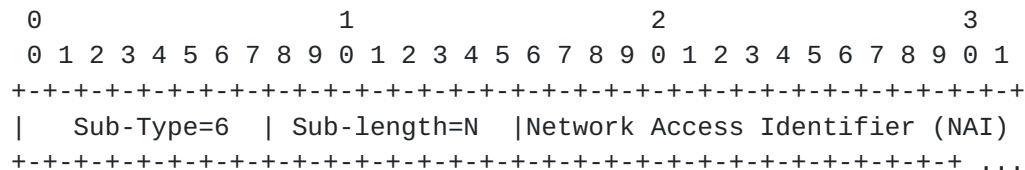
          Figure 13: Network Access Identifier (NAI) Sub-option

o Sub-Type is set to 6. If multiple instances appear in the same
   OMNI option the first is processed and all others are ignored.

o Sub-Length is set to N.

o A Network Access Identifier (NAI) up to 255 octets in length is
   coded per [RFC7542].

9.1.8. 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Sub-Type=7  | Sub-length=N  |      Geo Coordinates
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

                Figure 14: Geo Coordinates Sub-option

   o Sub-Type is set to 7. If multiple instances appear in the same
      OMNI option the first is processed and all others are ignored.

   o Sub-Length is set to N.

   o A set of Geo Coordinates up to 255 octets in length (format TBD).
      Includes Latitude/Longitude at a minimum; may also include
      additional attributes such as altitude, heading, speed, etc.).

9.1.9. DHCP Unique Identifier (DUID)

```
    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=8  | Sub-length=N  |            DUID-Type          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                                                               .
   .            type-specific DUID body (variable length)         .
   .                                                               .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

            Figure 15: DHCP Unique Identifier (DUID) Sub-option

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

   o Sub-Length is set to N (i.e., the length of the option beginning
      with the DUID-Type and continuing to the end of the type-specific
      body).

   o DUID-Type is a two-octet field coded in network byte order that
      determines the format and contents of the type-specific body
      according to Section 11 of [RFC8415]. DUID-Type 4 in particular
      corresponds to the Universally Unique Identifier (UUID) [RFC6355]
      which will occur in common operational practice.

o A type-specific DUID body up to 253 octets in length follows,
   formatted according to DUID-type. For example, for type 4 the
   body consists of a 128-bit UUID selected according to [RFC6355].

10. 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 ANET L2 elements use MLD snooping
[RFC4541].

11. 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.

After a packet enters the OMNI interface, an outbound underlying
interface is selected based on PBM traffic 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.

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. The OAL
also performs encapsulation when the nearest AR is located multiple
hops away as discussed in Section 12.1.

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.

11.1. Multiple OMNI Interfaces

MNs may connect to multiple independent OMNI links concurrently in
support of SBM. Each OMNI interface is distinguished by its Anycast
OMNI SLA (e.g., fec0::, fec1::, fec2::). 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 OMNI SLA 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 OMNI SLA 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 OMNI SLA 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 OMNI SLA.

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).

12. 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 ANET will process the
message and respond. The manner in which the ANET ensures AR
coordination is link-specific and outside the scope of this document
(however, considerations for ANETs that do not provide ARs that
recognize the OMNI option are discussed in Section 17).

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 suboption 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
suboption. The AR selects only a single wildcard MSE (i.e., even if
the RS MS-Register suboption 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 OMNI LLA as the source and with
destination set to All-Routers multicast (ff02::2) [RFC4291]. The RS
messages include an OMNI option per Section 9 with valid prefix
registration information, Interface Attributes appropriate for
underlying interfaces, MS-Register/Release sub-options containing
MSID values, and with any other necessary OMNI sub-options. The S/
T-ifIndex 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 Prefix Length, 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 9.

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 MN OMNI LLA (i.e., unicast), with source address set to
its own OMNI LLA, and with an OMNI option with S/T-ifIndex set to the
value included in the RS. The OMNI option also includes valid prefix
registration information, Interface Attributes, MS-Register/Release
and any other necessary OMNI sub-options. 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 ANET interface.

The AR MAY also send periodic and/or event-driven unsolicited RA
messages per [RFC4861]. In that case, the S/T-ifIndex field in the
OMNI header 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 ANET 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 OMNI 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 ANETs, it records the additional L2 AR addresses in each
MSID neighbor cache entry (i.e., as multilink neighbors). 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 a 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.

12.1. Router Discovery in IP Multihop and IPv4-Only Access Networks

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

A MN located potentially multiple ANET hops away from the nearst AR prepares an RS message with source address set to its OMNI LLA and with destination set to link-scoped All-Routers multicast the same as discussed above. For IPv6-enabled ANETs, the MN then encapsulates the message in an IPv6 header with source address set to the SLA corresponding to the LLA source address and with destination set to site-scoped All-Routers multicast (ff05::2)[RFC4291]. For IPv4-only ANETs, the MN instead encapsulates the RS message in an IPv4 header with source address set to the node's own IPv4 address. For multicast-capable IPv4-only ANETs, the MN then sets the destination address to the site-scoped IPv4 multicast address corresponding to link-scoped IPv6 All-Routers multicast [RFC2529]; for unicast-only IPv4-only ANETs, the MN instead sets the destination address to the unicast IPv4 adddress of an AR [RFC5214]. The MN then sends the encapsulated RS message via the ANET interface, where it will be forwarded by zero or more intermediate ANET hops.

When an intermediate ANET hop that particpates 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 ANET 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 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/SLA address and with destination set to the encapsulation source of the RS.

The AR then forwards the message to an ANET node within communications range, which forwards the message according to its

routing tables to an intermediate node. The multihop forwarding
process within the ANET 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
SLAs 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 OMNI LLA addresses 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.

12.2. 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 OMNI LLA containing the MSE 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 folllows:

   * 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 a uNA message to each MSID in the MS-Release list
      with the MS-Register/Release lists omitted and with an OMNI
      header with S/T-ifIndex 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 SLA of the AR as the source and the SLA
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
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
SLA of the Register MSE as the source and the SLA 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
the AR will generate multiple proxy RS and/or uNA messages - one for
each copy of the duplicate entries.

13. Secure Redirection

If the ANET 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 ANET 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
ANET 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 ANET 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.

14. 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.

15. 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 ANET 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 ANET. 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 MNs via the ANET 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 ANET interface that have been using the (now defunct) MSE will receive the RA messages and associate with a new MSE.

16. Transition Considerations

When a MN connects to an ANET 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 MS OMNI LLA as the source, the MN OMNI 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 ANET according to the legacy IPv6 link model and without the OMNI extensions specified in this document.

If the ANET 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 ANET link with an OMNI 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 ANET 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.

17. OMNI Interfaces on the Open Internet

OMNI interfaces configured over IPv6-enabled underlying interfaces on
the open Internet 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 prefxies, 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
the open Internet can apply security services such as VPNs to connect
to an MSE or establish a direct link to an MSE through some other
means. In environments where an explicit VPN or direct link may be
impractical, OMNI interfaces can instead use UDP/IP encapsulation per
[RFC6081][RFC4380]. (SEcure Neighbor Discovery (SEND) and
Cryptographically Generated Addresses (CGA) [RFC3971][RFC3972] or
other protocol-specific security services can can also be used if
additional authentication is necessary.)

After estabishing a VPN or preparing for UDP/IP encapsulation, OMNI
interfaces send control plane messages to interface with the MS. The
control plane messages must be authenticated while data plane
messages are delivered the same as for ordinary best-effort Internet
traffic with basic source address-based data origin verification.
Data plane communications via OMNI interfaces that connect over the
open Internet without an explicit VPN should therefore employ
transport- or higher-layer security to ensure integrity and/or
confidentiality.

When SEND/CGA are used over an open Internet underlying interfaces,
each OMNI node configures a link-local CGA for use as the source
address of IPv6 ND messages. The node then employs OMNI link
encapsualation and sets the IPv6 source address of the OMNI header to
the SLA corresponding to its OMNI LLA. Any Prefix Length values in
the IPv6 ND message OMNI option then apply to the SLA found in the
OMNI header, i.e., and not to the CGA found in the IPv6 ND message
source address.

OMNI interfaces in the open Internet are often located behind Network Address Translators (NATs). The OMNI interface accommodates NAT traversal using UDP/IP encapsulation and the mechanisms discussed in [RFC6081][RFC4380][I-D.templin-intarea-6706bis].

18. 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.

Prefix delegation services such as those discussed in [I-D.templin-6man-dhcpv6-ndopt] and [I-D.templin-intarea-6706bis] allow OMNI MNs that desire time-varying MNPs to obtain short-lived prefixes. In that case, the identity of the MN can be used as a prefix delegation seed (e.g., a DHCPv6 Device Unique IDentifier (DUID) [RFC8415]). 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.

When a MN wishes to invoke DHCPv6 Prefix Delegation (PD) services, it sets the source address of an RS message to fe80:: and includes a DUID sub-option and a desired Prefix Length value in the RS message OMNI option. When the first-hop AR receives the RS message, it performs a PD exchange with the DHCPv6 service to obtain an IPv6 MNP of the requested length then returns an RA message with the OMNI LLA corresponding to the MNP as the destination address. When the MN receives the RA message, it provisons the PD to its downstream-attached networks and begins using the OMNI LLA in subsequent IPv6 ND messaging.

19. 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 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 16: 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".

   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         Interface Attributes         [RFCXXXX]
3         Traffic Selector             [RFCXXXX]
4         MS-Register                  [RFCXXXX]
5         MS-Release                   [RFCXXXX]
6         Network Acceess Identifier   [RFCXXXX]
7         Geo Coordinates              [RFCXXXX]
8         DHCP Unique Identifier (DUID) [RFCXXXX]
9-252     Unassigned
253-254   Experimental                 [RFCXXXX]
255       Reserved                     [RFCXXXX]
```

                 Figure 17: OMNI Option Sub-Type Values

20. 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.

   OMNI interfaces configured over secured ANET interfaces inherit the
   physical and/or link-layer security properties of the connected
   ANETs. 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, an asymmetric security service such as SEcure
   Neighbor Discovery (SEND) [RFC3971] with Cryptographically Generated

Addresses (CGAs) [RFC3972], the authentication option specified in
[RFC4380] or other protocol control message security mechanisms may
be necessary. 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.

The Mobility Service MUST provide strong network layer security for
control plane messages and forwading path integrity for data plane
messages. In one example, the AERO service
[I-D.templin-intarea-6706bis] constructs a spanning tree between
mobility service elements 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.

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.

21. 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.

22. 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.

23. References

23.1. 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>.

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

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

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

   [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>.

   [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>.

23.2. Informative References

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

   [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.templin-6man-dhcpv6-ndopt]
              Templin, F., "A Unified Stateful/Stateless Configuration
              Service for IPv6", draft-templin-6man-dhcpv6-ndopt-10
              (work in progress), June 2020.

   [I-D.templin-intarea-6706bis]
              Templin, F., "Asymmetric Extended Route Optimization
              (AERO)", draft-templin-intarea-6706bis-66 (work in
              progress), October 2020.

   [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>.

   [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>.

   [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>.

   [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>.

   [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>.

   [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>.

   [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 Heuristic Bitmap Encoding

   Adaptation of the OMNI option Interface Attributes Heuristic 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=2   |  Sub-length=N  |    ifIndex     |    ifType     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Provider ID  |  Link   |R| APS | 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 18: 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=N  |    ifIndex     |    ifType     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Provider ID  |  Link   |R| APS | 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 19: 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=N |    ifIndex    |    ifType     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Provider ID  |  Link  |R| APS |  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 20: 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 ANET. 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 ANET links, ARs can maintain an
   OMNI-specific unicast L2 address ("MSADDR"). For Ethernet-compatible
   ANETs, this specification reserves one Ethernet unicast address TBD2
   (see: Section 19). 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 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:

   o Major clarifications on aspects such as "hard/soft" PTB error
      messages

   o 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:

   o MTU

o Support for multi-hop ANETS such as ISATAP.

Differences from [draft-templin-6man-omni-interface-29](#) to [draft-templin-6man-omni-interface-30](#):

o Moved link-layer addressing information into the OMNI option on a
   per-ifIndex basis

o Renamed "ifIndex-tuple" to "Interface Attributes"

Differences from [draft-templin-6man-omni-interface-27](#) to [draft-templin-6man-omni-interface-28](#):

o Updates based on implementation expereince.

Differences from [draft-templin-6man-omni-interface-25](#) to [draft-templin-6man-omni-interface-26](#):

o Further clarification on "aggregate" RA messages.

o 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](#):

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

Differences from [draft-templin-6man-omni-interface-18](#) to [draft-templin-6man-omni-interface-19](#):

o SEND/CGA.

Differences from [draft-templin-6man-omni-interface-17](#) to [draft-templin-6man-omni-interface-18](#):

o Teredo

Differences from [draft-templin-6man-omni-interface-14](#) to [draft-templin-6man-omni-interface-15](#):

o Prefix length discussions removed.

Differences from [draft-templin-6man-omni-interface-12](#) to [draft-templin-6man-omni-interface-13](#):

o Teredo

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

o Major simplifications and clarifications on MTU and fragmentation.

o Document now updates RFC4443 and RFC8201.

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

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

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

o OMNI MNs in the open Internet

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

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

o Expanded "Transition Considerations".

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

o 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:

o 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:

o 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:

o Removed "Primary" flag and supporting text.

o 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:

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

o 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):

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