Network Working Group
Internet-Draft
Obsoletes: rfc5320, rfc5558, rfc5720, rfc6179, rfc6706 (if approved)
Intended status: Standards Track
Expires: November 2, 2019

F. Templin, Ed.
Boeing Research & Technology

May 1, 2019

Asymmetric Extended Route Optimization (AERO)
draft-templin-intarea-6706bis-12.txt

Abstract

This document specifies the operation of IP over tunnel virtual links using Asymmetric Extended Route Optimization (AERO). AERO interfaces use an IPv6 link-local address format that supports operation of the IPv6 Neighbor Discovery (ND) protocol and links ND to IP forwarding. Prefix delegation services are employed to manage the routing system. Dynamic multilink operation, mobility management, quality of service (QoS) signaling and route optimization are naturally supported through dynamic neighbor cache updates. Standard IP multicasting services are also supported. AERO is a widely-applicable tunneling solution especially well-suited to aviation services, mobile Virtual Private Networks (VPNs) and many other applications.

Status of This Memo

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

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

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

This Internet-Draft will expire on November 2, 2019.

Table of Contents

1. Introduction

Asymmetric Extended Route Optimization (AERO) fulfills the requirements of Distributed Mobility Management (DMM) [RFC7333] and route optimization [RFC5522] for aeronautical networking and other network mobility use cases. AERO is based on a Non-Broadcast, Multiple Access (NBMA) virtual link model known as the AERO link. The AERO link is configured over one or more underlying Internetworks, and nodes on the link can exchange IP packets via tunneling.

AERO links provide a cloud-based service where mobile nodes may use any Mobility Anchor Point (MAP) and fixed nodes may use any Gateway

on the link for efficient communications. Fixed nodes forward packets destined to other AERO nodes to the nearest Gateway, which forwards them through the cloud. A mobile node's initial packets are forwarded through the MAP, while direct routing is supported through route optimization once an initial session has been established. Both unicast and multicast communications are supported, and mobile nodes may efficiently move between locations while maintaining continuous communications with correspondents and without changing their IP Address.

The AERO service comprises Clients, Proxies, Servers, and Gateways that are seen as AERO link neighbors. Each node's AERO interface uses an IPv6 link-local address format (known as the AERO address) that supports operation of the IPv6 Neighbor Discovery (ND) [RFC4861] protocol and links ND to IP forwarding. A node's AERO interface can be configured over multiple underlying interfaces, and may therefore may appear as a single interface with multiple link-layer addresses. Each link-layer address is subject to change due to mobility and/or QoS fluctuations, and link-layer address changes are signaled by ND messaging the same as for any IPv6 link.

AERO Relays are interconnected in a secured private BGP overlay routing instance known as the "SPAN". The SPAN provides a (virtual) layer 2 bridging service to join the underlying Internetworks of multiple disjoint administrative domains into a single unified AERO link. Each AERO link instance is characterized by the set of Mobility Service Prefixes (MSPs) common to all mobile nodes. The link should extend to the point where a Gateway/MAP is on the optimal route from any correspondent node on the link, and provides a gateway between the underlying Internetwork and the SPAN. To the underlying Internetwork, the Gateway/MAP is the source of a route to its MSP, and hence uplink traffic to the mobile node is naturally routed to the nearest Gateway/MAP.

AERO assumes the use of PIM Sparse Mode in support of multicast communication. In support of Source Specific Multicast (SSM) when a Mobile Node is the source, the AERO cloud is included within the multicast distribution tree unless and until it is optimized out by use of AERO Direct Routing. In all other multicast scenarios there are no AERO dependencies.

AERO is applicable to a wide variety of use cases. For example, it can be used to coordinate the Virtual Private Network (VPN) links of mobile nodes (e.g., cellphones, tablets, laptop computers, etc.) that connect into a home enterprise network via public access networks using services such as OpenVPN [OVPN]. AERO is also applicable to aeronautical networking for both manned and unmanned aircraft where

the aircraft is treated as a mobile node that can connect an Internet of Things (IoT). Other applicable use cases are also in scope.

The following numbered sections present the AERO specification. The appendices at the end of the document are non-normative.

2. Terminology

The terminology in the normative references applies; the following terms are defined within the scope of this document:

IPv6 Neighbor Discovery (ND)
: an IPv6 control message service for coordinating neighbor relationships between nodes connected to a common link. AERO interfaces use the ND service specified in [RFC4861].

IPv6 Prefix Delegation (PD)
: a networking service for delegating IPv6 prefixes to nodes on the link. The nominal PD service is DHCPv6 [RFC8415], however alternate services (e.g., based on ND messaging) are also in scope [I-D.templin-v6ops-pdhost][I-D.templin-6man-dhcpv6-ndopt]. Most notably, a form of PD known as "Prefix Assertion" can be used if the prefix can be represented in the IPv6 source address of an ND message.

access network (ANET)
: a node's first-hop data link service network, e.g., a radio access network, cellular service provider network, corporate enterprise network, or the public Internet itself. For secured ANETs, link-layer security services such as IEEE 802.1X and physical-layer security prevent unauthorized access internally while border network-layer security services such as firewalls and proxies prevent unauthorized outside access. When there is no administrative boundary established between an ANET and the outside Internetwork, the ANET and Internetwork are one and the same.

ANET interface
: a node's attachment to a link in an ANET.

ANET address
: an IP address assigned to a node's interface connection to an ANET.

Internetwork (INET)
: a connected IP network topology with a coherent routing and addressing plan and that provides an Internetworking backbone service for ANETs. INETs also provide an underlay service over

which the AERO virtual link is configured. Example INETs include corporate enterprise networks, aviation networks, and the public Internet itself.

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

INET address
an IP address assigned to a node's interface connection to an INET.

AERO link
a Non-Broadcast, Multiple Access (NBMA) tunnel virtual overlay configured over one or more underlying INETs. Nodes on the AERO link appear as single-hop neighbors from the perspective of the virtual overlay even though they may be separated by many underlying INET hops.

AERO interface
a node's attachment to an AERO link. Since the addresses assigned to an AERO interface are managed for uniqueness, AERO interfaces do not require Duplicate Address Detection (DAD) and therefore set the administrative variable 'DupAddrDetectTransmits' to zero [RFC4862].

AERO address
an IPv6 link-local address assigned to an AERO interface and constructed as specified in Section 3.4.

base AERO address
the lowest-numbered AERO address aggregated by the MNP (see Section 3.4).

Mobility Service Prefix (MSP)
an IP prefix assigned to the AERO link and from which more-specific Mobile Network Prefixes (MNPs) are derived.

Mobile Network Prefix (MNP)
an IP prefix derived from an MSP and delegated to an AERO Client or Gateway.

Fixed Node (FN)
a node on an INET link serviced by an AERO Gateway.

Mobile Node (MN)
an AERO Client and all of its downstream-attached networks.

Mobile Router (MR)

a MN's on-board router that forwards packets between any downstream-attached networks and the AERO link.

Correspondent Node (CN)
a MN or FN that is reachable over the AERO link

AERO node
a node that is connected to an AERO link.

AERO Client ("Client")
an AERO node that connects to one or more ANETs and requests MNP PDs from one or more AERO Servers. Following PD, the Client assigns a Client AERO address to the AERO interface for use in ND exchanges with other AERO nodes. A Client can also be deployed on the same platform as a Server, and a node that acts as a Client on one AERO interface can also act as an AERO Server on a different AERO interface.

AERO Server ("Server")
an INET node that configures an AERO interface to provide default forwarding services and a Mobility Anchor Point (MAP) for AERO Clients. The Server assigns an administratively-provisioned AERO address to the AERO interface to support the operation of the ND/PD services, and advertises all of its associated MNPs via BGP peerings with Relays.

AERO Gateway ("Gateway")
a combined AERO Server/Client that also provides forwarding services between CNs on the AERO link and FNs on INET links. AERO Gateways are provisioned with MNPs used for numbering nodes and networks on downstream-attached INET interfaces (i.e., the same as for an AERO Client) and run a dynamic routing protocol to discover any native INET prefixes. In both cases, the Gateway advertises the MSP(s) to FNs in downstream-attached INET networks, and distributes all of its associated MNPs and native INET prefixes via BGP peerings with Relays (i.e., the same as for an AERO Server).

AERO Relay ("Relay")
an INET node that provides both layer-3 routing and layer-2 bridging services (as well as a security trust anchor) for nodes on an AERO link. As a router, the Relay forwards data packets using standard IP forwarding. As a bridge, the Relay securely forwards control messages between other AERO nodes. AERO Relays peer with Servers and other Relays to discover the full set of MNPs

AERO Proxy ("Proxy")

a node that provides proxying services between Clients in an ANET and Servers in external INETs. The AERO Proxy is a conduit between the ANET and external INETs in the same manner as for common web proxies, and behaves in a similar fashion as for ND proxies [RFC4389].

Spanning Partitioned AERO Networks (SPAN)
a means for bridging disjoint INETs as segments (or, partitions) of a unified AERO link, i.e., the same as for a bridged campus LAN. The SPAN is a mid-layer encapsulation service in the AERO routing system that supports a unified AERO link view for all segments. Each segment in the SPAN is a distinct INET.

SPAN Service Prefix (SSP)
a global or unique local /96 IPv6 prefix assigned to the AERO link to support SPAN services.

SPAN Partition Prefix (SPP)
a sub-prefix of the SPAN Service Prefix uniquely assigned to a single AERO link segment.

SPAN Address
a global or unique local IPv6 address taken from a SPAN Partition Prefix.

ingress tunnel endpoint (ITE)
an AERO interface endpoint that injects encapsulated packets into an AERO link.

egress tunnel endpoint (ETE)
an AERO interface endpoint that receives encapsulated packets from an AERO link.

link-layer address
an IP address used as an encapsulation header source or destination address from the perspective of the AERO interface. When UDP encapsulation is used, the UDP port number is also considered as part of the link-layer address. From the perspective of the AERO interface, the link-layer address is either an INET address for intra-segment encapsulation or a SPAN address for inter-segment encapsulation.

network layer address
the source or destination address of an encapsulated IP packet presented to the AERO interface.

end user network (EUN)

an internal virtual or external edge IP network that an AERO Client or Gateway connects to the rest of the network via the AERO interface. The Client/Gateway sees each EUN as a "downstream" network, and sees the AERO interface as the point of attachment to the "upstream" network.

Mobility Anchor Point (MAP)
an AERO Server that is currently tracking and reporting the mobility events of its associated Clients.

Mobile Router (MR)
a router on an AERO Client that provides routing services between the Client's EUNs and the AERO interface.

MAP List
a geographically and/or topologically referenced list of IP addresses of Servers for the AERO link.

Distributed Mobility Management (DMM)
a BGP-based overlay routing service coordinated by Servers and Relays that tracks all MAP-to-Client associations.

Route Optimization Source (ROS)
the AERO node nearest the source Client that initiates route optimization. The ROS may be one of the Client's Servers, Proxies or in some cases even the Client itself.

Route Optimization Responder (ROR)
a Server of the target Client to which a route optimization request is directed. The ROR (acting as a MAP) returns the most current information about the target Client's underlying interface connections.

Throughout the document, the simple terms "Client", "Server", "Relay", "Proxy" and "Gateway" refer to "AERO Client", "AERO Server", "AERO Relay", "AERO Proxy" and "AERO Gateway", respectively. Capitalization is used to distinguish these terms from DHCPv6 client/server/relay [RFC8415].

The terminology of DHCPv6 [RFC8415] and IPv6 ND [RFC4861] (including the names of node variables, messages and protocol constants) is used throughout this document. Also, the term "IP" is used to generically refer to either Internet Protocol version, i.e., IPv4 [RFC0791] or IPv6 [RFC8200].

The terms Mobility Anchor Point (MAP), Mobile Router (MR) and Distributed Mobility Management (DMM) are used in the same sense as standard Internetworking terminology.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. Lower case uses of these words are not to be interpreted as carrying RFC2119 significance.

3. Asymmetric Extended Route Optimization (AERO)

The following sections specify the operation of IP over Asymmetric Extended Route Optimization (AERO) links:

3.1. AERO Link Reference Model

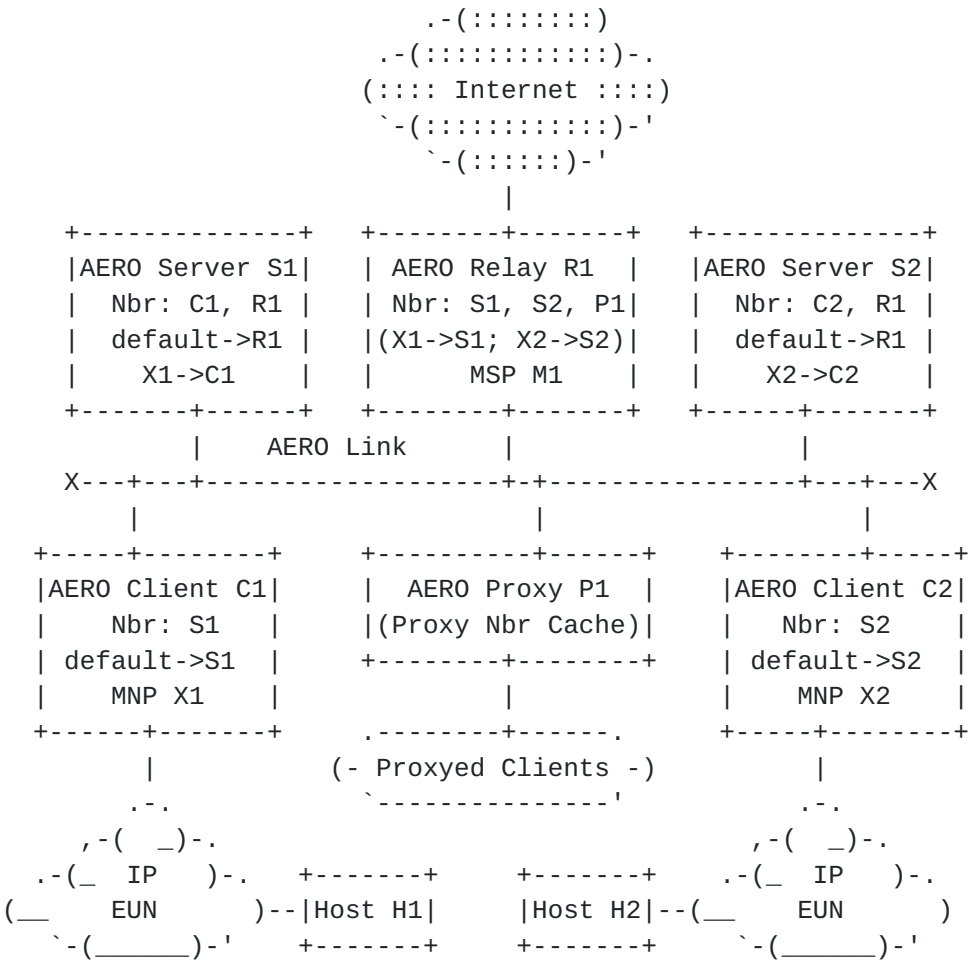

Figure 1: AERO Link Reference Model

Figure 1 presents the AERO link reference model. In this model:

- the AERO link is an overlay Layer 3 service configured over one or more underlying INETs which may be managed by different

administrative authorities and have incompatible protocols and/or addressing plans.

- AERO Relay R1 aggregates Mobility Service Prefix (MSP) M1, discovers Mobile Network Prefixes (MNPs) X*, acts as a default router for its associated Servers and Proxies (S1, S2, P1), and connects the AERO link to the external Internet. AERO Relays also use the SPAN service to bridge disjoint segments (i.e., INETs) of a partitioned AERO link.

- AERO Servers S1 and S2 associate with Relay R1 and also act as Mobility Anchor Points (MAPs) and default routers for their associated Clients C1 and C2.

- AERO Clients C1 and C2 associate with Servers S1 and S2, respectively. They receive Mobile Network Prefix (MNP) delegations X1 and X2, and also act as default routers for their associated physical or internal virtual EUNs. Simple hosts H1 and H2 attach to the EUNs served by Clients C1 and C2, respectively.

- AERO Proxy P1 provides proxy services for AERO Clients in secured enclaves that cannot associate directly with other AERO link neighbors.

Each node on the AERO link maintains an AERO interface neighbor cache and an IP forwarding table the same as for any link. Although the figure shows a limited deployment, in common operational practice there will normally be many additional Relays, Servers, Clients and Proxies.

3.2. AERO Node Types

AERO Relays provide both layer-3 routing and layer-2 bridging services (as well as a security trust anchor) for nodes on an AERO link. As a router, the Relay forwards data packets using standard IP forwarding. As a bridge, the Relay securely forwards control messages between Proxies and Servers both within the same INET and between disjoint INETs. Each Relay also peers with Servers and other Relays in a dynamic routing protocol instance to provide a Distributed Mobility Management (DMM) service for the list of active MNPs (see Section 3.3). Relays forward packets between neighbors connected to the same AERO link and also forward packets between the AERO link and the outside world. Relays present the AERO link as a set of one or more Mobility Service Prefixes (MSPs). Relays maintain neighbor cache entries for Servers and Proxies, and maintain an IP forwarding table entry for each Mobile Network Prefix (MNP).

AERO Servers provide default forwarding services and a Mobility Anchor Point (MAP) for AERO Client Mobile Nodes (MNs). Each Server also peers with Relays in a dynamic routing protocol instance to advertise its list of associated MNPs (see Section 3.3). Servers facilitate PD exchanges with Clients, where each delegated prefix becomes an MNP taken from an MSP. Servers forward packets between AERO interface neighbors, and maintain neighbor cache entries for Relays. They also maintain both neighbor cache entries and IP forwarding table entries for each of their associated Clients, and track each Client's mobility profiles.

AERO Clients act as requesting Mobile Routers (MRs) to receive MNPs through PD exchanges with AERO Servers over the AERO link, and distribute the MNPs to nodes on EUNs. Each Client can associate with a single Server or with multiple Servers, e.g., for fault tolerance, load balancing, etc. Each IPv6 Client receives at least a /64 IPv6 MNP, and may receive even shorter prefixes. Similarly, each IPv4 Client receives at least a /32 IPv4 MNP (i.e., a singleton IPv4 address), and may receive even shorter prefixes. Clients maintain an AERO interface neighbor cache entry for each of their associated Servers as well as for each of their correspondent Clients. A Client may also be co-resident on the same physical or virtual platform as a Server; in that case, the Client and Server behave as a single functional unit and without the need for any Client/Server control messaging.

AERO Proxies provide a conduit for AERO Clients in ANETs to associate with AERO Servers in external INETs. The Client sends all of its control plane messages to the Server via the Proxy, which intercepts them before they leave the ANET. The Proxy forwards the Client's control and data plane messages to and from the Client's current Server(s). The Proxy may also discover a better route toward a target destination via AERO route optimization, in which case future outbound data packets would be forwarded via the more direct route. Proxies maintain AERO interface neighbor cache entries for Relays, i.e., the same as Servers. The Proxy function is specified in Section 3.16.

AERO Gateways are combined Client/Servers that also provide forwarding services between correspondent nodes (CNs) on the AERO interface and fixed nodes (FNs) on INET interfaces. AERO Gateways are provisioned with MNPs used for numbering nodes and networks on downstream-attached INET interfaces (i.e., the same as for an AERO Client) and may also run a dynamic routing protocol to discover any native INET prefixes. In both cases, the Gateway advertises the MSP(s) to correspondent nodes in downstream-attached INET networks, and distributes all of its associated MNPs and native INET prefixes via BGP peerings with Relays.

AERO Relays, Servers, Proxies and Gateways are critical infrastructure elements in fixed (i.e., non-mobile) INET deployments and hence have permanent and unchanging INET addresses. AERO Clients are MNs that connect via ANET interfaces, i.e., their ANET addresses may change when the Client moves to a new ANET connection.

3.3. AERO Routing System

The AERO routing system comprises a private instance of the Border Gateway Protocol (BGP) [RFC4271] that is coordinated between Relays and Servers and does not interact with either the public Internet BGP routing system or any underlying INET routing systems. Relays advertise only a small and unchanging set of MSPs to the outside world instead of the full dynamically changing set of MNPs.

In a reference deployment, each Server is configured as an Autonomous System Border Router (ASBR) for a stub Autonomous System (AS) using an AS Number (ASN) that is unique within the BGP instance, and each Server further uses eBGP to peer with one or more Relays but does not peer with other Servers. Each INET of a multi-segment AERO link must include one or more Relays, which peer with the Servers and Proxies within that INET. All Relays within the same INET are members of the same hub AS using a common ASN, and use iBGP to maintain a consistent view of all active MNPs currently in service. The Relays of different INETs peer with one another using eBGP.

Each Server maintains a working set of associated MNPs and native INET prefixes, and dynamically announces new prefixes and withdraws departed prefixes in its eBGP updates to Relays. Clients are expected to remain associated with their current Servers for extended timeframes, however Servers SHOULD selectively suppress updates for impatient Clients that repeatedly associate and disassociate with them in order to dampen routing churn. Servers that are configured as Gateways advertise the MSP into the INET and forward packets between INET interfaces and the AERO interface.

Each Relay configures a black-hole route for each of its MSPs. By black-holing the MSPs, the Relay will maintain forwarding table entries only for the MNPs that are currently active, and packets destined to all other MNPs will correctly incur Destination Unreachable messages due to the black hole route. Relays do not send eBGP updates for MNPs to Servers, but instead only originate a default route. In this way, Servers have only partial topology knowledge (i.e., they know only about the MNPs of their directly associated Clients) and they forward all other packets to Relays which have full topology knowledge.

For IPv6 MNPs, the AERO routing system includes only IPv6 routes. For IPv4 MNPs, the AERO routing system includes both IPv4 routes and also IPv6 routes based on the IPv4-mapped IPv6 address corresponding to the MNP and with prefix length set to 96 plus the length of the IPv4 prefix. (For example, if the IPv4 MNP is 192.0.2.0/24 then the IPv4-mapped prefix is 0:0:0:0:0:FFFF:192.0.2.0/120.)

Scaling properties of the AERO routing system are limited by the number of BGP routes that can be carried by Relays. As of 2015, the global public Internet BGP routing system manages more than 500K routes with linear growth and no signs of router resource exhaustion [BGP]. More recent network emulation studies have also shown that a single Relay can accommodate at least 1M dynamically changing BGP routes even on a lightweight virtual machine, i.e., and without requiring high-end dedicated router hardware.

Therefore, assuming each Relay can carry 1M or more routes, this means that at least 1M Clients can be serviced by a single set of Relays. A means of increasing scaling would be to assign a different set of Relays for each set of MSPs. In that case, each Server still peers with one or more Relays, but institutes route filters so that BGP updates are only sent to the specific set of Relays that aggregate the MSP. For example, if the MSP for the AERO link is 2001:db8::/32, a first set of Relays could service the MSP segment 2001:db8::/40, a second set of Relays could service 2001:db8:0100::/40, a third set could service 2001:db8:0200::/40, etc.

Assuming up to 1K sets of Relays, the AERO routing system can then accommodate 1B or more MNPs with no additional overhead (for example, it should be possible to service 1B /64 MNPs taken from a /34 MSP and even more for shorter prefixes). In this way, each set of Relays services a specific set of MSPs that they advertise to the native Internetwork routing system, and each Server configures MSP-specific routes that list the correct set of Relays as next hops. This arrangement also allows for natural incremental deployment, and can support small scale initial deployments followed by dynamic deployment of additional Clients, Servers and Relays without disturbing the already-deployed base.

A full discussion of the BGP-based routing system used by AERO is found in [I-D.ietf-rtgwg-atn-bgp]. The system provides for Distributed Mobility Management (DMM) per the distributed mobility anchoring architecture [I-D.ietf-dmm-distributed-mobility-anchoring].

3.4. AERO Addresses

A Client's AERO address is an IPv6 link-local address with an interface identifier based on the Client's delegated MNP. Relay, Server and Proxy AERO addresses are assigned from the range fe80::/96 and include an administratively-provisioned value in the lower 32 bits.

For IPv6, Client AERO addresses begin with the prefix fe80::/64 and include in the interface identifier (i.e., the lower 64 bits) a 64-bit prefix taken from one of the Client's IPv6 MNPs. For example, if the AERO Client receives the IPv6 MNP:

   2001:db8:1000:2000::/56

it constructs its corresponding AERO addresses as:

   fe80::2001:db8:1000:2000

   fe80::2001:db8:1000:2001

   fe80::2001:db8:1000:2002

   ... etc. ...

   fe80::2001:db8:1000:20ff

For IPv4, Client AERO addresses are based on an IPv4-mapped IPv6 address formed from an IPv4 MNP and with a Prefix Length of 96 plus the MNP prefix length. For example, for the IPv4 MNP 192.0.2.32/28 the IPv4-mapped IPv6 MNP is:

   0:0:0:0:0:FFFF:192.0.2.16/124

The Client then constructs its AERO addresses with the prefix fe80::/64 and with the lower 64 bits of the IPv4-mapped IPv6 address in the interface identifier as:

   fe80::FFFF:192.0.2.16

   fe80::FFFF:192.0.2.17

   fe80::FFFF:192.0.2.18

   ... etc. ...

   fe80:FFFF:192.0.2.31

Relay, Server and Proxy AERO addresses are allocated from the range fe80::/96, and MUST be managed for uniqueness. The lower 32 bits of the AERO address includes a unique integer value (e.g., fe80::1, fe80::2, fe80::3, etc.) as assigned by the administrative authority for the link. If the link spans multiple segments (i.e., multiple INETs), the AERO addresses are assigned to each INET in 1x1 correspondence with SPAN addresses (see: Section 3.5). The address fe80:: is reserved as the IPv6 link-local Subnet Router Anycast address [RFC4291], and the address fe80::ffff:ffff is reserved as the unspecified AERO address; hence, these values are not available general assignment.

The lowest-numbered AERO address from a Client's MNP delegation serves as the "base" AERO address (for example, for the MNP 2001:db8:1000:2000::/56 the base AERO address is fe80::2001:db8:1000:2000). The Client then assigns the base AERO address to the AERO interface and uses it for the purpose of maintaining the neighbor cache entry. The Server likewise uses the AERO address as its index into the neighbor cache for this Client.

If the Client has multiple AERO addresses (i.e., when there are multiple MNPs and/or MNPs with prefix lengths shorter than /64), the Client originates ND messages using the base AERO address as the source address and accepts and responds to ND messages destined to any of its AERO addresses as equivalent to the base AERO address. In this way, the Client maintains a single neighbor cache entry that may be indexed by multiple AERO addresses.

Client AERO addresses can be statelessly transformed into an IPv6 Subnet Router Anycast address and vice-versa. For example, for the AERO address fe80::2001:db8:2000:3000 the corresponding Subnet Router Anycast address is 2001:db8:2000:3000::. In the same way, for the IPv6 Subnet Router Anycast address 2001:db8:1:2:: the corresponding AERO address is fe80::2001:db8:1:2. In other words, the low-order 64 bits of an AERO address can be used as the high-order 64 bits of a Subnet Router Anycast address, and vice-versa.

3.5. Spanning Partitioned AERO Networks (SPAN)

In the simplest case, an AERO link configured over a single INET appears as a single unified link with a consistent underlying network addressing plan. In that case, all nodes on the link can exchange packets via encapsulation with INET addresses, since the underlying INET is connected. In common practice, however, an AERO link may be partitioned into multiple "segments", where each segment is a distinct INET managed under a different administrative authority (e.g., as for worldwide aviation service providers such as ARINC, SITA, Inmarsat, etc.). Individual INETs may themselves be

partitioned internally, in which case each internal partition is seen as a separate segment.

The addressing plan of each segment is consistent internally but will often bear no relation to the addressing plans of other segments. Each segment is also likely to be separated from others by network security devices (e.g., firewalls, proxies, packet filtering gateways, etc.), and in many cases disjoint segments may not even have any common physical link connections at all. Therefore, nodes can only be assured of exchanging packets directly with nodes in the same segment, and not with nodes in other segments. The only means for joining the segments therefore is through inter-domain peerings between AERO Relays acting as bridges.

The same as for traditional campus LANs, multiple AERO link segments can be joined into a single unified link via a bridging service termed the "SPAN". The SPAN performs link-layer packet forwarding between segments (i.e., bridging) without decrementing the network-layer TTL/Hop Limit. The SPAN model is depicted in Figure 2:

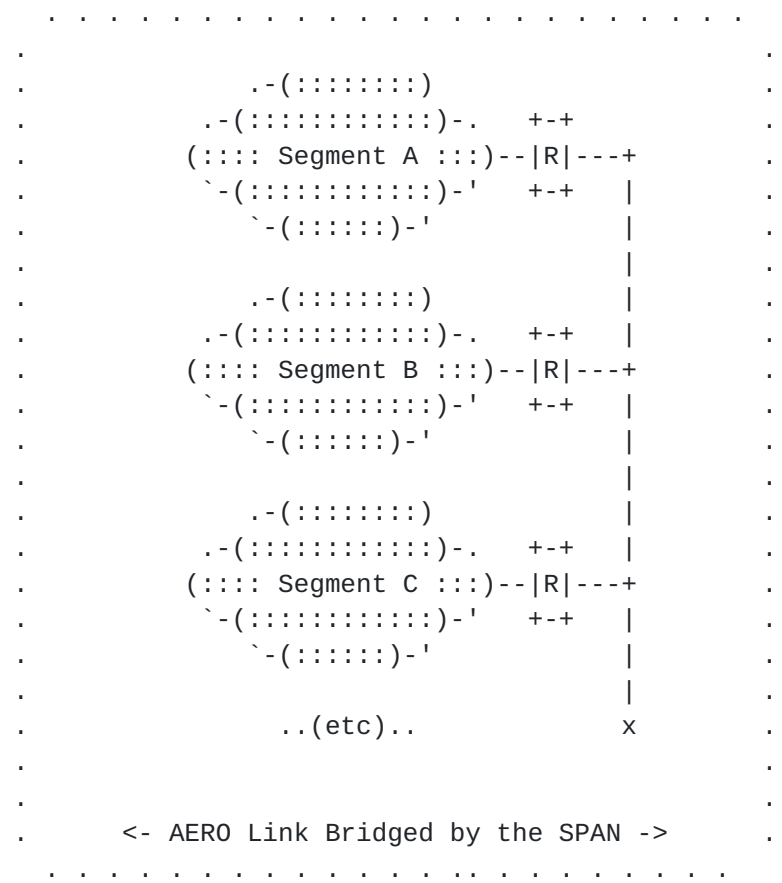

Figure 2: The SPAN

To support the SPAN, AERO links require a reserved /96 IPv6 "SPAN Service Prefix (SSP)". Although any routable IPv6 prefix can be used, a Unique Local Address (ULA) prefix (e.g., fd00::/96) [RFC4389] is preferred since border routers are commonly configured to prevent packets with ULAs from being injected into the AERO link by an external IPv6 node and from leaking out of the AERO link to the outside world.

Each segment in the SPAN assigns a unique sub-prefix of the SSP termed a "SPAN Partition Prefix (SPP)". For example, a first segment could assign fd00::/116, a second could assign fd00::1000/116, a third could assign fd00::2000/116, etc. The administrative authorities for each segment must therefore coordinate to assure mutually-exclusive SPP assignments, but internal provisioning of the SPP is a local consideration for each administrative authority.

A "SPAN address" is an address taken from a SPP and assigned to a Relay, Server or Proxy AERO interface. SPAN addresses are formed by simply replacing the upper portion of an administratively-assigned AERO address with the SPP. For example, if the SPP is fd00::/116, the SPAN address formed from the AERO address fe80::1 is simply fd00::1. (As with AERO addresses, the values ::0 and ::ffff:ffff are reserved and not available for general assignment.)

An "INET address" is an address of a node's interface connection to an INET segment. Each Relay, Server and Proxy connected to the same segment maintains a static mapping of AERO/SPAN addresses to INET addresses for all fixed infrastructure elements in that segment. For example, if a Server has AERO/SPAN addresses fe80::1/fd00::1 and INET address 192.0.2.100, then all other Relays, Servers and Proxys in that segment keep a static mapping for those addresses. In that way, any of the AERO/SPAN/INET addresses can be derived from a static lookup without the need for protocol messaging.

AERO Relays serve as bridges to join multiple segments into a unified AERO link over multiple diverse administrative domains. They support the bridging function by first establishing forwarding table entries for their SPPs either via standard BGP routing or static routes. For example, if three Relays (Relays 'A', 'B' and 'C') from different segments serviced the SPPs fd00::1000/116, fd00::2000/116 and fd00::3000/116 respectively, then the forwarding tables in each Relay are as follows:

A: fd00::1000/116->local, fd00::2000/116->B, fd00::3000/116->C

B: fd00::1000/116->A, fd00::2000/116->local, fd00::3000/116->C

C: fd00::1000/116->A, fd00::2000/116->B, fd00::3000/116->local

These forwarding table entries are permanent and never change, since they correspond to fixed infrastructure elements in their respective segments. This point is of critical importance, since it provides the basis for a link-layer forwarding service that cannot be disrupted by routing updates due to node mobility.

With the SPPs in place in each Relay's forwarding table, control and data packets sent between AERO nodes in different segments can therefore be carried over the SPAN via encapsulation. For example, when a source node in segment A forwards a packet with IPv6 address 2001:db8:1:2::1 to a destination node in segment C with IPv6 address 2001:db8:1000:2000::1, it first encapsulates the packet in a SPAN header with source SPAN address taken from fd00::1000/116 (e.g., fd00::1001) and destination SPAN address taken from fd00::3000/116 (e.g., fd00::3001). Next, it encapsulates the SPAN message in an INET header with source address set to its own INET address (e.g., 192.0.2.100) and destination set to the INET address of a Relay (e.g., 192.0.2.1).

SPAN encapsulation is based on Generic Packet Tunneling in IPv6 [RFC2473]; the encapsulation format in the above example is shown inFigure 3:

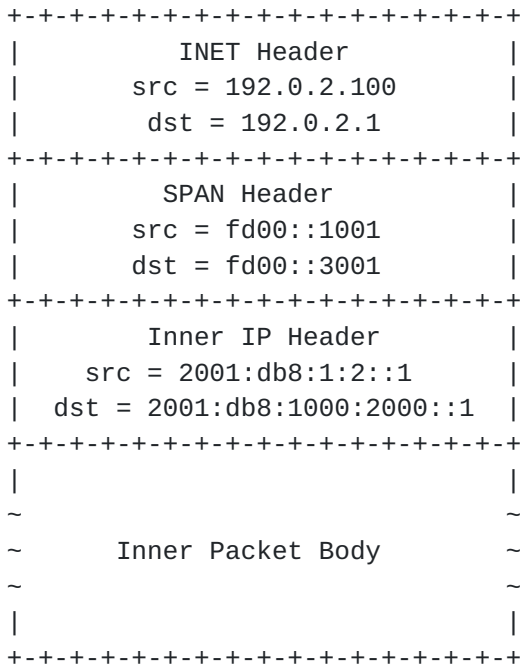

Figure 3: SPAN Encapsulation

In this format, the inner IP header and packet body are the original IP packet, the SPAN header is an IPv6 header prepared according to [RFC2473], and the INET header is prepared according to Section 3.10.

A packet is said to be "forwarded/sent into the SPAN" when it is encapsulated as described above then forwarded to a neighboring Relay. This terminology appears throughout the remainder of the document.

This gives rise to a routing system that contains both MNP routes that may change dynamically due to regional node mobility and SPAN routes that never change. The Relays can therefore provide link-layer bridging by sending packets into the SPAN instead of network-layer routing according to MNP routes. As a result, opportunities for packet loss due to node mobility between different segments are mitigated.

NB: With reference to Figure 3, the destination SPAN address may not be known in advance for the first few packets of a flow sent via the SPAN. In that case, the SPAN destination address is set to the subnet router anycast address corresponding to the original packet's destination, and the SPAN routing system will direct the packet to the correct SPAN egress node. (In the above example, the subnet router anycast address is simply 2001:db8:1000:2000::.)

3.6. AERO Interface Characteristics

AERO interfaces use encapsulation (see: Section 3.10) to exchange packets with neighbors attached to the AERO link.

AERO interfaces maintain a neighbor cache for tracking per-neighbor state the same as for any interface. AERO interfaces use ND messages including Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS) and Neighbor Advertisement (NA) for neighbor cache management.

AERO interface ND messages include one or more Source/Target Link-Layer Address Options (S/TLLAOs) formatted as shown in Figure 4:

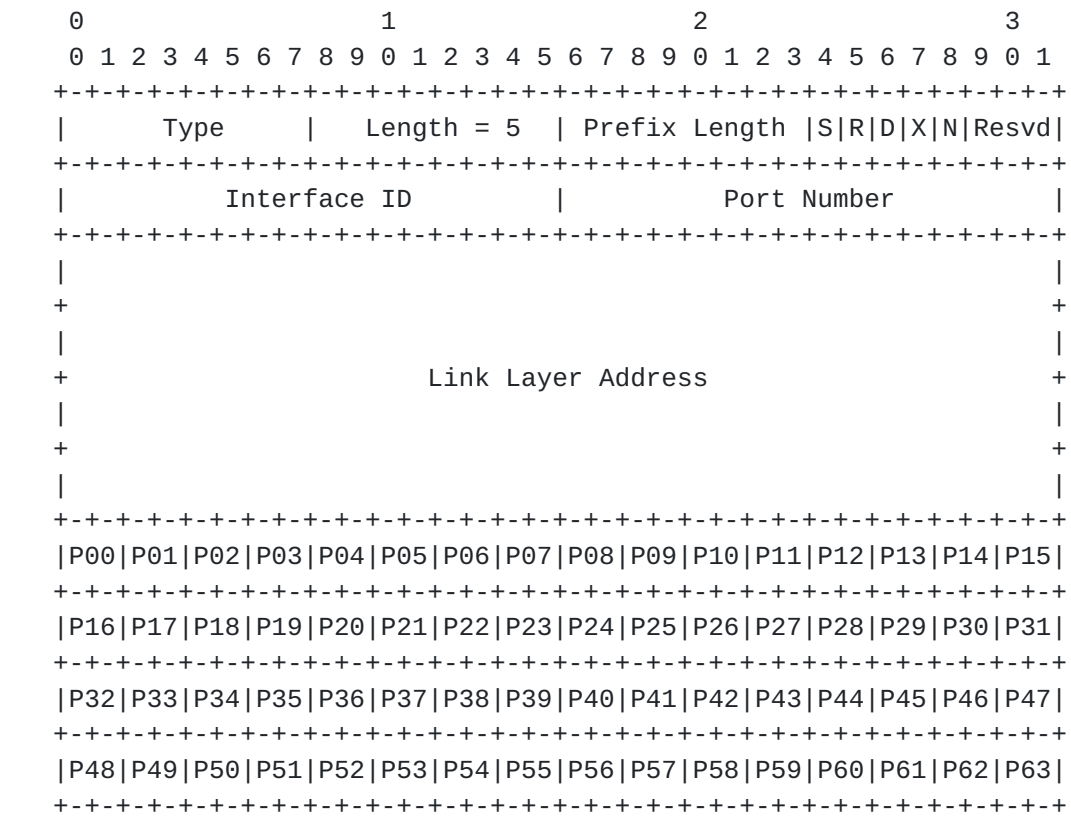

Figure 4: AERO Source/Target Link-Layer Address Option (S/TLLAO) Format

In this format:

- Type is set to '1' for SLLAO or '2' for TLLAO.

- Length is set to the constant value '5' (i.e., 5 units of 8 octets).

- Prefix Length is set to the MNP prefix length in an ND message for the Client AERO address found in the source (RS), destination (RA) or target (NA) address; otherwise set to 0 if the message is not being used for PD or neighbor prefix discovery. If the message contains multiple SLLAOs, only the Prefix Length value in the SLLAO with S set to 1 is consulted and the values in other SLLAOs are ignored.

- S (the 'Source' bit) is set to '1' in the S/TLLAO of an ND message that corresponds to the ANET/INET interface over which the ND message is sent, and set to 0 in all other S/TLLAOs.

- R (the "Release" bit) is set to '1' in the SLLAO of an RS message sent for the purpose of departing from a Server; otherwise, set to

'0'. If the message contains multiple SLLAOs, only the R value in the SLLAO with S set to 1 is consulted and the values in other SLLAOs are ignored. The Server places the corresponding neighbor cache entry in the DEPARTED state and releases the corresponding PD, then returns an RA with Router Lifetime set to '0'.

- D (the "Disable" bit) is set to '1' in the S/TLLAOs of an RS/NA message for each Interface ID that is to be disabled in the neighbor cache entry; otherwise, set to '0'. If the message contains an S/TLLAO with Interface ID 0xffff, the node places the corresponding neighbor cache entry in the DEPARTED state. If the message contains multiple S/TLLAOs the D value in each S/TLLAO is consulted.

- X (the "proXy" bit) is set to '1' in the SLLAO of an RS/RA message by the Proxy when there is a Proxy in the path; otherwise, set to '0'. If the message contains multiple SLLAOs, only the X value in the first SLLAO is consulted and the values in other SLLAOs are ignored.

- N (the "(Network Address) Translator (NAT)" bit) is set to '1' in the SLLAO of an RA message by the Server if there is a translator in the path; otherwise, set to '0'. If the message contains multiple SLLAOs, only the N value in the first SLLAO is consulted and the values in other SLLAOs are ignored.

- Resvd is set to the value '0' on transmission and ignored on receipt.

- Interface ID is set to a 16-bit integer value corresponding to an AERO node's ANET/INET interface. Once the node has assigned an Interface ID to an ANET interface, the assignment must remain unchanged until the node fully detaches from the AERO link. The value 0xffff is reserved as the AERO Server's INET Interface ID, i.e., Servers MUST use Interface ID 0xffff, and Clients MUST number their ANET Interface IDs with values in the range of 0-0xfffe.

- Port Number and Link Layer Address are set to the addresses used by the AERO node when it sends encapsulated packets over the specified ANET/INET interface (or to '0' when the addresses are left unspecified). When UDP is not used as part of the encapsulation, Port Number is set to '0'. When the encapsulation IP address family is IPv4, IP Address is formed as an IPv4-mapped IPv6 address as specified in Section 3.4.

- P(i) is a set of Preferences that correspond to the 64 Differentiated Service Code Point (DSCP) values [RFC2474]. Each

P(i) is set to the value '0' ("disabled"), '1' ("low"), '2' ("medium") or '3' ("high") to indicate a QoS preference level for packet forwarding purposes.

A Client's AERO interface may be configured over multiple ANET interface connections. For example, common mobile handheld devices have both wireless local area network ("WLAN") and cellular wireless links. These links are typically used "one at a time" with low-cost WLAN preferred and highly-available cellular wireless as a standby. In a more complex example, aircraft frequently have many wireless data link types (e.g. satellite-based, cellular, terrestrial, air-to-air directional, etc.) with diverse performance and cost properties.

A Client's ANET interfaces are classified as follows:

- Native interfaces connect to the open INET, and have a global IP address that is reachable from any INET correspondent.

- NATed interfaces connect to an ANET behind a Network Address Translator (NAT). The NAT does not participate in any AERO control message signaling, but the AERO Server can issue control messages on behalf of the Client. Clients that are behind a NAT are required to send periodic keepalive messages to keep NAT state alive when there are no data packets flowing.

- VPNed interfaces use security encapsulation over the ANET to a Virtual Private Network (VPN) server that also acts as an AERO Server. As with NATed links, the AERO Server can issue control messages on behalf of the Client, but the Client need not send periodic keepalives in addition to those already used to maintain the VPN connection.

- Proxyed interfaces connect to an ANET that is separated from the open INET by an AERO Proxy. Unlike NATed and VPNed interfaces, the AERO Proxy can actively issue control messages on behalf of the Client.

- Direct interfaces connect the Client directly to a neighbor without crossing any ANET/INET paths. An example is a line-of-sight link between a remote pilot and an unmanned aircraft.

If a Client's multiple ANET interfaces are used "one at a time" (i.e., all other interfaces are in standby mode while one interface is active), then ND messages include only a single S/TLLAO with Interface ID set to a constant value. In that case, the Client would appear to have a single ANET interface but with a dynamically changing ANET address.

If the Client has multiple active ANET interfaces, then from the perspective of ND it would appear to have multiple link-layer addresses. In that case, ND messages MAY include multiple S/TLLAOs -- each with an Interface ID that corresponds to a specific ANET interface. The S bit must be set to 1 in the S/TLLAO corresponding to the AERO node's ANET interface used to transmit the message and set to 0 in all other S/TLLAOs.

When the Client includes an S/TLLAO for an ANET interface for which it is aware that there is a NAT on the path to the Server, or when a node includes an S/TLLAO solely for the purpose of announcing new QoS preferences, the node MAY set both Port Number and Link-Layer Address to 0 to indicate that the addresses are unspecified at the network layer and must instead be derived from the link-layer encapsulation headers.

3.7. AERO Interface Initialization

3.7.1. AERO Relay Behavior

When a Relay enables an AERO interface, it first assigns an administratively-provisioned AERO address (e.g., fe80::1) and its companion SPAN address (e.g., fd00::1), where each address MUST be unique among all AERO nodes on the link. The Relay also configures a neighbor cache entry for Servers, Gateways and Proxys on the local segment, and maintains a list of INET address mappings for all fixed infrastructure elements on the local segment. The Relay then engages in a BGP routing protocol session with Servers/Gateways on the local segment and other Relays on the AERO link (see: Section 3.3). Each Relay subsequently maintains an IP forwarding table entry for each active MNP covered by its MSP(s) as well as for each SPAN prefix.

3.7.2. AERO Server Behavior

When a Server enables an AERO interface, it assigns AERO/SPAN addresses and maintains a list of INET address mappings the same as for Relays. The Server further configures a service to facilitate ND/PD exchanges with AERO Clients, maintains neighbor cache entries for one or more Relays on the link, and manages per-Client neighbor cache entries and IP forwarding table entries based on control message exchanges. The Server also engages in a BGP routing protocol session with its neighboring Relays via the AERO interface, and also engages in a dynamic routing protocol over its INET interfaces (see: Section 3.3).

When the Server receives an NS/RS message on the AERO interface it authenticates the message and returns a solicited NA/RA message. (When the Server receives an unsolicited NA message, it likewise

authenticates the message and processes it locally.) The Server further provides a simple link-layer conduit between AERO interface neighbors. In particular, when a packet sent by a source CN arrives on the Server's AERO interface and is destined to a CN belonging to a MNP not assigned to one of the Server's INET interfaces, the Server forwards the packet from within the AERO interface at the link layer without ever disturbing the network layer.

3.7.3. AERO Gateway Behavior

Gateways are simply Servers that run a dynamic routing protocol between the AERO and INET interfaces. The Gateway provisions MNPs to networks on the downstream-attached INET interfaces (i.e., the same as a Client would do) and advertises the MSP(s) for the AERO link over the INET interfaces.

3.7.4. AERO Proxy Behavior

When a Proxy enables an AERO interface, it assigns AERO/SPAN addresses and maintains a list of INET address mappings the same as for Relays, Servers and Gateways. The Proxy further maintains neighbor cache entires for one or more Relays, and maintains per-Client neighbor cache entries based on control message exchanges. Proxies forward packets between each Client and their associated Servers and neighbors.

When the Proxy receives an RS message from a Client, it creates an incomplete neighbor cache entry and sends a proxyed RS message to a Server via the SPAN while using its own INET address as the source address. When the Server returns an RA message, the Proxy completes the proxy neighbor cache entry based on autoconfiguration information in the RA and sends a proxyed RA to the Client while using its own ANET address as the source address. The Client, Server and Proxy will then have the necessary state for managing the proxy neighbor association.

3.7.5. AERO Client Behavior

When a Client enables an AERO interface, it sends RS messages with ND/PD parameters over an ANET interface to one or more AERO Servers, which return RA messages with corresponding PD parameters. (The RS/RA messages may pass through a Proxy in the case of a Client's Proxyed interface.) See [I-D.templin-6man-dhcpv6-ndopt] for the types of ND/PD parameters that can be included in the RS/RA message exchanges.

After the initial ND/PD message exchange, the Client assigns AERO addresses to the AERO interface based on the delegated prefix(es).

The Client can then register additional ANET interfaces with the Server by sending a simple RS message (i.e., one with no PD parameters) over each ANET interface using its base AERO address as the source network layer address. The Server will update its neighbor cache entry for the Client and return a simple RA message.

The Client maintains a neighbor cache entry for each of its Servers and each of its active target Clients. When the Client receives ND messages on the AERO interface it updates or creates neighbor cache entries, including link-layer address and QoS preferences.

When there is a NAT on the path, the Client must send periodic messages to keep NAT state alive. If no other periodic messaging service is available, the Client can send RS messages to receive RA replies from its Server(s).

A Client may be configured as a co-resident function on the same platform as a Server. In that case, no Client/Server ND messaging is required and the Client and Server operate as a single functional unit. The Client function can use its MNP(s) to number downstream-attached networks, which may connect very large numbers of nodes.

3.8. AERO Interface Neighbor Cache Maintenance

Each AERO interface maintains a conceptual neighbor cache that includes an entry for each neighbor it communicates with on the AERO link per [RFC4861]. AERO interface neighbor cache entries are said to be one of "permanent", "symmetric", "asymmetric" or "proxy".

Permanent neighbor cache entries are created through explicit administrative action; they have no timeout values and remain in place until explicitly deleted. AERO Relays maintain permanent neighbor cache entries for their associated Relays, Servers, Gateways and Proxys, and AERO Servers and Proxys maintain permanent neighbor cache entries for their associated Relays. Each entry maintains the mapping between the neighbor's network-layer AERO address and corresponding INET address.

Symmetric neighbor cache entries are created and maintained through ND/PD exchanges as specified in Section 3.15, and remain in place for durations bounded by ND/PD lifetimes. AERO Servers maintain symmetric neighbor cache entries for each of their associated Clients, and AERO Clients maintain symmetric neighbor cache entries for each of their associated Servers.

Asymmetric neighbor cache entries are created or updated based on route optimization messaging as specified in Section 3.17, and are garbage-collected when keepalive timers expire. AERO route

optimization sources (ROSs) maintain asymmetric neighbor cache entries for each of their active target Clients with lifetimes based on ND messaging constants. Asymmetric neighbor cache entries are unidirectional since only the ROS (i.e., and not the route optimization responder (ROR)) creates an entry.

Proxy neighbor cache entries are created and maintained by AERO Proxies when they process Client/Server ND/PD exchanges, and remain in place for durations bounded by ND/PD lifetimes. AERO Proxies maintain proxy neighbor cache entries for each of their associated Clients. Proxy neighbor cache entries track the Client state and the state of each of the Client's associated Servers.

To the list of neighbor cache entry states in Section 7.3.2 of [RFC4861], AERO interfaces add an additional state DEPARTED that applies to symmetric and proxy neighbor cache entries for Clients that have recently departed. The interface sets a "DepartTime" variable for the neighbor cache entry to "DEPARTTIME" seconds. DepartTime is decremented unless a new ND message causes the state to return to REACHABLE. While a neighbor cache entry is in the DEPARTED state, packets destined to the target Client are forwarded to the Client's new location instead of being dropped. When DepartTime decrements to 0, the neighbor cache entry is deleted. It is RECOMMENDED that DEPARTTIME be set to the default constant value 40 seconds to allow for packets in flight to be delivered while stale route optimization state may be present.

When a target AERO Server (acting as a Mobility Anchor Point (MAP)) receives a valid NS message used for route optimization, it searches for a symmetric neighbor cache entry for the target Client. The Server then acts as an ROR and returns a solicited NA message without creating a neighbor cache entry for the ROS, but creates a target Client "Report List" entry for the ROS and sets a "ReportTime" variable for the entry to REPORTTIME seconds. The ROR resets ReportTime when it receives a new authentic NS message, and otherwise decrements ReportTime while no NS messages have been received. It is RECOMMENDED that REPORTTIME be set to the default constant value 40 seconds to allow a 10 second window so that route optimization can converge before ReportTime decrements below REACHABLETIME.

When the ROS receives a solicited NA message response to its NS message, it creates or updates an asymmetric neighbor cache entry for the target network-layer and link-layer addresses. The ROS then (re)sets ReachableTime for the neighbor cache entry to REACHABLETIME seconds and uses this value to determine whether packets can be forwarded directly to the target, i.e., instead of via a default route. The ROS otherwise decrements ReachableTime while no further solicited NA messages arrive. It is RECOMMENDED that REACHABLETIME

be set to the default constant value 30 seconds as specified in [RFC4861].

The ROS also uses the value MAX_UNICAST_SOLICIT to limit the number of NS keepalives sent when a correspondent may have gone unreachable, the value MAX_RTR_SOLICITATIONS to limit the number of RS messages sent without receiving an RA and the value MAX_NEIGHBOR_ADVERTISEMENT to limit the number of unsolicited NAs that can be sent based on a single event. It is RECOMMENDED that MAX_UNICAST_SOLICIT, MAX_RTR_SOLICITATIONS and MAX_NEIGHBOR_ADVERTISEMENT be set to 3 the same as specified in [RFC4861].

Different values for DEPARTTIME, REPORTTIME, REACHABLETIME, MAX_UNICAST_SOLICIT, MAX_RTR_SOLCITATIONS and MAX_NEIGHBOR_ADVERTISEMENT MAY be administratively set; however, if different values are chosen, all nodes on the link MUST consistently configure the same values. Most importantly, DEPARTTIME and REPORTTIME SHOULD be set to a value that is sufficiently longer than REACHABLETIME to avoid packet loss due to stale route optimization state.

3.9. AERO Interface Forwarding Algorithm

IP packets enter a node's AERO interface either from the network layer (i.e., from a local application or the IP forwarding system) or from the link layer (i.e., from an AERO interface neighbor). Packets that enter the AERO interface from the network layer are encapsulated and forwarded into the AERO link, i.e., they are tunneled to an AERO interface neighbor. Packets that enter the AERO interface from the link layer are either re-admitted into the AERO link or forwarded to the network layer where they are subject to either local delivery or IP forwarding. In all cases, the AERO interface itself MUST NOT decrement the network layer TTL/Hop-count since its forwarding actions occur below the network layer.

AERO interfaces may have multiple underlying ANET/INET interfaces and/or neighbor cache entries for neighbors with multiple Interface ID registrations (see Section 3.6). The AERO interface uses each packet's DSCP value (and/or port number) to select an outgoing ANET/INET interface based on the node's own QoS preferences, and also to select a destination link-layer address based on the neighbor's ANET/INET interface with the highest preference. AERO implementations SHOULD allow for QoS preference values to be modified at runtime through network management.

If multiple outgoing interfaces and/or neighbor interfaces have a preference of "high", the AERO node replicates the packet and sends one copy via each of the (outgoing / neighbor) interface pairs;

otherwise, the node sends a single copy of the packet via the interface with the highest preference. AERO nodes keep track of which ANET/INET interfaces are currently "reachable" or "unreachable", and only use "reachable" interfaces for forwarding purposes.

For control messages, the source node always encapsulates the message in SPAN/INET headers, and forwards the message into the SPAN (i.e., it forwards the message to a Relay). For data packets, if the neighboring node can only be reached via the SPAN (or, if it is not yet know that the neighboring node is within the local segment) the source node encapsulates packets in a SPAN/INET headers and forwards them into the SPAN. Otherwise, the source node encapsulates packets in only an INET header for transmission within the local segment.

The following sections discuss the AERO interface forwarding algorithms for Clients, Proxies, Servers and Relays. In the following discussion, a packet's destination address is said to "match" if it is a non-link-local address with a prefix covered by an MSP/MNP, or if it is an AERO address that embeds an MNP, or if it is the same as an administratively-provisioned AERO address.

3.9.1. Client Forwarding Algorithm

When an IP packet enters a Client's AERO interface from the network layer the Client searches for an asymmetric neighbor cache entry that matches the destination. If there is a match, the Client uses one or more "reachable" underlying ANET interfaces in the entry for packet forwarding. If there is no asymmetric neighbor cache entry, the Client instead forwards the packets to a Server.

When an IP packet enters a Client's AERO interface from the link-layer, if the destination matches one of the Client's MNPs or link-local addresses the Client decapsulates the packet (if necessary) and delivers it to the network layer. Otherwise, the Client drops the packet and MAY return a network-layer ICMP Destination Unreachable message subject to rate limiting (see: Section 3.14).

3.9.2. Proxy Forwarding Algorithm

For control messages originating from or destined to a Client, the Proxy intercepts the message and updates its proxy neighbor cache entry for the Client. The Proxy then forwards a (proxyed) copy of the control message.

When the Proxy receives a data packet from a Client within the ANET, the Proxy searches for an asymmetric neighbor cache entry that matches the network-layer destination. If there is a match, the

Proxy uses one or more "reachable" neighbor interfaces in the entry for packet forwarding. Otherwise, the Proxy uses the SPAN/INET address in a permanent neighbor cache entry for a Relay (selected through longest-prefix match) as the encapsulation addresses and forwards the packet into the SPAN.

When the Proxy receives an encapsulated data packet from the INET, it searches for a proxy neighbor cache entry that matches the destination. If there is a proxy neighbor cache entry in the REACHABLE state, the Proxy forwards the packet to the Client; if the neighbor cache entry is in the DEPARTED state, the Proxy instead forwards the packet to the Client's Server and may return an unsolicited NA message as discussed in Section 3.19. If there is no neighbor cache entry, the Proxy discards the packet.

3.9.3. Server Forwarding Algorithm

When an IP packet enters a Server's AERO interface from either the network or link-layer, it decapsulates the packet (if the packet arrived from the link-layer) then processes the packet according to the network-layer destination address as follows:

- if the destination matches one of the Server's own addresses the Server forwards it to the network layer for local delivery.

- else, if the destination matches a symmetric neighbor cache entry the Server forwards the packet according to the neighbor cache state and link-layer address information. If the neighbor cache entry is in the REACHABLE state, the Server forwards the packet according to the cached link-layer information. If the neighbor cache entry is in the DEPARTED state, the Server instead continues to forward packets to the Client's new Server as discussed in Section 3.19. If the packet is destined to the same Client from which it arrived, however, the Server must forward the packet via a different "reachable" Interface ID than the one the packet arrived on. If there are no "reachable" Interface IDs, the Server must drop the packet.

- else, if the destination matches an asymmetric neighbor cache entry for a target Client, the Server forwards the packet according to the cached link-layer information.

- else, the Server uses the SPAN/INET address in a permanent neighbor cache entry for a Relay (selected through longest-prefix match) as the encapsulation addresses.

3.9.4. Gateway Forwarding Algorithm

Gateways perform the same forwarding procedures as for Servers, but also forward packets between the AERO interface and any downstream-attached INET interfaces. In particular, if the destination address of a packet that arrives on an AERO interfaces matches a prefix associated with a downstream-attached INET interface, the Gateway forwards the packet to the next hop via the INET interface. Conversely, the Gateway forwards packets that arrive on an INET interface to the next hop via the AERO interface or another INET interface according to longest prefix match.

3.9.5. Relay Forwarding Algorithm

Relays forward packets the same as any IP router. When the Relay receives an encapsulated packet via the AERO link, it removes the INET header and searches for a forwarding table entry that matches the destination address in the SPAN header. When the Relay receives an unencapsulated packet from a node outside the AERO link, it searches for a forwarding table entry that matches the IP destination address. The Relay then processes the packet as follows:

- if the destination does not match an MSP or the SSP, or if the destination matches one of the Relay's own addresses, the Relay submits the packet for either IP forwarding or local delivery.

- else, if the destination matches an MNP/SPP entry in the IP forwarding table the Relay encapsulates the packet in an INET header and forwards it to the neighbor.

- else, the Relay drops the packet and returns an ICMP Destination Unreachable message subject to rate limiting (see: Section 3.14).

As for any IP router, the Relay decrements the TTL/Hop Count when it forwards the packet.

3.10. AERO Interface Encapsulation and Re-encapsulation

AERO interfaces encapsulate packets in ANET/INET headers according to whether they are entering the AERO interface from the network layer or if they are being re-admitted into the same AERO link they arrived on. This latter form of encapsulation is known as "re-encapsulation". Note that Clients can avoid encapsulation when the first-hop access router is AERO-aware.

The AERO interface encapsulates the packet in an ANET/INET header per the Generic UDP Encapsulation (GUE) procedures in [I-D.ietf-intarea-gue][I-D.ietf-intarea-gue-extensions], or through

an alternate encapsulation format (e.g., see: Appendix A, [RFC2784], [RFC8086], [RFC4301], etc.).

For packets entering the AERO interface from the network layer, the AERO interface copies the "TTL/Hop Limit", "Type of Service/Traffic Class" [RFC2983], "Flow Label"[RFC6438] (for IPv6) and "Congestion Experienced" [RFC3168] values in the packet's IP header into the corresponding fields in the encapsulation header(s). For packets undergoing re-encapsulation, the AERO interface instead copies these values from the original encapsulation header into the new encapsulation header, i.e., the values are transferred between encapsulation headers and *not* copied from the encapsulated packet's network-layer header. (Note especially that by copying the TTL/Hop Limit between encapsulation headers the value will eventually decrement to 0 if there is a (temporary) routing loop.) For IPv4 encapsulation/re-encapsulation, the AERO interface sets the DF bit as discussed in Section 3.13.

When GUE encapsulation is used, the AERO interface next sets the UDP source port to a constant value that it will use in each successive packet it sends, and sets the UDP length field to the length of the encapsulated packet plus 8 bytes for the UDP header itself plus the length of the GUE header (or 0 if GUE direct IP encapsulation is used). For packets sent to a Server or Relay, the AERO interface sets the UDP destination port to 8060, i.e., the IANA-registered port number for AERO. For packets sent to a Client, the AERO interface sets the UDP destination port to the port value stored in the neighbor cache entry for this Client. The AERO interface then either includes or omits the UDP checksum according to the GUE specification.

3.11. AERO Interface Decapsulation

AERO interfaces decapsulate packets destined either to the AERO node itself or to a destination reached via an interface other than the AERO interface the packet was received on. Decapsulation is per the procedures specified for the appropriate encapsulation format.

3.12. AERO Interface Data Origin Authentication

AERO nodes employ simple data origin authentication procedures for encapsulated packets they receive from other nodes on the AERO link. In particular:

- AERO Relays and Servers accept encapsulated packets with a link-layer source address that matches a permanent neighbor cache entry.

- AERO Servers accept authentic encapsulated ND messages from Clients (either directly or via a Proxy), and create or update a symmetric neighbor cache entry for the Client based on the specific message type.

- AERO Clients and Servers accept encapsulated packets if there is a symmetric neighbor cache entry with a link-layer address that matches the packet's link-layer source address.

- AERO Proxies accept encapsulated packets if there is a proxy neighbor cache entry that matches the packet's network-layer address.

Each packet should include a signature that the recipient can use to authenticate the message origin, e.g., as for common VPN systems such as OpenVPN [OVPN]. In some environments, however, it may be sufficient to require signatures only for ND control plane messages (see: Section 14) and omit signatures for data plane messages.

3.13. AERO Interface Packet Size Issues

The AERO interface is the node's attachment to the AERO link. The AERO interface acts as a tunnel ingress when it sends a packet to an AERO link neighbor and as a tunnel egress when it receives a packet from an AERO link neighbor. AERO interfaces observe the packet sizing considerations for tunnels discussed in [I-D.ietf-intarea-tunnels] and as specified below.

The Internet Protocol expects that IP packets will either be delivered to the destination or a suitable Packet Too Big (PTB) message returned to support the process known as IP Path MTU Discovery (PMTUD) [RFC1191][RFC8201]. However, PTB messages may be crafted for malicious purposes such as denial of service, or lost in the network [RFC2923]. This can be especially problematic for tunnels, where a condition known as a PMTUD "black hole" can result. For these reasons, AERO interfaces employ operational procedures that avoid interactions with PMTUD, including the use of fragmentation when necessary.

AERO interfaces observe two different types of fragmentation. Source fragmentation occurs when the AERO interface (acting as a tunnel ingress) fragments the encapsulated packet into multiple fragments before admitting each fragment into the tunnel. Network fragmentation occurs when an encapsulated packet admitted into the tunnel by the ingress is fragmented by an IPv4 router on the path to the egress. Note that an IPv4 packet that incurs source fragmentation may also incur network fragmentation.

IPv6 specifies a minimum link Maximum Transmission Unit (MTU) of 1280 bytes [RFC8200]. Although IPv4 specifies a smaller minimum link MTU of 68 bytes [RFC0791], AERO interfaces also observe the IPv6 minimum for IPv4 even if encapsulated packets may incur network fragmentation.

IPv6 specifies a minimum Maximum Reassembly Unit (MRU) of 1500 bytes [RFC8200], while the minimum MRU for IPv4 is only 576 bytes [RFC1122] (but, note that many standard IPv6 over IPv4 tunnel types already assume a larger MRU than the IPv4 minimum).

AERO interfaces therefore configure an MTU that MUST NOT be smaller than 1280 bytes, MUST NOT be larger than the minimum MRU among all nodes on the AERO link minus the encapsulation overhead ("ENCAPS"), and SHOULD NOT be smaller than 1500 bytes. AERO interfaces also configure a Maximum Segment Unit (MSU) as the maximum-sized encapsulated packet that the ingress can inject into the tunnel without source fragmentation. The MSU value MUST NOT be larger than (MTU+ENCAPS) and MUST NOT be larger than 1280 bytes unless there is operational assurance that a larger size can traverse the link along all paths.

All AERO nodes MUST configure the same MTU value for reasons cited in [RFC3819][RFC4861]; in particular, multicast support requires a common MTU value among all nodes on the link. All AERO nodes MUST configure an MRU large enough to reassemble packets up to (MTU+ENCAPS) bytes in length; nodes that cannot configure a large-enough MRU MUST NOT enable an AERO interface.

The network layer proceeds as follow when it presents an IP packet to the AERO interface. For each IPv4 packet that is larger than the AERO interface MTU and with the DF bit set to 0, the network layer uses IPv4 fragmentation to break the packet into a minimum number of non-overlapping fragments where the first fragment is no larger than the MTU and the remaining fragments are no larger than the first. For all other IP packets, if the packet is larger than the AERO interface MTU, the network layer drops the packet and returns a PTB message to the original source. Otherwise, the network layer admits each IP packet or fragment into the AERO interface.

For each IP packet admitted into the AERO interface, the interface (acting as a tunnel ingress) encapsulates the packet. If the encapsulated packet is larger than the AERO interface MSU the ingress source-fragments the encapsulated packet into a minimum number of non-overlapping fragments where the first fragment is no larger than the MSU and the remaining fragments are no larger than the first. The ingress then admits each encapsulated packet or fragment into the tunnel, and for IPv4 sets the DF bit to 0 in the IP encapsulation

header in case any network fragmentation is necessary. The encapsulated packets will be delivered to the egress, which reassembles them into a whole packet if necessary.

Several factors must be considered when fragmentation is needed. For AERO links over IPv4, the IP ID field is only 16 bits in length, meaning that fragmentation at high data rates could result in data corruption due to reassembly misassociations [RFC6864][RFC4963]. In environments where IP fragmentation issues could result in operational problems, the ingress SHOULD employ intermediate-layer source fragmentation (see: [RFC2764] and [I-D.ietf-intarea-gue-extensions]) before appending the outer encapsulation headers to each fragment. Since the encapsulation fragment header reduces the room available for packet data, but the original source has no way to control its insertion, the ingress MUST include the fragment header length in the ENCAPS length even for packets in which the header is absent.

3.14. AERO Interface Error Handling

When an AERO node admits encapsulated packets into the AERO interface, it may receive link-layer or network-layer error indications.

A link-layer error indication is an ICMP error message generated by a router in the INET on the path to the neighbor or by the neighbor itself. The message includes an IP header with the address of the node that generated the error as the source address and with the link-layer address of the AERO node as the destination address.

The IP header is followed by an ICMP header that includes an error Type, Code and Checksum. Valid type values include "Destination Unreachable", "Time Exceeded" and "Parameter Problem" [RFC0792][RFC4443]. (AERO interfaces ignore all link-layer IPv4 "Fragmentation Needed" and IPv6 "Packet Too Big" messages since they only emit packets that are guaranteed to be no larger than the IP minimum link MTU as discussed in Section 3.13.)

The ICMP header is followed by the leading portion of the packet that generated the error, also known as the "packet-in-error". For ICMPv6, [RFC4443] specifies that the packet-in-error includes: "As much of invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU" (i.e., no more than 1280 bytes). For ICMPv4, [RFC0792] specifies that the packet-in-error includes: "Internet Header + 64 bits of Original Data Datagram", however [RFC1812] Section 4.3.2.3 updates this specification by stating: "the ICMP datagram SHOULD contain as much of the original datagram as

possible without the length of the ICMP datagram exceeding 576 bytes".

The link-layer error message format is shown in Figure 5 (where, "L2" and "L3" refer to link-layer and network-layer, respectively):

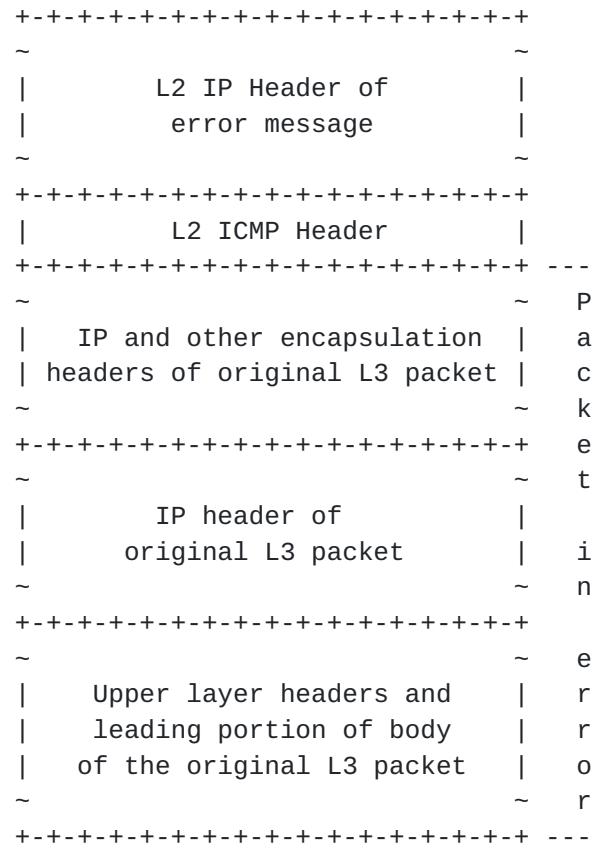

Figure 5: AERO Interface Link-Layer Error Message Format

The AERO node rules for processing these link-layer error messages are as follows:

- When an AERO node receives a link-layer Parameter Problem message, it processes the message the same as described as for ordinary ICMP errors in the normative references [RFC0792][RFC4443].

- When an AERO node receives persistent link-layer Time Exceeded messages, the IP ID field may be wrapping before earlier fragments awaiting reassembly have been processed. In that case, the node SHOULD begin including integrity checks and/or institute rate limits for subsequent packets.

- When an AERO node receives persistent link-layer Destination Unreachable messages in response to encapsulated packets that it

sends to one of its asymmetric neighbor correspondents, the node SHOULD process the message as an indication that a path may be failing, and MAY initiate NUD over that path. If it receives Destination Unreachable messages on many or all paths, the node SHOULD set ReachableTime for the corresponding asymmetric neighbor cache entry to 0 and allow future packets destined to the correspondent to flow through a default route.

- When an AERO Client receives persistent link-layer Destination Unreachable messages in response to encapsulated packets that it sends to one of its symmetric neighbor Servers, the Client SHOULD mark the path as unusable and use another path. If it receives Destination Unreachable messages on many or all paths, the Client SHOULD associate with a new Server and release its association with the old Server as specified in Section 3.19.7.

- When an AERO Server receives persistent link-layer Destination Unreachable messages in response to encapsulated packets that it sends to one of its symmetric neighbor Clients, the Server SHOULD mark the underlying path as unusable and use another underlying path. If it receives Destination Unreachable messages on multiple paths, the Server should take no further actions unless it receives an explicit ND/PD release message or if the PD lifetime expires. In that case, the Server MUST release the Client's delegated MNP, withdraw the MNP from the AERO routing system and delete the neighbor cache entry.

- When an AERO Relay or Server receives link-layer Destination Unreachable messages in response to an encapsulated packet that it sends to one of its permanent neighbors, it treats the messages as an indication that the path to the neighbor may be failing. However, the dynamic routing protocol should soon reconverge and correct the temporary outage.

When an AERO Relay receives a packet for which the network-layer destination address is covered by an MSP, if there is no more-specific routing information for the destination the Relay drops the packet and returns a network-layer Destination Unreachable message subject to rate limiting. The Relay writes the network-layer source address of the original packet as the destination address and uses one of its non link-local addresses as the source address of the message.

When an AERO node receives an encapsulated packet for which the reassembly buffer it too small, it drops the packet and returns a network-layer Packet Too Big (PTB) message. The node first writes the MRU value into the PTB message MTU field, writes the network-layer source address of the original packet as the destination

address and writes one of its non link-local addresses as the source address.

3.15. AERO Router Discovery, Prefix Delegation and Autoconfiguration

AERO Router Discovery, Prefix Delegation and Autoconfiguration are coordinated as discussed in the following Sections.

3.15.1. AERO ND/PD Service Model

Each AERO Server on the link configures a PD service to facilitate Client requests. Each Server is provisioned with a database of MNP-to-Client ID mappings for all Clients enrolled in the AERO service, as well as any information necessary to authenticate each Client. The Client database is maintained by a central administrative authority for the AERO link and securely distributed to all Servers, e.g., via the Lightweight Directory Access Protocol (LDAP) [RFC4511], via static configuration, etc. Therefore, no Server-to-Server PD state synchronization is necessary, and Clients can optionally hold separate PDs for the same MNPs from multiple Servers. Clients can receive new PDs from new Servers before releasing PDs received from existing Servers for service continuity. Clients receive the same service regardless of the Servers they select, although selecting Servers that are topologically nearby may provide better routing.

AERO Clients and Servers use ND messages to maintain neighbor cache entries. AERO Servers configure their AERO interfaces as advertising interfaces, and therefore send unicast RA messages with configuration information in response to a Client's RS message. Thereafter, Clients send additional RS messages to refresh prefix and/or router lifetimes.

AERO Clients and Servers include PD parameters in RS/RA messages to be used for Prefix Delegation (see [I-D.templin-6man-dhcpv6-ndopt] for ND/PD alternatives). The unified ND/PD messages are exchanged between Client and Server according to the prefix management schedule required by the PD service. If the Client knows its MNP in advance, it can include its AERO address as the source address of an RS message and with an SLLAO with a valid Prefix Length for the MNP. If the Server (and Proxy) accept the Client's MNP assertion, they inject the prefix into the routing system and establish the necessary neighbor cache state.

The following sections specify the Client and Server behavior.

3.15.2. AERO Client Behavior

AERO Clients can discover the INET and AERO addresses of AERO Servers in the MAP list via static configuration (e.g., from a flat-file map of Server addresses and locations), or through an automated means such as Domain Name System (DNS) name resolution [RFC1035]. In the absence of other information, the Client can resolve the DNS Fully-Qualified Domain Name (FQDN) "linkupnetworks.[domainname]" where "linkupnetworks" is a constant text string and "[domainname]" is a DNS suffix for the Client's ANET interface (e.g., "example.com"). Alternatively, the Client can discover the Server's address through a multicast RS as described below.

To associate with a Server, the Client acts as a requesting router to request MNPs. The Client prepares an RS message with PD parameters (e.g., with an SLLAO with non-zero Prefix Length). If the Client already knows the Server's AERO address, it includes the AERO address as the network-layer destination address; otherwise, it includes all-routers multicast (ff02::2) as the network-layer destination address. If the Client already knows its own AERO address, it uses the AERO address as the network-layer source address; otherwise, it uses the unspecified AERO address (fe80::ffff:ffff) as the network-layer source address.

The Client next includes an SLLAO in the RS message formatted as described in Section 3.6 to register its link-layer information with the Server. The SLLAO corresponding to the ANET interface over which the Client will send the RS message MUST set the S bit to 1. The Client MAY include additional SLLAOs specific to other underlying interfaces, but if so it MUST set their S, Port Number and Link Layer Address fields to 0. If the Client is connected to an ANET for which encapsulation is required, the Client finally encapsulates the RS message in an ANET header with its own ANET address as the source address and the INET address of the Server as the destination.

The Client then sends the RS message (either via a VPN for VPNed interfaces, via a Proxy for proxyed interfaces or via the SPAN for native interfaces) and waits for an RA message reply (see Section 3.15.3) while retrying up to MAX_RTR_SOLICITATIONS times until an RA is received. If the Client receives no RAs, or if it receives an RA with Router Lifetime set to 0, the Client SHOULD abandon this Server and try another Server. Otherwise, the Client processes the PD information found in the RA message.

Next, the Client creates a symmetric neighbor cache entry with the Server's AERO address as the network-layer address and the address in the first SLLAO as the Server's INET address. The Client records the RA Router Lifetime field value in the neighbor cache entry as the

time for which the Server has committed to maintaining the MNP in the routing system. The Client then autoconfigures AERO addresses for each of the delegated MNPs and assigns them to the AERO interface. The Client also caches any MSPs included in Route Information Options (RIOs) [RFC4191] as MSPs to associate with the AERO link, and assigns the MTU value in the MTU option to its AERO interface while configuring an appropriate MRU.

The Client then registers additional ANET interfaces with the Server by sending additional RS messages including SLLAOs via other ANET interfaces after the initial RS/RA exchange. The Client sends the RS messages to the Server's AERO address but omits PD parameters since the initial RS/RA exchange has already established PD state.

The Client examines the X and N bits in the first SLLAO of each RA message it receives. If the X bit value is 1 the Client infers that there is a Proxy on the path, and if the N bit value is 1 the Client infers that there is a NAT on the path. If N is '1', the Client SHOULD set Port Number and Link-Layer Address to 0 in the first S/TLLAO of any subsequent ND messages it sends to the Server over that link.

Following autoconfiguration, the Client sub-delegates the MNPs to its attached EUNs and/or the Client's own internal virtual interfaces as described in [I-D.templin-v6ops-pdhost] to support the Client's downstream attached "Internet of Things (IoT)". The Client subsequently maintains its MNP delegations through each of its Servers by sending additional RS messages with PD parameters before Router Lifetime expires.

After the Client registers its ANET interfaces, it may wish to change one or more registrations, e.g., if an ANET interface changes address or becomes unavailable, if QoS preferences change, etc. To do so, the Client prepares an RS message to send over any available ANET interface. The RS MUST include an SLLAO specific to the selected ANET interface as the first SLLAO and MAY include any additional SLLAOs specific to other ANET interfaces. The Client includes fresh 'P(i)' values in each SLLAO to update the Server's neighbor cache entry. If the Client wishes to update only the 'P(i)' values, it sets the Port Number and Link-Layer Address fields to 0. If the Client wishes to disable the underlying interface, it sets the D bit to 1. When the Client receives the Server's RA response, it has assurance that the Server has been updated with the new information.

If the Client wishes to associate with multiple Servers, it repeats the same procedures above for each additional Server. If the Client wishes to discontinue use of a Server it issues an RS message over any underlying interface with the R bit set to 1 in the first SLLAO.

When the Server processes the message, it releases the MNP, sets the symmetric neighbor cache entry state for the Client to DEPARTED, withdraws the IP route from the routing system and returns an RA reply with Router Lifetime set to 0.

3.15.3. AERO Server Behavior

AERO Servers act as IPv6 routers and support a PD service for Clients. AERO Servers arrange to add their AERO and INET addresses to a static map of Server addresses for the link and/or the DNS resource records for the FQDN "linkupnetworks.[domainname]" before entering service. The list of Server addresses should be geographically and/or topologically referenced, and forms the MAP list for the AERO link.

When an AERO Server receives a prospective Client's RS message with PD parameters on its AERO interface, it SHOULD return an immediate RA reply with Router Lifetime set to 0 if it is currently too busy or otherwise unable to service the Client. Otherwise, the Server authenticates the RS message and processes the PD parameters. The Server first determines the correct MNPs to delegate to the Client by searching the Client database. When the Server delegates the MNPs, it also creates an IP forwarding table entry for each MNP so that the MNPs are propagated into the routing system (see: Section 3.3). For IPv6, the Server creates a single IPv6 forwarding table entry for each MNP. For IPv4, the Server creates both an IPv4 forwarding table entry and an IPv6 forwarding table entry with the IPv4-mapped IPv6 address corresponding to the IPv4 address.

The Server next creates a symmetric neighbor cache entry for the Client using the base AERO address as the network-layer address and with lifetime set to no more than the smallest PD lifetime. Next, the Server updates the neighbor cache entry by recording the information in each SLLAO in the RS indexed by the Interface ID and including the Port Number, Link Layer Address and P(i) values. For the SLLAO with S set to 1, however, the Server records the actual INET header source addresses instead of those that appear in the SLLAO in case there was a NAT in the path. The Server also records the value of the X bit to indicate whether there is a Proxy on the path.

Next, the Server prepares an RA message using its AERO address as the network-layer source address and the network-layer source address of the RS message as the network-layer destination address. The Server includes the delegated MNPs, any other PD parameters and an SLLAO with the Link Layer Address set to the Server's SPAN address and with Interface ID set to 0xffff. The Server then includes one or more RIOs that encode the MSPs for the AERO link, plus an MTU option for

the link MTU (see Section 3.13). The Server finally encapsulates the message in a SPAN header with source address set to its own SPAN address and destination address set to the Client's (or Proxy's) SPAN address, then forwards the message into the SPAN.

After the initial RS/RA exchange, the AERO Server maintains the symmetric neighbor cache entry for the Client. If the Client (or Proxy) issues additional NS/RS messages, the Server resets ReachableTime. If the Client (or Proxy) issues an RS with PD release parameters (e.g., by including an SLLAO with R set to 1), or if the Client becomes unreachable, the Server sets the Client's symmetric neighbor cache entry to the DEPARTED state and withdraws the IP routes from the AERO routing system.

The Server processes these and any other Client ND/PD messages, and returns an NA/RA reply. The Server may also issue an unsolicited RA message with PD reconfigure parameters to cause the Client to renegotiate its PDs, and may issue an unsolicited RA message with Router Lifetime set to 0 if it can no longer service this Client. Finally, If the symmetric neighbor cache entry is in the DEPARTED state, the Server deletes the entry after DepartTime expires.

3.15.3.1. Lightweight DHCPv6 Relay Agent (LDRA)

When DHCPv6 is used as the ND/PD service back end, AERO Clients and Servers are always on the same link (i.e., the AERO link) from the perspective of DHCPv6. However, in some implementations the DHCPv6 server and ND function may be located in separate modules. In that case, the Server's AERO interface module can act as a Lightweight DHCPv6 Relay Agent (LDRA)[RFC6221] to relay PD messages to and from the DHCPv6 server module.

When the LDRA receives an authentic RS message, it extracts the PD message parameters and uses them to construct an IPv6/UDP/DHCPv6 message. It sets the IPv6 source address to the source address of the RS message, sets the IPv6 destination address to 'All_DHCP_Relay_Agents_and_Servers' and sets the UDP fields to values that will be understood by the DHCPv6 server.

The LDRA then wraps the message in a DHCPv6 'Relay-Forward' message header and includes an 'Interface-Id' option that includes enough information to allow the LDRA to forward the resulting Reply message back to the Client (e.g., the Client's link-layer addresses, a security association identifier, etc.). The LDRA also wraps the information in all of the SLLAOs from the RS message into the Interface-Id option, then forwards the message to the DHCPv6 server.

When the DHCPv6 server prepares a Reply message, it wraps the message in a 'Relay-Reply' message and echoes the Interface-Id option. The DHCPv6 server then delivers the Relay-Reply message to the LDRA, which discards the Relay-Reply wrapper and IPv6/UDP headers, then uses the DHCPv6 message to construct an RA response to the Client. The Server uses the information in the Interface-Id option to prepare the RA message and to cache the link-layer addresses taken from the SLLAOs echoed in the Interface-Id option.

3.16. The AERO Proxy

Clients may connect to ANETs that do not support direct communications to Servers in outside INETs. In that case, the ANET can employ an AERO Proxy. The Proxy is located at the ANET/INET border and listens for encapsulated RS messages originating from or RA messages destined to ANET Clients. The Proxy acts on these control messages as follows:

- when the Proxy receives an RS message from a new ANET Client, it first authenticates the message then examines the RS message network-layer destination address. If the destination address is a Server's AERO address, the Proxy proceeds to the next step. Otherwise, if the destination is all-routers multicast the Proxy selects a "nearby" Server that is likely to be a good candidate to serve the Client and replaces the RS destination address with the Server's AERO address. Next, the Proxy creates a proxy neighbor cache entry and caches the Client and Server addresses along with any identifying information including Transaction IDs, Client Identifiers, Nonce values, etc. The Proxy then examines the address in the RS message SLLAO with S set to 1. If the address is different than the Client's ANET address, the Proxy notes that the Client is behind a NAT. The Proxy then sets the X flag in the SLLAO to 1 and changes the address in the SLLAO to its own SPAN address. The Proxy finally re-encapsulates the RS message in a SPAN header using its own SPAN address as the source address and the SPAN address of the Server as the destination address, then forwards the message to the Server via the SPAN.

- when the Server receives the RS message, it authenticates the message then creates or updates a symmetric neighbor cache entry for the Client with the Proxy's SPAN address as the link-layer address. The Server then sends an RA message with a single SLLAO back to the Proxy via the SPAN.

- when the Proxy receives the RA message, it matches the message with the RS that created the proxy neighbor cache entry. The Proxy then caches the route information in the message as a mapping from the Client's MNPs to the Client's ANET address, and

sets the neighbor cache entry state to REACHABLE. The Proxy then changes the Link Layer Address in the SLLAO to its own ANET address, re-encapsulates the RA message in an ANET header, sets the X flag in the SLLAO to 1, sets the N flag in the SLLAO to 1 if the Client is behind a NAT, and forwards the message to the Client.

After the initial RS/RA exchange, the Proxy forwards any Client data packets for which there is no matching asymmetric neighbor cache entry to a Relay via the SPAN. Finally, the Proxy forwards any Client data destined to an asymmetric neighbor cache target directly to the target according to the link-layer information - the process of establishing asymmetric neighbor cache entries is specified in Section 3.17.

While the Client is still attached to the ANET, the Proxy continues to send NS/RS messages to update each Server's symmetric neighbor cache entries on behalf of the Client and/or to convey QoS updates. If the Server ceases to send solicited NA/RA responses, the Proxy marks the Server as unreachable and sends an unsolicited RA with Router Lifetime set to zero to inform the Client that this Server is no longer able to provide Service. If the Client becomes unreachable, the Proxy sets the neighbor cache entry state to DEPARTED and sends an RS message to each Server with an SLLAO with D set to 1 and with Interface ID set to the Client's interface ID so that the Server will de-register this Interface ID. Although the Proxy engages in these ND exchanges on behalf of the Client, the Client can also send ND messages on its own behalf, e.g., if it is in a better position than the Proxy to convey QoS changes, etc.

In some ANETs that employ a Proxy, the Client's MNP can be injected into the ANET routing system. In that case, the Client can send data messages without encapsulation so that the ANET native routing system transports the unencapsulated packets to the Proxy. This can be very beneficial, e.g., if the Client connects to the ANET via low-end data links such as some aviation wireless links. This encapsulation avoidance represents a form of "header compression", meaning that the MTU should be sized based on the size of full encapsulated messages even if most messages are sent unencapsulated.

If the first-hop ANET access router is AERO-aware, the Client can avoid encapsulation for both its control and data messages. When the Client connects to the link, it can send an unencapsulated RS message with source address set to its AERO address and with destination address set to the AERO address of the Client's selected Server or to all-routers multicast. The Client includes an SLLAO with Interface ID, Prefix Length and P(i) information but with Port Number and Link-Layer Address set to 0.

The Client then sends the unencapsulated RS message, which will be intercepted by the AERO-Aware access router. The access router then encapsulates the RS message in an ANET header with its own address as the source address and the address of a Proxy as the destination address. The access router further remembers the address of the Proxy so that it can encapsulate future data packets from the Client via the same Proxy. If the access router needs to change to a new Proxy, it simply sends another RS message toward the Server via the new Proxy on behalf of the Client.

In this arrangement, the only control messages sent by the Client are unencapsulated RS messages with its AERO address as the source address and the AERO address of the Server as the destination address. The Client will also receive unencapsulated RA messages from the Server via both the Proxy and access router.

In some cases, the access router and Proxy may be one and the same node. In that case, the node would be located on the same physical link as the Client, but its message exchanges with the Server would need to pass through a security gateway at the ANET/INET border. The method for deploying access routers and Proxys (i.e. as a single node or multiple nodes) is an ANET-local administrative consideration.

3.17. AERO Route Optimization

While data packets are flowing between a source and target node, route optimization SHOULD be used. Route optimization is initiated by the first eligible Route Optimization Source (ROS) closest to the source as follows:

- For Clients on VPNed, NATed and Direct interfaces, the Server is the ROS.
- For Clients on Proxyed interfaces, the Proxy is the ROS.
- For Clients on native interfaces, the Client itself is the ROS.
- For INET interfaces serviced by a Gateway, the Gateway is the ROS.

The route optimization procedure is conducted between the ROS and a Route Optimization Responder (ROR) in the same manner as for IPv6 ND Address Resolution, and using the same NS/NA messaging. The ROR is the Server (MAP) for MN targets, or the Gateway for FN targets. The procedures are specified in the following sections.

3.17.1. Route Optimization Initiation

While the data packets are flowing from the source CN toward a target CN, the ROS also sends an NS message to receive a solicited NA message from the ROR .

When the ROS sends an NS, it includes the AERO address of the ROS as the source address (e.g.,fe80::1) and the AERO address corresponding to the data packet's destination address as the destination address (e.g., if the destination address is 2001:db8:1:2::1 then the corresponding AERO address is fe80::2001:db8:1:2). The NS message includes no SLLAOs, but SHOULD include a Timestamp and Nonce option.

The ROS then encapsulates the message in a SPAN header with source set to its own SPAN address and destination set to the inner packet destination, then sends the message into the SPAN without decrementing the network-layer TTL/Hop Limit field.

3.17.2. Relaying the NS

When the Relay receives the (double-encapsulated) NS message from the ROS, it discards the outer IP header and determines that the ROR is the next hop by consulting its standard IP forwarding table for the SPAN header destination address. The Relay then forwards the SPAN message toward the ROR the same as for any IP router. The final-hop Relay in the SPAN will encapsulate the message in an INET header when it delivers the message to the ROR.

3.17.3. Processing the NS and Sending the NA

When the ROR receives the (double-encapsulated) NS message, it examines the AERO destination address to determine whether it is the aggregation point for the target CN; if not, it drops the NS message. Otherwise, if the target CN is serviced by a Client in the DEPARTED state the ROR changes the NS message SPAN destination address to the address of the Client's new Server, re-encapsulates the message in the appropriate SPAN/INET headers and forwards the message to new Server. If the target CN is serviced by a Client in the REACHABLE state the ROR adds the AERO source address to the target Client's Report List with time set to ReportTime.

For both Servers and Gateways, the ROR next prepares a solicited NA message to send back to the ROS but does not create a neighbor cache entry. The ROR sets the NA source address to its own AERO address and sets the destination address to the AERO address of the ROS. The NA message includes the Nonce value received in the NS, the current Timestamp, and a first TLLAO with Interface ID set to 0xffff, with all P(i) values set to "low", with Prefix Length set to the prefix

length of the target Client's MNP and with Link Layer Address set to the ROR's SPAN address.

The ROR next includes additional TLLAOs for all of the target Client's Interface IDs. For NATed, VPNed and Direct interfaces, the TLLAO Link Layer Addresses are the SPAN address of the ROR. For Proxyed interfaces, the TLLAO Link Layer Addresses are the SPAN addresses of the target Client's Proxies, and for native interfaces the TLLAO Link Layer Addresses are the SPAN addresses of the target Client.

The ROR finally encapsulates the NA message in a SPAN header with source set to its own SPAN address and destination set to the source SPAN address of the NS message, then sends the message into the SPAN without decrementing the network-layer TTL/Hop Limit field.

3.17.4. Relaying the NA

When the Relay receives the (double-encapsulated) NA message from the ROR, it discards the INET header and determines that the ROS is the next hop by consulting its standard IP forwarding table for the SPAN header destination address. The Relay then forwards the SPAN-encapsulated NA message toward the ROS the same as for any IP router. The final-hop Relay in the SPAN will encapsulate the message in an INET header when it delivers the message to the ROS.

3.17.5. Processing the NA

When the ROS receives the (double-encapsulated) solicited NA message, it discards the INET and SPAN headers. The ROS next verifies the Nonce and Timestamp values, then creates an asymmetric neighbor cache entry for the target Client or Gateway and caches all information found in the solicited NA TLLAOs. The ROS finally sets the asymmetric neighbor cache entry lifetime to ReachableTime seconds.

3.17.6. Route Optimization Maintenance

Following route optimization, the ROS forwards future data packets destined to one of the target's CNs via the addresses found in the cached link-layer information. The route optimization is shared by all sources that send packets to the target node via the ROS, i.e., and not just the source on behalf of which the route optimization was initiated.

While new data packets destined to one of the target's CNs are flowing through the ROS, it sends additional NS messages to the ROR before ReachableTime expires to receive a fresh solicited NA message the same as described in the previous sections. The ROS then updates

the asymmetric neighbor cache entry to refresh ReachableTime, while (for target Clients) the ROR adds or updates the ROS address to the target Client's Report List and with time set to ReportTime. While no data packets are flowing, the ROS instead allows ReachableTime for the asymmetric neighbor cache entry to expire. When ReachableTime expires, the ROS deletes the asymmetric neighbor cache entry. Future data packets flowing through the ROS will again trigger a new route optimization exchange while initial data packets travel over a suboptimal route via Servers and/or Relays.

The ROS may also receive unsolicited NA messages from the ROR at any time. If there is an asymmetric neighbor cache entry for the target, the ROS updates the link-layer information but does not update ReachableTime since the receipt of an unsolicited NA does not confirm that the forward path is still working. If there is no asymmetric neighbor cache entry, the route optimization source simply discards the unsolicited NA. Cases in which unsolicited NA messages are generated are specified in Section 3.19.

In this arrangement, the ROS holds an asymmetric neighbor cache entry for the ROR, but the ROR does not hold an asymmetric neighbor cache entry for the ROS. The route optimization neighbor relationship is therefore asymmetric and unidirectional. If the target node also has packets to send back to the source node, then a separate route optimization procedure is required in the reverse direction. But, there is no requirement that the forward and reverse paths be symmetric.

3.18. Neighbor Unreachability Detection (NUD)

AERO nodes perform Neighbor Unreachability Detection (NUD) as described in [RFC4861]. NUD is performed either reactively in response to persistent link-layer errors (see Section 3.14) or proactively to confirm bi-directional reachability. The NUD algorithm may further be seeded by ND hints of forward progress, but care must be taken to avoid inferring reachability based on spoofed information.

When an ROR directs an ROS to one or more target link-layer addresses, the ROS SHOULD proactively test the direct path to each address by sending an initial NS message to elicit a solicited NA response. While testing the path, the ROS can optionally continue sending packets via its default router, maintain a small queue of packets until target reachability is confirmed, or (optimistically) allow packets to flow directly to the target.

AERO nodes may have multiple link-layer addresses for the target neighbor. In that case, NUD SHOULD be performed over each address

individually, and the source node should only consider the neighbor unreachable if NUD fails over multiple underlying interface paths.

When a source node sends an NS message used for NUD, it uses its AERO addresses as the IPv6 source address and the AERO address corresponding to each target link-layer address as the destination. For each target link-layer address, if the address is not located within the same AERO link segment the source node encapsulates the NS message in a SPAN header with its own SPAN address as the source and the SPAN address of the target as the destination, then forwards the message into the SPAN. If the target address is located within the same segment, however, the source node omits the SPAN header and encapsulates the message in an INET header with is own INET address as the source and the INET address of the target as the destination, then sends the message directly to the target.

Paths that pass NUD tests are marked as "reachable", while those that do not are marked as "unreachable". These markings inform the AERO interface forwarding algorithm specified in Section 3.9.

Proxies can perform NUD to verify Server reachability on behalf of their proxyed Clients so that the Clients need not engage in NUD messaging themselves.

3.19. Mobility Management and Quality of Service (QoS)

AERO is a Distributed Mobility Management (DMM) service. Each Server is responsible for only a subset of the Clients on the AERO link, as opposed to a Centralized Mobility Management (CMM) service where there is a single network mobility service for all Clients. Clients coordinate with their associated Servers via RS/RA exchanges to maintain the DMM profile, and the AERO routing system tracks all current Client/Server peering relationships.

Servers provide a Mobility Anchor Point (MAP) for their dependent Clients. Clients are responsible for maintaining neighbor relationships with their Servers through periodic RS/RA exchanges, which also serves to confirm neighbor reachability. When a Client's underlying interface address and/or QoS information changes, the Client is responsible for updating the Server with this new information. Note that for Proxyed interfaces, however, the Proxy can perform the RS/RA exchanges on the Client's behalf.

Mobility management considerations are specified in the following sections.

3.19.1. Mobility Update Messaging

RORs (i.e., Servers acting as MAPs) accommodate mobility and/or QoS change events by sending an unsolicited NA message to each ROS in the target Client's Report List. When an ROR sends an unsolicited NA message, it sets the IPv6 source address to the Client's AERO address and sets the IPv6 destination address to all-nodes multicast (ff02::1). The ROR also includes a TLLAO with Interface ID 0xffff with Link Layer address set to the ROR's SPAN address, and includes additional TLLAOs for all of the target Client's Interface IDs with Link Layer Address set to the corresponding SPAN addresses. The ROR finally encapsulates the message in a SPAN header with source set to its own SPAN address and destination set to the SPAN address of the ROS, then sends the message into the SPAN.

As for the hot-swap of interface cards discussed in Section 7.2.6 of [RFC4861], the transmission and reception of unsolicited NA messages is unreliable but provides a useful optimization. In well-connected Internetworks with robust data links unsolicited NA messages will be delivered with high probability, but in any case the ROR can optionally send up to MAX_NEIGHBOR_ADVERTISEMENT unsolicited NAs to each ROS to increase the likelihood that at least one will be received.

When an ROS receives an unsolicited NA message, it ignores the message if there is no existing neighbor cache entry for the Client. Otherwise, it uses the included TLLAOs to update the address and QoS information in the neighbor cache entry, but does not reset ReachableTime since the receipt of an unsolicited NA message from the target Server does not provide confirmation that any forward paths to the target Client are working.

If unsolicited NA messages are lost, the ROS may be left with stale address and/or QoS information for the Client for up to ReachableTime seconds. During this time, the ROS can continue sending packets to the target Client according to its current neighbor cache information but may receive persistent unsolicited NA messages as discussed in Section 3.19.2.

3.19.2. Forwarding Packets on Behalf of Departed Clients

When a Server receives packets with destination addresses that match a symmetric neighbor cache entry in the DEPARTED state, it forwards the packets to the SPAN address corresponding to the Client's new Server. If the encapsulation source is in the Report List, the Server also sends an unsolicited NA message via the SPAN (subject to rate limiting) with a TLLAO with Interface ID 0xffff and with D set to 1. The ROS will then realize that it needs to set its asymmetric

neighbor cache entry state for the target to DEPARTED, and SHOULD re-initiate route optimization after a short delay.

When a Proxy receives packets with destination addresses that match a proxy neighbor cache entry in the DEPARTED state, it forwards the packets to one of the target Client's Servers. If the encapsulation source is not one of its proxy neighbor Clients, the Proxy also returns an unsolicited NA message via the SPAN (subject to rate limiting) with a single TLLAO with the target Client's Interface ID and with D set to 1. The source will then realize that it needs to mark its neighbor cache entry Interface ID for the Proxy as "unreachable", and SHOULD re-initiate route optimization while continuing to forward packets according to the remaining neighbor cache entry state.

When a Client receives packets with destination addresses that do not match one of its MNPs, it drops the packets silently.

3.19.3. Announcing Link-Layer Address and/or QoS Preference Changes

When a Client needs to change its ANET addresses and/or QoS preferences (e.g., due to a mobility event), either the Client or Proxy sends RS messages to its Servers via the SPAN with SLLAOs that include the new Client Port Number, Link Layer Address and P(i) values. If the RS messages are sent solely for the purpose of updating QoS preferences, S, Port Number and Link-Layer Address are set to 0. If the RS message is not sent for the purpose of asserting a PD, the Prefix Length is set to 0.

Up to MAX_RTR_SOLICITATION RS messages MAY be sent in parallel with sending actual data packets in case one or more RAs are lost. If all RAs are lost, the Client SHOULD re-associate with a new Server.

3.19.4. Bringing New Links Into Service

When a Client needs to bring new ANET interfaces into service (e.g., when it activates a new data link), it sends RS messages to its Servers via the ANET interface with SLLAOs that include the new Client Link Layer Address information. If the RS message is not sent for the purpose of asserting a PD, the Prefix Length is set to 0.

3.19.5. Removing Existing Links from Service

When a Client needs to remove existing ANET interfaces from service (e.g., when it de-activates an existing data link), it sends RS messages to its Servers with SLLAOs with the D flag set to 1.

If the Client needs to send RS messages over an ANET interface other than the one being removed from service, it MUST include a current SLLAO for the sending interface as the first SLLAO and include SLLAOs for any ANET interfaces being removed from service as additional SLLAOs.

3.19.6. Implicit Mobility Management

AERO interface neighbors MAY provide a configuration option that allows them to perform implicit mobility management in which no ND messaging is used. In that case, the Client only transmits packets over a single interface at a time, and the neighbor always observes packets arriving from the Client from the same link-layer source address.

If the Client's ANET interface address changes (either due to a readdressing of the original interface or switching to a new interface) the neighbor immediately updates the neighbor cache entry for the Client and begins accepting and sending packets according to the Client's new ANET address. This implicit mobility method applies to use cases such as cellphones with both WiFi and Cellular interfaces where only one of the interfaces is active at a given time, and the Client automatically switches over to the backup interface if the primary interface fails.

3.19.7. Moving to a New Server

When a Client associates with a new Server, it performs the Client procedures specified in Section 3.15.2. The Client then sends an RS message over any working ANET interface with destination set to the old Server's AERO address, with R set to 1 in the first SLLAO and with PD parameters to fully release itself from the old Server. The SLLAO also includes the SPAN address of the new Server in the Link Layer Address. If the Client does not receive an RA reply after MAX_RTR_SOLICITATIONS attempts over multiple underlying interfaces, the old Server may have failed and the Client should discontinue its release attempts.

When the old Server processes the RS, it sends unsolicited NA messages with a single TLLAO with Interface ID set to 0xffff and with D set to 1 to all ROSs in the Client's Report List. The Server also changes the symmetric neighbor cache entry state to DEPARTED, sets the link-layer address of the Client to the address found in the RS SLLAO, and sets a timer to DepartTime seconds. The Server then returns an RA message to the Client with Router Lifetime set to 0. After DepartTime seconds expires, the Server deletes the symmetric neighbor cache entry.

When the Client receives the RA message with Router Lifetime set to 0, it still must inform each of its remaining Proxies that it has released the old Server from service. To do so, it sends an RS over each remaining proxyed ANET interface with destination set to the old Server's AERO address and with R set to 1 in the first SLLAO but with no PD parameters. The Proxy will mark this Server as DEAPARTED and return an immediate RA without first performing an RS/RA exchange with the old Server.

Clients SHOULD NOT move rapidly between Servers in order to avoid causing excessive oscillations in the AERO routing system. Examples of when a Client might wish to change to a different Server include a Server that has gone unreachable, topological movements of significant distance, movement to a new geographic region, movement to a new segment, etc.

3.20. Multicast

The AERO Client serves as an IGMPv2 (IPv4) [RFC2236] or MLDv2 (IPv6) proxy [RFC3810][RFC4605] for its EUNs and/or hosted applications. The Client forwards IGMPv2/MLDv2 messages over any of its ANET interfaces for which group membership is required. The IGMP/MLDv2 messages may be further forwarded by a first-hop ANET access router acting as an IGMPv2/MLDv2-snooping switch [RFC4541], then ultimately delivered to an AERO Proxy/Server acting as a Protocol Independent Multicast - Sparse-Mode (PIM-SM) router [RFC7761]. AERO Gateways act as PIM-SM routers the same as AERO Proxys/Servers, except that no IGMPv2/MLDv2 proxying/snooping are necessary on the Gateway's attached EUNs.

When an AERO Proxy/Server/Gateway "X" acting as a PIM-SM router receives a Source-Specific Multicast (SSM) "Join" message for source "S" and group "G" (i.e., (S,G)), it forwards the message to a Relay via the SPAN. The SPAN then forwards the message to AERO Server "Y" which forwards the message to Proxy "Z" that services "S" (note that when "Y" is a Gateway there is no need for Proxy "Z".) Since the Relays in the SPAN do not examine Layer 3 control messages, this means that the (reverse) multicast tree path is simply from "S" to "Z" to "Y" to "X" with no other Layer 3 multicast-aware routers in the path. If "Z", "Y" and "X" are located on the same SPAN segment, the multicast data traffic between them can be sent via simple INET encapsulation and need not go over the SPAN. If any of "Z", "Y" and X" are located in different SPAN segments, however, SPAN encapsulation is necessary.

When an AERO Proxy/Server/Gateway "X" acting as a PIM-SM router receives an Any Source Multicast (ASM) "Join" message for source "*" and group "G" (i.e., (*,G)), it forwards the message toward the

Rendezvous Point "RP" for group "G" the same as if "RP" was the source "S". The (reverse) multicast tree path is therefore established in the same way as above.

After the (reverse) multicast tree path has been established via AERO Proxy "Z", the AERO Client "C" that hosts "S" may move to a different Proxy "Z2". In that case, the Client's multicast Server "Y" sends a PIM-SM "Join" to the new Proxy "Z2" for each multicast group "G", then sends a PIM-SM "Prune" to the old Proxy "Z".

After the (reverse) multicast tree path has been established, the AERO Client "C" may move to a different Server "Y2". In that case, the old Server "Y" must transfer its multicast tree state for all of Client "C"'s multicast sources to the new Server "Y2", and "Y2" must issue PIM-SM "Join" messages for each of Client "C"'s new Proxys.

4. Direct Underlying Interfaces

When a Client's AERO interface is configured over a Direct interface, the neighbor at the other end of the Direct link can receive packets without any encapsulation. In that case, the Client sends packets over the Direct link according to QoS preferences. If the Direct interface has the highest QoS preference, then the Client's IP packets are transmitted directly to the peer without going through an ANET/INET. If other interfaces have higher QoS preferences, then the Client's IP packets are transmitted via a different interface, which may result in the inclusion of Proxies, Servers and Relays in the communications path. Direct interfaces must be tested periodically for reachability, e.g., via NUD.

5. AERO Clients on the Open Internetwork

AERO Clients that connect to the open Internetwork via either a native or NATed interface can establish a VPN to securely connect to a Server. Alternatively, the Client can exchange ND messages directly with other AERO nodes on the same Internetwork using INET encapsulation only and without joining the SPAN. In that case, however, the Client must apply asymmetric security for ND messages to ensure routing and neighbor cache integrity (see: Section 14).

6. Operation over Multiple AERO Links

An AERO Client can connect to mutliple AERO links the same as for any Layer 2 service. In that case, the Client maintains a distinct AERO interface for each link, e.g., 'aero0' for the first link, 'aero1' for the second, 'aero2' for the third, etc. Each AERO link would include its own distinct set of Relays, Servers and Proxies, thereby providing redundancy in case of failures. Each AERO link would

service a distinct MSP such that the Client would receive multiple MNP delegations - one for each link.

The Relays, Servers and Proxies on each AERO link can assign AERO and SPAN addresses that use the same or different numberings from those on other links. Since the links are distinct there is no requirement for avoiding inter-link address duplication, e.g., the same AERO address such as fe80::1000 could be used to number distinct nodes that connect to different links.

Each AERO link could utilize the same or different ANET connections. The links can be distinguished at the link-layer via Virtual Local Area Network (VLAN) tagging the same as definied in IEEE 802.1Q. This gives rise to the opportunity for supporting multiple redundant networked paths, where each VLAN is distinguished by a different label (e.g., colors such as Red, Green, Blue, etc.). In particular, the Client can tag its RS messages with the appropriate label to cause the network to select the desired VLAN.

7. Operation on AERO Links with /64 ASPs

IPv6 AERO links typically have MSPs that cover many candidate MNPs of length /64 or shorter. However, in some cases it may be desirable to use AERO over links that have only a /64 MSP. This can be accommodated by treating all Clients on the AERO link as simple hosts that receive /128 prefix delegations.

In that case, the Client sends an RS message to the Server the same as for ordinary AERO links. The Server responds with an RA message that includes one or more /128 prefixes (i.e., singleton addresses) that include the /64 MSP prefix along with an interface identifier portion to be assigned to the Client. The Client and Server then configure their AERO addresses based on the interface identifier portions of the /128s (i.e., the lower 64 bits) and not based on the /64 prefix (i.e., the upper 64 bits).

For example, if the MSP for the host-only IPv6 AERO link is 2001:db8:1000:2000::/64, each Client will receive one or more /128 IPv6 prefix delegations such as 2001:db8:1000:2000::1/128, 2001:db8:1000:2000::2/128, etc. When the Client receives the prefix delegations, it assigns the AERO addresses fe80::1, fe80::2, etc. to the AERO interface, and assigns the global IPv6 addresses (i.e., the /128s) to either the AERO interface or an internal virtual interface such as a loopback. In this arrangement, the Client conducts route optimization in the same sense as discussed in Section 3.17.

This specification has applicability for nodes that act as a Client on an "upstream" AERO link, but also act as a Server on "downstream"

AERO links. More specifically, if the node acts as a Client to receive a /64 prefix from the upstream AERO link it can then act as a Server to provision /128s to Clients on downstream AERO links.

8. AERO Adaptations for SEcure Neighbor Discovery (SEND)

SEcure Neighbor Discovery (SEND) [RFC3971] and Cryptographically Generated Addresses (CGAs) [RFC3972] were designed to secure IPv6 ND messaging in environments where symmetric network and/or transport-layer security services are impractical (see: Section 14). AERO nodes that use SEND/CGA employ the following adaptations.

When a source AERO node prepares a SEND-protected ND message, it uses a link-local CGA as the IPv6 source address and writes the prefix embedded in its AERO address (i.e., instead of fe80::/64) in the CGA parameters Subnet Prefix field. When the neighbor receives the ND message, it first verifies the message checksum and SEND/CGA parameters while using the link-local prefix fe80::/64 (i.e., instead of the value in the Subnet Prefix field) to match against the IPv6 source address of the ND message.

The neighbor then derives the AERO address of the source by using the value in the Subnet Prefix field as the interface identifier of an AERO address. For example, if the Subnet Prefix field contains 2001:db8:1:2, the neighbor constructs the AERO address as fe80::2001:db8:1:2. The neighbor then caches the AERO address in the neighbor cache entry it creates for the source, and uses the AERO address as the IPv6 destination address of any ND message replies.

9. AERO Critical Infrastructure Considerations

AERO Relays can be either Commercial off-the Shelf (COTS) standard IP routers or virtual machines in the cloud. Relays must be provisioned, supported and managed by the INET administrative authority, and connected to the Relays of other INETs via inter-domain peerings. Cost for purchasing, configuring and managing Relays is nominal even for very large AERO links.

AERO Servers can be standard dedicated server platforms, but most often will be deployed as virtual machines in the cloud. The only requirements for Servers are that they can run the AERO user-level code and have at least one network interface connection to the INET. As with Relays, Servers must be provisioned, supported and managed by the INET administrative authority. Cost for purchasing, configuring and managing Servers is nominal especially for virtual Servers hosted in the cloud.

AERO Proxies are most often standard dedicated server platforms with one network interface connected to the ANET and a second interface connected to an INET. As with Servers, the only requirements are that they can run the AERO user-level code and have at least one interface connection to the INET. Proxies must be provisioned, supported and managed by the ANET administrative authority. Cost for purchasing, configuring and managing Proxies is nominal, and borne by the ANET administrative authority.

AERO combined Client/Servers can be any dedicated server or COTS router platform with one network interface connected to the INET and a second interface connected to a downstream attached network. The Client/Server joins the SPAN over the INET interface and engages in eBGP peering with one or more Relays as a stub AS. The Client/Server then injects its MNP into the BGP routing system, and provisions the MNP to its downstream-attached networks. No Client/Server ND messaging is necessary, and the Client/Server can perform ROS and ROR services the same as for any Server. The combined Client/Server construct is useful for connecting large fixed networks to the AERO link.

10. DNS Considerations

AERO Client MNs and INET correspondent nodes consult the Domain Name System (DNS) the same as for any Internetworking node. When correspondent nodes and Client MNs use different IP protocol versions (e.g., IPv4 correspondents and IPv6 MNs), the INET DNS must maintain A records for IPv4 address mappings to MNs which must then be populated in Gateway NAT64 mapping caches. In that way, an IPv4 correspondent node can send packets to the IPv4 address mapping of the target MN, and the Gateway will translate the IPv4 header and destination address into an IPv6 header and IPv6 destination address of the MN.

When an AERO Client registers with an AERO Server, the Server returns the address(es) of DNS servers in RDNSS options [RFC6106]. The DNS Server provides the IP addresses of other MNs and correspondent nodes in AAAA records for IPv6 or A records for IPv4.

11. Transition Considerations

The SPAN ensures that dissimilar INET segments can be joined into a single unified AERO link, even though the INET segments themselves may have differing protocol versions and/or incompatible addressing plans. However, a commonality can be achieved by incrementally distributing MNP prefixes to eventually reach all nodes (both mobile and fixed) in all segments. This can be accomplished by

incrementally deploying AERO Gateways on each INET segment, with each Gateway distributing its MNPs to its downstream-attached INET links.

This gives rise to the opportunity to eventually distribute MNP-based addresses to all nodes, and to present a unified AERO link view (bridged by the SPAN) even if the INET segments remain in their current protocol and addressing plans. In that way, the AERO link can serve the dual purpose of providing a mobility service and a transition service. Or, if an INET segment is transitioned to a protocol version and addressing scheme that is compatible with the AERO link MNP-based addressing scheme, the NET segment and AERO link can be joined by standard routers.

12. Implementation Status

An AERO implementation based on OpenVPN (https://openvpn.net/) was announced on the v6ops mailing list on January 10, 2018. The latest version is available at: http://linkupnetworks.net/aero/AERO-OpenVPN-2.0.tgz.

An initial public release of the AERO proof-of-concept source code was announced on the intarea mailing list on August 21, 2015. The latest version is available at: http://linkupnetworks.net/aero/aero-4.0.0.tgz.

A survey of public domain and commercial SEND implementations is available at https://www.ietf.org/mail-archive/web/its/current/msg02758.html.

13. IANA Considerations

The IANA has assigned a 4-octet Private Enterprise Number "45282" for AERO in the "enterprise-numbers" registry.

The IANA has assigned the UDP port number "8060" for an earlier experimental version of AERO [RFC6706]. This document obsoletes [RFC6706] and claims the UDP port number "8060" for all future use.

No further IANA actions are required.

14. Security Considerations

AERO link security considerations include considerations for both the data plane and the control plane.

Data plane security considerations are the same as for ordinary Internet communications. Application endpoints in AERO Clients and their EUNs SHOULD use application-layer security services such as

TLS/SSL [RFC8446], DTLS [RFC6347] or SSH [RFC4251] to assure the same level of protection as for critical secured Internet services. AERO Clients that require host-based VPN services SHOULD use symmetric network and/or transport layer security services such as TLS/SSL, DTLS, IPsec [RFC4301], etc. AERO Proxies and Servers can also provide a network-based VPN service on behalf of the Client, e.g., if the Client is located within a secured enclave and cannot establish a VPN on its own behalf.

Control plane security considerations are the same as for standard IPv6 Neighbor Discovery [RFC4861]. As fixed infrastructure elements, AERO Servers/Gateways and Proxies SHOULD pre-configure security associations for one or more Relays on their SPAN segments (e.g., using pre-placed keys) and use symmetric network and/or transport layer security services such as IPsec, TLS/SSL or DTLS to secure ND messages. The AERO Relays of all SPAN segments in turn SHOULD pre-configure security associations for their neighboring AERO Relays. AERO Clients that connect to secured enclaves need not apply security to their ND messages, since the messages will be intercepted by an enclave perimeter Proxy. AERO Clients located outside of secured enclaves SHOULD use symmetric network and/or transport layer security to secure their ND exchanges with Servers, but when there are many prospective neighbors with dynamically changing connectivity an asymmetric security service such as SEND may be needed (see: Section 8).

AERO Servers/Gateways and Relays present targets for traffic amplification Denial of Service (DoS) attacks. This concern is no different than for widely-deployed VPN security gateways in the Internet, where attackers could send spoofed packets to the gateways at high data rates. This can be mitigated by connecting Servers/Gateways and Relays over dedicated links with no connections to the Internet and/or when connections to the Internet are only permitted through well-managed firewalls. Traffic amplification DoS attacks can also target an AERO Client's low data rate links. This is a concern not only for Clients located on the open Internet but also for Clients in secured enclaves. AERO Servers/Gateways and Proxies can institute rate limits that protect Clients from receiving packet floods that could DoS low data rate links.

AERO Relays must implement ingress filtering to avoid a spoofing attack in which spurious SPAN messages are injected into an AERO link from an outside attacker. Restricting access to the link can be achieved by having Internetwork border routers implement ingress filtering to discard encapsulated packets injected into the link by an outside agent.

AERO Clients MUST ensure that their connectivity is not used by unauthorized nodes on their EUNs to gain access to a protected network, i.e., AERO Clients that act as routers MUST NOT provide routing services for unauthorized nodes. (This concern is no different than for ordinary hosts that receive an IP address delegation but then "share" the address with other nodes via some form of Internet connection sharing such as tethering.)

The MAP list MUST be well-managed and secured from unauthorized tampering, even though the list contains only public information. The MAP list can be conveyed to the Client, e.g., through secure upload of a static file, through DNS lookups, etc.

Although public domain and commercial SEND implementations exist, concerns regarding the strength of the cryptographic hash algorithm have been documented [RFC6273] [RFC4982].

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

15. Acknowledgements

Discussions in the IETF, aviation standards communities and private exchanges helped shape some of the concepts in this work. Individuals who contributed insights include Mikael Abrahamsson, Mark Andrews, Fred Baker, Bob Braden, Stewart Bryant, Brian Carpenter, Wojciech Dec, Ralph Droms, Adrian Farrel, Nick Green, Sri Gundavelli, Brian Haberman, Bernhard Haindl, Joel Halpern, Tom Herbert, Sascha Hlusiak, Lee Howard, Andre Kostur, Hubert Kuenig, Ted Lemon, Andy Malis, Satoru Matsushima, Tomek Mrugalski, Madhu Niraula, Alexandru Petrescu, Behcet Saikaya, Michal Skorepa, Joe Touch, Bernie Volz, Ryuji Wakikawa, Tony Whyman, Lloyd Wood and James Woodyatt. Members of the IESG also provided valuable input during their review process that greatly improved the document. Special thanks go to Stewart Bryant, Joel Halpern and Brian Haberman for their shepherding guidance during the publication of the AERO first edition.

This work has further been encouraged and supported by Boeing colleagues including Kyle Bae, M. Wayne Benson, Dave Bernhardt, Cam Brodie, Balaguruna Chidambaram, Irene Chin, Bruce Cornish, Claudiu Danilov, Wen Fang, Anthony Gregory, Jeff Holland, Seth Jahne, Ed King, Gene MacLean III, Rob Muszkiewicz, Sean O'Sullivan, Greg Saccone, Kent Shuey, Brian Skeen, Mike Slane, Carrie Spiker, Brendan Williams, Julie Wulff, Yueli Yang, Eric Yeh and other members of the BR&T and BIT mobile networking teams. Kyle Bae, Wayne Benson and Eric Yeh are especially acknowledged for implementing the AERO functions as extensions to the public domain OpenVPN distribution.

Earlier works on NBMA tunneling approaches are found in [RFC2529][RFC5214][RFC5569].

Many of the constructs presented in this second edition of AERO are based on the author's earlier works, including:

- The Internet Routing Overlay Network (IRON) [RFC6179][I-D.templin-ironbis]
- Virtual Enterprise Traversal (VET) [RFC5558][I-D.templin-intarea-vet]
- The Subnetwork Encapsulation and Adaptation Layer (SEAL) [RFC5320][I-D.templin-intarea-seal]
- AERO, First Edition [RFC6706]

Note that these works cite numerous earlier efforts that are not also cited here due to space limitations. The authors of those earlier works are acknowledged for their insights.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

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

This work is aligned with the Boeing autonomy program.

16. References

16.1. Normative References

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

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

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

[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>.

[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>.

[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>.

[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>.

16.2. Informative References

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

[I-D.ietf-dmm-distributed-mobility-anchoring]
Chan, A., Wei, X., Lee, J., Jeon, S., and C. Bernardos, "Distributed Mobility Anchoring", draft-ietf-dmm-distributed-mobility-anchoring-13 (work in progress), March 2019.

[I-D.ietf-intarea-gue]
Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", draft-ietf-intarea-gue-07 (work in progress), March 2019.

[I-D.ietf-intarea-gue-extensions]
Herbert, T., Yong, L., and F. Templin, "Extensions for Generic UDP Encapsulation", draft-ietf-intarea-gue-extensions-06 (work in progress), March 2019.

[I-D.ietf-intarea-tunnels]
Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", draft-ietf-intarea-tunnels-09 (work in progress), July 2018.

[I-D.ietf-rtgwg-atn-bgp]
Templin, F., Saccone, G., Dawra, G., Lindem, A., and V. Moreno, "A Simple BGP-based Mobile Routing System for the Aeronautical Telecommunications Network", draft-ietf-rtgwg-atn-bgp-01 (work in progress), January 2019.

[I-D.templin-6man-dhcpv6-ndopt]
Templin, F., "A Unified Stateful/Stateless Configuration Service for IPv6", draft-templin-6man-dhcpv6-ndopt-07 (work in progress), December 2018.

[I-D.templin-intarea-grefrag]
Templin, F., "GRE Tunnel Level Fragmentation", draft-templin-intarea-grefrag-04 (work in progress), July 2016.

[I-D.templin-intarea-seal]
Templin, F., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", draft-templin-intarea-seal-68 (work in progress), January 2014.

[I-D.templin-intarea-vet]
Templin, F., "Virtual Enterprise Traversal (VET)", draft-templin-intarea-vet-40 (work in progress), May 2013.

[I-D.templin-ironbis]
Templin, F., "The Interior Routing Overlay Network (IRON)", draft-templin-ironbis-16 (work in progress), March 2014.

[I-D.templin-v6ops-pdhost]
Templin, F., "IPv6 Prefix Delegation and Multi-Addressing Models", draft-templin-v6ops-pdhost-23 (work in progress), December 2018.

[OVPN] OpenVPN, O., "http://openvpn.net", October 2016.

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

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

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

[RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <https://www.rfc-editor.org/info/rfc1812>.

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

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <https://www.rfc-editor.org/info/rfc2236>.

[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>.

[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>.

[RFC2764] Gleeson, B., Lin, A., Heinanen, J., Armitage, G., and A. Malis, "A Framework for IP Based Virtual Private Networks", RFC 2764, DOI 10.17487/RFC2764, February 2000, <https://www.rfc-editor.org/info/rfc2764>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <https://www.rfc-editor.org/info/rfc2890>.

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

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

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

[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>.

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

[RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/info/rfc4251>.

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

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[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>.

[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>.

[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>.

[RFC4511] Sermersheim, J., Ed., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, DOI 10.17487/RFC4511, June 2006, <https://www.rfc-editor.org/info/rfc4511>.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, DOI 10.17487/RFC4541, May 2006, <https://www.rfc-editor.org/info/rfc4541>.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, DOI 10.17487/RFC4605, August 2006, <https://www.rfc-editor.org/info/rfc4605>.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <https://www.rfc-editor.org/info/rfc4607>.

[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>.

[RFC4982] Bagnulo, M. and J. Arkko, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)", RFC 4982, DOI 10.17487/RFC4982, July 2007, <https://www.rfc-editor.org/info/rfc4982>.

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, DOI 10.17487/RFC5214, March 2008, <https://www.rfc-editor.org/info/rfc5214>.

[RFC5320] Templin, F., Ed., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", RFC 5320, DOI 10.17487/RFC5320, February 2010, <https://www.rfc-editor.org/info/rfc5320>.

[RFC5522] Eddy, W., Ivancic, W., and T. Davis, "Network Mobility Route Optimization Requirements for Operational Use in Aeronautics and Space Exploration Mobile Networks", RFC 5522, DOI 10.17487/RFC5522, October 2009, <https://www.rfc-editor.org/info/rfc5522>.

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

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, DOI 10.17487/RFC5569, January 2010, <https://www.rfc-editor.org/info/rfc5569>.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, DOI 10.17487/RFC6106, November 2010, <https://www.rfc-editor.org/info/rfc6106>.

[RFC6179] Templin, F., Ed., "The Internet Routing Overlay Network (IRON)", RFC 6179, DOI 10.17487/RFC6179, March 2011, <https://www.rfc-editor.org/info/rfc6179>.

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

[RFC6273] Kukec, A., Krishnan, S., and S. Jiang, "The Secure Neighbor Discovery (SEND) Hash Threat Analysis", RFC 6273, DOI 10.17487/RFC6273, June 2011, <https://www.rfc-editor.org/info/rfc6273>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

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

[RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field", RFC 6864, DOI 10.17487/RFC6864, February 2013, <https://www.rfc-editor.org/info/rfc6864>.

[RFC7269] Chen, G., Cao, Z., Xie, C., and D. Binet, "NAT64 Deployment Options and Experience", RFC 7269, DOI 10.17487/RFC7269, June 2014, <https://www.rfc-editor.org/info/rfc7269>.

[RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", RFC 7333, DOI 10.17487/RFC7333, August 2014, <https://www.rfc-editor.org/info/rfc7333>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <https://www.rfc-editor.org/info/rfc8086>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

Appendix A. AERO Alternate Encapsulations

When GUE encapsulation is not needed, AERO can use common encapsulations such as IP-in-IP [RFC2003][RFC2473][RFC4213], Generic Routing Encapsulation (GRE) [RFC2784][RFC2890] and others. The encapsulation is therefore only differentiated from non-AERO tunnels through the application of AERO control messaging and not through, e.g., a well-known UDP port number.

As for GUE encapsulation, alternate AERO encapsulation formats may require encapsulation layer fragmentation. For simple IP-in-IP encapsulation, an IPv6 fragment header is inserted directly between the inner and outer IP headers when needed, i.e., even if the outer header is IPv4. The IPv6 Fragment Header is identified to the outer IP layer by its IP protocol number, and the Next Header field in the IPv6 Fragment Header identifies the inner IP header version. For GRE encapsulation, a GRE fragment header is inserted within the GRE header [I-D.templin-intarea-grefrag].

Figure 6 shows the AERO IP-in-IP encapsulation format before any fragmentation is applied:

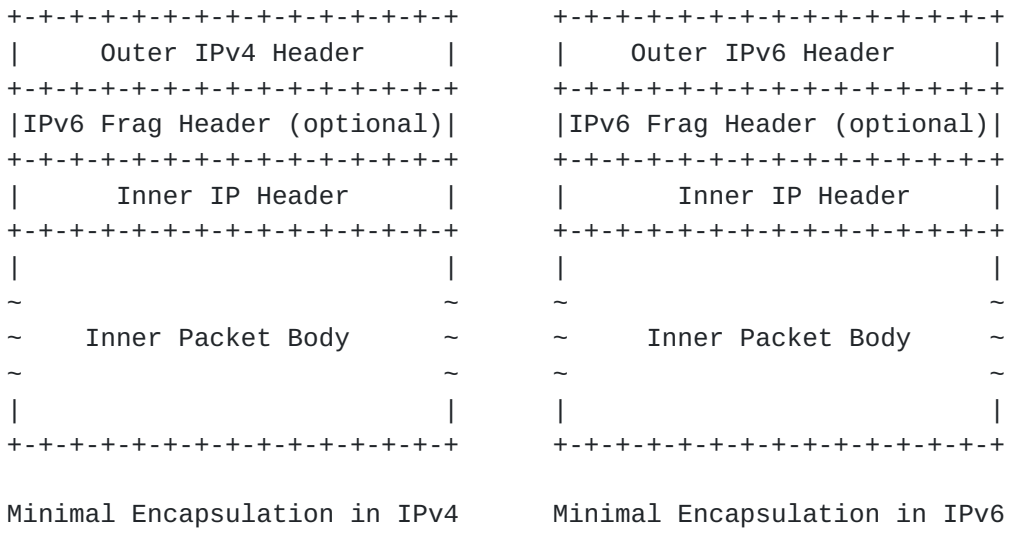

Figure 6: Minimal Encapsulation Format using IP-in-IP

Figure 7 shows the AERO GRE encapsulation format before any fragmentation is applied:

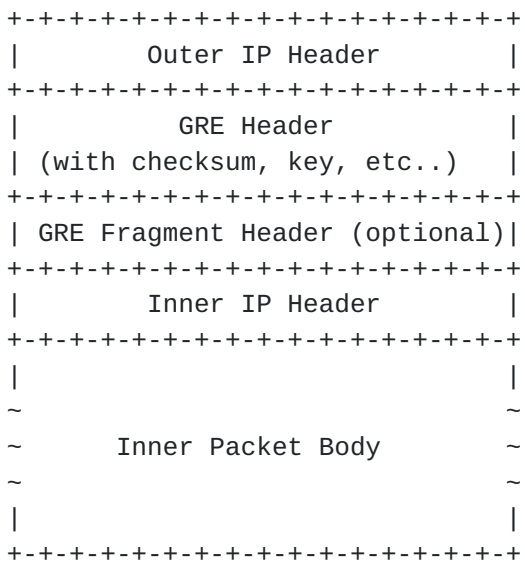

Figure 7: Minimal Encapsulation Using GRE

Alternate encapsulation may be preferred in environments where GUE encapsulation would add unnecessary overhead. For example, certain low-bandwidth wireless data links may benefit from a reduced encapsulation overhead.

GUE encapsulation can traverse network paths that are inaccessible to non-UDP encapsulations, e.g., for crossing Network Address Translators (NATs). More and more, network middleboxes are also being configured to discard packets that include anything other than a well-known IP protocol such as UDP and TCP. It may therefore be necessary to determine the potential for middlebox filtering before enabling alternate encapsulation in a given environment.

In addition to IP-in-IP, GRE and GUE, AERO can also use security encapsulations such as IPsec, TLS/SSL, DTLS, etc. In that case, AERO control messaging and route determination occur before security encapsulation is applied for outgoing packets and after security decapsulation is applied for incoming packets.

AERO is especially well suited for use with VPN system encapsulations such as OpenVPN [OVPN].

Appendix B. S/TLLAO Extensions for Special-Purpose Links

The AERO S/TLLAO format specified in Section 3.6 includes a Length value of 5 (i.e., 5 units of 8 octets). However, special-purpose links may extend the basic format to include additional fields and a Length value larger than 5.

For example, adaptation of AERO to the Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS) includes link selection preferences based on transport port numbers in addition to the existing DSCP-based preferences. ATN/IPS nodes maintain a map of transport port numbers to 64 possible preference fields, e.g., TCP port 22 maps to preference field 8, TCP port 443 maps to preference field 20, UDP port 8060 maps to preference field 34, etc. The extended S/TLLAO format for ATN/IPS is shown in Figure 8, where the Length value is 7 and the 'Q(i)' fields provide link preferences for the corresponding transport port number.

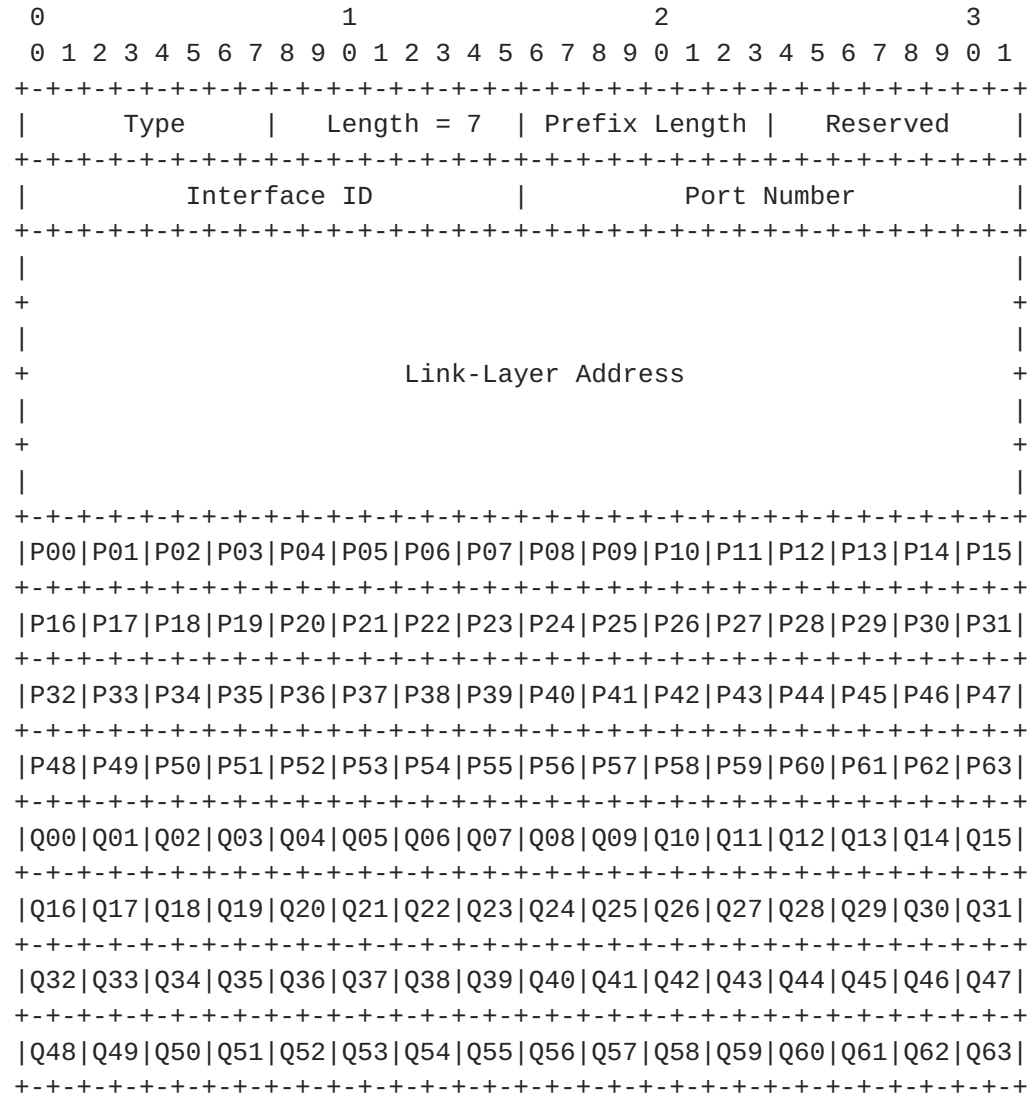

Figure 8: ATN/IPS Extended S/TLLAO Format

Appendix C. Change Log

<< RFC Editor - remove prior to publication >>

Changes from draft-templin-intarea-6706bis-11 to draft-templin-intrea-6706bis-12:

- Introduced Gateways as a new AERO element for connecting Correspondent Nodes on INET links
- Introduced terms "Access Network (ANET)" and "Internetwork (INET)"
- Changed "ASP" to "MSP", and "ACP" to "MNP"
- New figure on the relation of Segments to the SPAN and AERO link
- New "S" bit in S/TLLAO to indicate the "Source" S/TLLAO as opposed to additional S/TLLAOs
- Changed Interface ID for Servers from 255 to 0xffff
- Significant updates to Route Optimization, NUD, and Mobility Management
- New Section on Multicast
- New Section on AERO Clients in the open Internetwork
- New Section on Operation over multiple AERO links (VLANs over the SPAN)
- New Sections on DNS considerations and Transition considerations
-

Changes from draft-templin-intarea-6706bis-10 to draft-templin-intrea-6706bis-11:

- Added The SPAN

Changes from draft-templin-intarea-6706bis-09 to draft-templin-intrea-6706bis-10:

- Orphaned packets in flight (e.g., when a neighbor cache entry is in the DEPARTED state) are now forwarded at the link layer instead of at the network layer. Forwarding at the network layer can result in routing loops and/or excessive delays of forwarded packets while the routing system is still reconverging.

- Update route optimization to clarify the unsecured nature of the first NS used for route discovery

- Many cleanups and clarifications on ND messaging parameters

Changes from draft-templin-intarea-6706bis-08 to draft-templin-intrea-6706bis-09:

- Changed PRL to "MAP list"

- For neighbor cache entries, changed "static" to "symmetric", and "dynamic" to "asymmetric"

- Specified Proxy RS/RA exchanges with Servers on behalf of Clients

- Added discussion of unsolicited NAs in Section 3.16, and included forward reference to Section 3.18

- Added discussion of AERO Clients used as critical infrastructure elements to connect fixed networks.

- Added network-based VPN under security considerations

Changes from draft-templin-intarea-6706bis-07 to draft-templin-intrea-6706bis-08:

- New section on AERO-Aware Access Router

Changes from draft-templin-intarea-6706bis-06 to draft-templin-intrea-6706bis-07:

- Added "R" bit for release of PDs. Now have a full RS/RA service that can do PD without requiring DHCPv6 messaging over-the-air

- Clarifications on solicited vs unsolicited NAs

- Clarified use of MAX_NEIGHBOR_ADVERTISEMENTS for the purpose of increase reliability

Changes from draft-templin-intarea-6706bis-05 to draft-templin-intrea-6706bis-06:

- Major re-work and simplification of Route Optimization function

- Added Distributed Mobility Management (DMM) and Mobility Anchor Point (MAP) terminology

- New section on "AERO Critical Infrastructure Element Considerations" demonstrating low overall cost for the service
- minor text revisions and deletions
- removed extraneous appendices

Changes from draft-templin-intarea-6706bis-04 to draft-templin-intrea-6706bis-05:

- New Appendix E on S/TLLAO Extensions for special-purpose links. Discussed ATN/IPS as example.
- New sentence in introduction to declare appendices as non-normative.

Changes from draft-templin-intarea-6706bis-03 to draft-templin-intrea-6706bis-04:

- Added definitions for Potential Router List (PRL) and secure enclave
- Included text on mapping transport layer port numbers to network layer DSCP values
- Added reference to DTLS and DMM Distributed Mobility Anchoring working group document
- Reworked Security Considerations
- Updated references.

Changes from draft-templin-intarea-6706bis-02 to draft-templin-intrea-6706bis-03:

- Added new section on SEND.
- Clarifications on "AERO Address" section.
- Updated references and added new reference for RFC8086.
- Security considerations updates.
- General text clarifications and cleanup.

Changes from draft-templin-intarea-6706bis-01 to draft-templin-intrea-6706bis-02:

- Note on encapsulation avoidance in Section 4.

Changes from draft-templin-intarea-6706bis-00 to draft-templin-intrea-6706bis-01:

- Remove DHCPv6 Server Release procedures that leveraged the old way Relays used to "route" between Server link-local addresses

- Remove all text relating to Relays needing to do any AERO-specific operations

- Proxy sends RS and receives RA from Server using SEND. Use CGAs as source addresses, and destination address of RA reply is to the AERO address corresponding to the Client's ACP.

- Proxy uses SEND to protect RS and authenticate RA (Client does not use SEND, but rather relies on subnetwork security. When the Proxy receives an RS from the Client, it creates a new RS using its own addresses as the source and uses SEND with CGAs to send a new RS to the Server.

- Emphasize distributed mobility management

- AERO address-based RS injection of ACP into underlying routing system.

Changes from draft-templin-aerolink-82 to draft-templin-intarea-6706bis-00:

- Document use of NUD (NS/NA) for reliable link-layer address updates as an alternative to unreliable unsolicited NA. Consistent with Section 7.2.6 of RFC4861.

- Server adds additional layer of encapsulation between outer and inner headers of NS/NA messages for transmission through Relays that act as vanilla IPv6 routers. The messages include the AERO Server Subnet Router Anycast address as the source and the Subnet Router Anycast address corresponding to the Client's ACP as the destination.

- Clients use Subnet Router Anycast address as the encapsulation source address when the access network does not provide a topologically-fixed address.

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org