Network Working Group
Internet-Draft
Obsoletes: rfc6706 (if approved)
Intended status: Standards Track
Expires: June 22, 2014

F. Templin, Ed.
Boeing Research & Technology

December 19, 2013

Transmission of IPv6 Packets over Asymmetric Extended Route Optimization (AERO) Links

draft-templin-aerolink-00.txt

Abstract

This document specifies the operation of IPv6 over tunnel virtual Non-Broadcast, Multiple Access (NBMA) links using Asymmetric Extended Route Optimization (AERO). Nodes attached to AERO links can exchange packets via trusted intermediate routers on the link that provide forwarding services to reach off-link destinations and/or redirection services to inform the node of an on-link neighbor that is closer to the final destination. Operation of the IPv6 Neighbor Discovery (ND) over AERO links is based on an IPv6 link local address format known as the AERO address.

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 http://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 June 22, 2014.

Table of Contents

1. Introduction

This document specifies the operation of IPv6 over tunnel virtual Non-Broadcast, Multiple Access (NBMA) links using Asymmetric Extended Route Optimization (AERO). Nodes attached to AERO links can exchange packets via trusted intermediate routers on the link that provide forwarding services to reach off-link destinations and/or redirection services to inform the node of an on-link neighbor that is closer to the final destination.

AERO uses an IPv6 link-local address format known as the AERO Address. This address type has properties that statelessly link IPv6 Neighbor Discovery (ND) to IPv6 routing. The AERO link can be used for tunneling to neighboring nodes on either IPv6 or IPv4 networks, i.e., AERO views the IPv6 and IPv4 networks as equivalent links for tunneling. The remainder of this document presents the AERO specification.

2. Terminology

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

AERO link
: a Non-Broadcast, Multiple Access (NBMA) tunnel virtual overlay configured over a node's attached IPv6 and/or IPv4 networks. All nodes on the AERO link appear as single-hop neighbors from the perspective of IPv6.

AERO interface
: a node's attachment to an AERO link.

AERO address
: an IPv6 link-local address assigned to an AERO interface and constructed as specified in Section 3.3.

AERO node
: a node that is connected to an AERO link and that participates in IPv6 Neighbor Discovery over the link.

AERO Server ("server")
: a node that configures an advertising router interface on an AERO link over which it can provide default forwarding and redirection services for other AERO nodes.

AERO Client ("client")
a node that configures a non-advertising router interface on an AERO link over which it can connect End User Networks (EUNs) to the AERO link.

AERO Relay ("relay")
a node that relays IPv6 packets between Servers on the same AERO link, and/or that forwards IPv6 packets between the AERO link and the IPv6 Internet. An AERO Relay may or may not also be configured as an AERO Server.

ingress tunnel endpoint (ITE)
a node that injects tunneled packets into an AERO link.

egress tunnel endpoint (ETE)
a node that receives tunneled packets from an AERO link.

underlying network
a connected IPv6 or IPv4 network routing region over which AERO links tunnel IPv6 packets.

underlying interface
a node's interface point of attachment to an underlying network.

underlying address
an IPv6 or IPv4 address assigned to a node's underlying interface. When UDP encapsulation is used, the UDP port number is also considered as part of the link-layer address. Link-layer addresses are used as the source and destination addresses of the AERO encapsulation header.

link-layer address
the same as defined for "underlying address" above.

network layer address
an IPv6 address used as the source or destination address of the inner IPv6 packet header.

end user network (EUN)
an IPv6 network attached to a downstream interface of an AERO Client (where the AERO interface is seen as the upstream interface).

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].

3. Asymmetric Extended Route Optimization (AERO)

The following sections specify the operation of IPv6 over Asymmetric Extended Route Optimization (AERO) links:

3.1. AERO Interface Characteristics

All nodes connected to an AERO link configure their AERO interfaces as router interfaces (not host interfaces). End system applications therefore do not bind directly to the AERO interface, but rather bind to end user network (EUN) interfaces beyond which their packets may be forwarded over an AERO interface.

AERO interfaces use IPv6-in-IPv6 encapsulation [RFC2473] to exchange tunneled packets with AERO neighbors attached to an underlying IPv6 network and use IPv6-in-IPv4 encapsulation [RFC4213] to exchange tunneled packets with AERO neighbors attached to an underlying IPv4 network. AERO interfaces can also use IPsec encapsulation [RFC4301] (either IPv6-in-IPv6 or IPv6-in-IPv4) in environments where strong authentication and confidentiality are required.

AERO interfaces further use the Subnetwork Encapsulation and Adaptation Layer (SEAL)[I-D.templin-intarea-seal] and can therefore configure an unlimited Maximum Transmission Unit (MTU). This entails the insertion of a SEAL header (i.e., an IPv6 fragment header with the S bit set to 1) above the outer IP encapsulation header. When NAT traversal and/or filtering middlebox traversal is necessary, a UDP header is further inserted between the outer IP encapsulation header and the SEAL header.

AERO interfaces maintain a neighbor cache and use an adaptation of standard unicast IPv6 ND messaging in which Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages do not include Source/Target Link Layer Address Options (S/TLLAO). Instead, AERO nodes discover the link-layer addresses of neighbors by examining the encapsulation source address of any RS/RA/NS/NA messages they receive and ignore any S/TLLAOs included in these messages. This is vital to the operation of AERO in environments in which AERO neighbors are separated by Network Address Translators (NATs) - either IPv4 or IPv6.

AERO Redirect messages include a TLLAO the same as for any IPv6 link. The TLLAO includes the link-layer address of the target node, including both the IP address and the UDP source port number used by the target when it sends UDP-encapsulated packets over the AERO interface (the TLLAO instead encodes the value 0 when the target does not use UDP encapsulation). TLLAOs for target nodes that use an IPv6 underlying address include the full 16 bytes of the IPv6 address as

shown in Figure 1, while TLLAOs for target nodes that use an IPv4 underlying address include only the 4 bytes of the IPv4 address as shown in Figure 2.

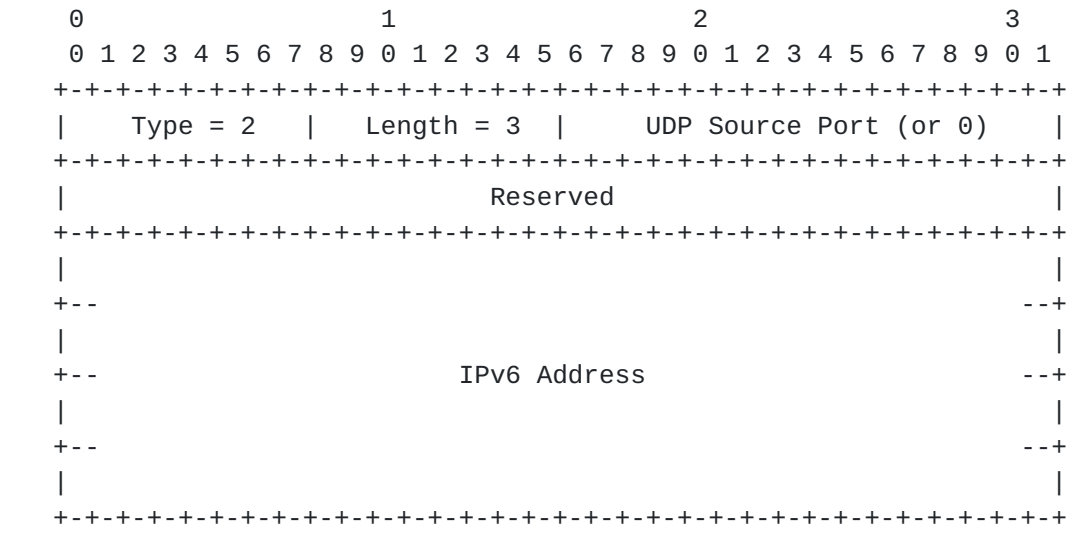

Figure 1: AERO TLLAO Format for IPv6

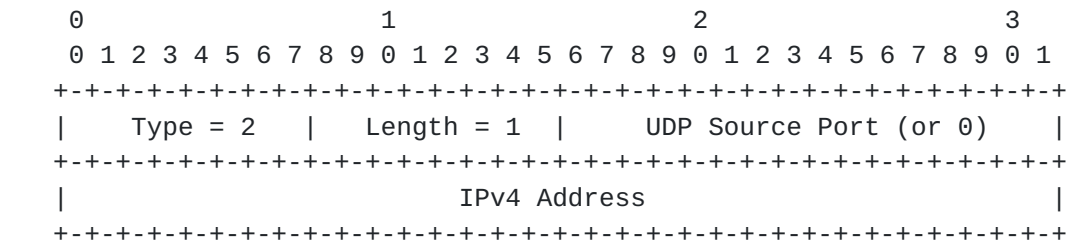

Figure 2: AERO TLLAO Format for IPv4

Finally, nodes on AERO interfaces use a simple data origin authentication for encapsulated packets they receive from other nodes. In particular, AERO Clients accept encapsulated packets with a link-layer source address belonging to their current AERO Server. AERO nodes also accept encapsulated packets with a link-layer source address that is correct for the network-layer source address. The AERO node considers the link-layer source address correct for the network-layer source address if there is an IPv6 route that matches the network-layer source address as well as a neighbor cache entry corresponding to the next hop that includes the link-layer address.

3.2. AERO Node Types

AERO Servers configure their AERO link interfaces as advertising router interfaces (see [RFC4861], Section 6.2.2) and may therefore send Router Advertisement (RA) messages that include non-zero Router Lifetimes.

AERO Clients configure their AERO link interfaces as non-advertising router interfaces, i.e., even if the AERO Client otherwise displays the outward characteristics of an ordinary host (for example, the Client may internally configure both an AERO interface and (virtual) EUN interfaces). AERO Clients are provisioned with IPv6 Prefix Delegations either through a DHCPv6 Prefix Delegation exchange with an AERO Server over the AERO link or via a static delegation obtained through an out-of-band exchange with an AERO link prefix delegation authority.

AERO Relays relay packets between Servers connected to the same AERO link and also forward packets between the AERO link and the native IPv6 network. The relaying process entails re-encapsulation of IPv6 packets that were received from a first AERO Server and are to be forwarded without modification to a second AERO Server. This relaying process can best be understood as a form of bridging.

3.3. AERO Addresses

An AERO address is an IPv6 link-local address assigned to an AERO interface and with an IPv6 prefix embedded within the 64-bit interface identifier.

Each AERO Client configures an AERO address based on the delegated prefix it has received from the AERO link prefix delegation authority. The address begins with the prefix fe80::/64 and includes in its interface identifier the base /64 prefix from the Client's delegated IPv6 prefix. For example, if an AERO Client has received the prefix delegation 2001:db8:1000:2000::/56 it would construct its AERO address as fe80::2001:db8:1000:2000. An AERO Client may receive multiple IPv6 prefix delegations, in which case it would configure multiple AERO addresses - one for each delegated prefix.

Each AERO Server configures the special AERO address fe80::1 to support the operation of IPv6 Neighbor Discovery over the AERO link; the address therefore has the properties of an IPv6 Anycast address. While all Servers configure the same AERO address and therefore cannot be distinguished from one another at the network layer, Clients can still distinguish Servers at the link layer by examining the Servers' link-layer addresses.

Nodes that are configured as pure AERO Relays (i.e., and that do not also act as Servers) do not configure an IPv6 address of any kind on their AERO interfaces. The Relay's AERO interface is therefore used purely for transit and does not participate in IPv6 ND message exchanges.

3.4. AERO Reference Operational Scenario

Figure 3 depicts the AERO reference operational scenario. The figure shows an AERO Server('A'), two AERO Clients ('B', 'D') and three ordinary IPv6 hosts ('C', 'E', 'F'):

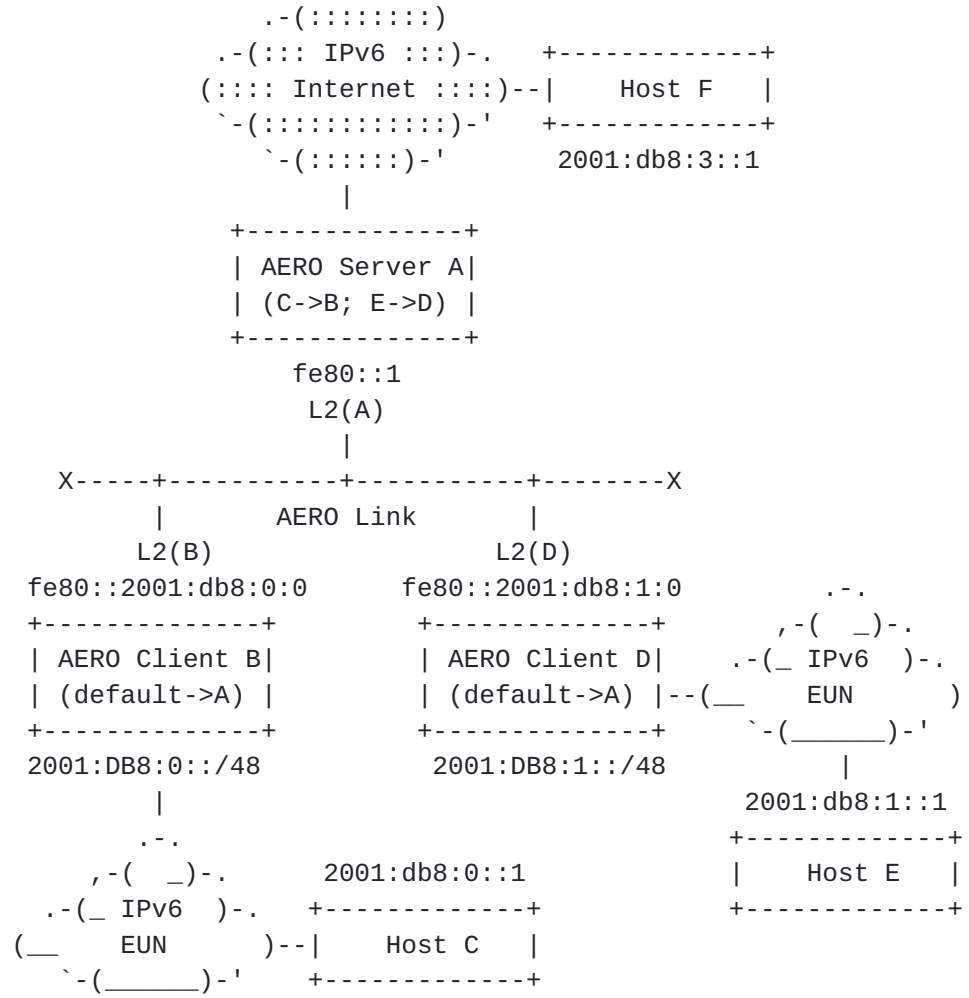

Figure 3: AERO Reference Operational Scenario

In Figure 3, AERO Server ('A') connects to the AERO link and connects to the IPv6 Internet, either directly or via other IPv6 routers (not shown). Server ('A') assigns the address fe80::1 to its AERO interface with link-layer address L2(A). Server ('A') next arranges to add L2(A) to a published list of valid Servers for the AERO link.

AERO Client ('B') assigns the address fe80::2001:db8:0:0 to its AERO interface with link-layer address L2(B). Client ('B') configures a default route via the AERO interface with next-hop network-layer address fe80::1 and link-layer address L2(A), then sub-delegates the prefix 2001:db8:0::/48 to its attached EUNs. IPv6 host ('C') connects to the EUN, and configures the network-layer address 2001:db8:0::1.

AERO Client ('D') assigns the address fe80::2001:db8:1:0 to its AERO interface with link-layer address L2(D). Client ('D') configures a default route via the AERO interface with next-hop network-layer address fe80::1 and link-layer address L2(A), then sub-delegates the network-layer prefix 2001:db8:1::/48 to its attached EUNs. IPv6 host ('E') connects to the EUN, and configures the network-layer address 2001:db8:1::1.

Finally, IPv6 host ('F') connects to an IPv6 network outside of the AERO link domain. Host ('F') configures its IPv6 interface in a manner specific to its attached IPv6 link, and assigns the network-layer address 2001:db8:3::1 to its IPv6 link interface.

3.5. AERO Prefix Delegation and Router Discovery

3.5.1. AERO Client Behavior

AERO Clients observe the IPv6 router requirements defined in [RFC6434] except that they act as "hosts" on their AERO interfaces for the purpose of prefix delegation and router discovery in the same fashion as for IPv6 Customer Premises Equipment (CPE) routers [RFC6204]. AERO Clients first discover the link-layer address of an AERO Server via static configuration, or through an automated means such as DNS name resolution. After discovering the link-layer address, the Client then acts as a requesting router to obtain IPv6 prefixes through DHCPv6 Prefix Delegation [RFC3633] via the Server. (The Client can also obtain prefixes through out-of-band means such as static administrative configuration, etc.). After the Client acquires prefixes, it sub-delegates them to nodes and links within its attached EUNs. It also assigns the link-local AERO address(es) taken from its delegated prefix(es) to the AERO interface (see: Section 3.3).

After acquiring prefixes, the Client next prepares a unicast IPv6 Router Solicitation (RS) message using its AERO address as the network-layer source address and fe80::1 as the network-layer destination address. The Client then tunnels the packet to the Server using one of its underlying addresses as the link-layer source address and using an underlying address of the Server as the link-layer destination address. The Server in turn returns a unicast

Router Advertisement (RA) message, which the Client uses to create an IPv6 neighbor cache entry for the Server on the AERO interface per [RFC4861]. The link-layer address for the neighbor cache entry is taken from the link-layer source address of the RA message.

After obtaining prefixes and performing an initial RS/RA exchange with a Server, the Client continues to send periodic RS messages to the server to obtain new RAs in order to keep neighbor cache entries alive. The Client can also forward IPv6 packets destined to networks beyond its local EUNs via the Server as an IPv6 default router. The Server may in turn return a Redirect message informing the Client of a neighbor on the AERO link that is topologically closer to the final destination as specified in Section 3.7.

3.5.2. AERO Server Behavior

AERO Servers observe the IPv6 router requirements defined in [RFC6434]. They further configure a DHCPv6 relay/server function on their AERO links and/or provide an administrative interface for delegation of network-layer addresses and prefixes. When the Server delegates prefixes, it also establishes forwarding table entries that list the AERO address of the Client as the next hop toward the delegated IPv6 prefixes (where the AERO address is constructed as specified in Section 3.3).

Servers respond to RS messages from Clients on their advertising AERO interfaces by returning an RA message. When the Server receives an RS message, it creates or updates a neighbor cache entry using the network layer source address as the neighbor's network layer address and using the link-layer source address of the RS message as the neighbor's link-layer address.

When the Server forwards a packet via the same AERO interface on which it arrived, it initiates an AERO route optimization procedure as specified in Section 3.7.

3.6. AERO Neighbor Unreachability Detection

When an AERO Client forwards a packet originating from one of its EUNs via an IPv6 route for which the next hop is reached via the AERO interface, it first consults its neighbor cache to determine the link-layer address of the next hop. The Client then encapsulates the packet and uses its link-layer address as the link-layer source address and the link-layer address of the neighbor as the link-layer destination address. If the IPv6 route is more-specific than "default", the Client also follows the Neighbor Unreachability Detection (NUD) procedures in Section 7.3 of [RFC4861] to keep neighbor cache entries alive. In particular, the Client sends NS

messages including its AERO address as the network-layer source address, the next hop's AERO address as the network-layer destination address, and the same link-layer addresses used to forward the packet. If the Client receives a solicited NA message response from the next hop, it updates its neighbor cache entry state for the next hop to REACHABLE, and records the link-layer source address of the NA as the neighbor's link layer address.

When an AERO Client receives an NS message used for NUD on an AERO interface, it either creates or updates a neighbor cache entry for the neighbor that sent the NS including recording the link-layer source address of the NS as the link layer address. If the Client creates a new neighbor cache entry it sets the neighbor's state to STALE. The Client then sends a solicited NA message back to the source including the Client's AERO address as the network-layer source address, the network-layer source address of the NS message as the network-layer destination address, its own link-layer address as the link-layer source address, and the link-layer source address of the NS message as the link-layer destination address.

AERO Servers process NS/NA messages in the same way as AERO Clients. However, they need not actively perform NUD if they have other means of determining Client reachability (e.g., through RS/RA exchanges, through reachability confirmation from the prefix delegation service, through link-layer probing, etc.).

3.7. AERO Redirection

Section 3.4 describes the AERO reference operational scenario. We now discuss the operation and protocol details of AERO Redirection with respect to this reference scenario.

3.7.1. Classical Redirection Approaches

With reference to Figure 3, when the IPv6 source host ('C') sends a packet to an IPv6 destination host ('E'), the packet is first forwarded via the EUN to AERO Client ('B'). Client ('B') then forwards the packet over its AERO interface to AERO Server ('A'), which then forwards the packet to AERO Client ('D'), where the packet is finally forwarded to the IPv6 destination host ('E'). When Server ('A') forwards the packet back out on its advertising AERO interface, it must arrange to redirect Client ('B') toward Client ('D') as a better next-hop node on the AERO link that is closer to the final destination. However, this redirection process should only occur if there is assurance that both Client nodes are willing participants.

Consider a first alternative in which Server ('A') informs Client ('B') only and does not inform Client ('D') (i.e., "classical

redirection"). In that case, Client ('D') has no way of knowing that Client ('B') is authorized to forward packets from their claimed network-layer source addresses, and it may simply elect to drop the packets. Also, Client ('B') has no way of knowing whether Client ('D') is performing some form of source address filtering that would reject packets arriving from a node other than a trusted default router, nor whether Client ('D') is even reachable via a direct path that does not involve Server ('A'). Finally, Client ('B') has no way of knowing whether the final destination ('E') has moved away from Client ('D').

Consider a second alternative in which Server ('A') informs both Client ('B') and Client ('D') separately, via independent redirection control messages (i.e., "augmented redirection"). In that case, several conditions can occur that could result in communication failures. First, if Client ('B') receives the redirection control message but Client ('D') does not, subsequent packets sent by Client ('B') could be dropped due to filtering since Client ('D') would not have cached state to verify their source network-layer addresses. Second, if Client ('D') receives the redirection control message but Client ('B') does not, subsequent packets sent in the reverse direction by Client ('D') would be lost. Finally, timing issues surrounding the establishment and garbage collection of neighbor state at the two Client nodes could yield unpredictable behavior. For example, unless the timing were carefully coordinated through some form of synchronization loop, there would invariably be instances in which one node has the correct neighbor state and the other node does not resulting in non-deterministic packet loss.

Since both of these alternatives have shortcomings, a new redirection technique (i.e., "AERO redirection") is needed.

3.7.2. AERO Redirection Concept of Operations

Again, with reference to Figure 3, when source host ('C') sends a packet to destination host ('E'), the packet is first forwarded over the source host's attached EUN to Client ('B'), which then forwards the packet via its AERO interface to Server ('A').

Using AERO redirection, Server ('A') then forwards the packet out the same AERO interface toward Client ('D') and also sends an AERO "Predirect" message forward to Client ('D') as specified in Section 3.7.4. The Predirect message includes Client ('B')'s network- and link-layer addresses as well as information that Client ('D') can use to determine the IPv6 prefix used by Client ('B') . After Client ('D') receives the Predirect message, it process the message and returns an AERO Redirect message to Server ('A') as specified in Section 3.7.5. During the process, Client ('D') also

creates or updates a neighbor cache entry for Client ('B'), and creates an IPv6 route to be used as ingress filtering information to accept future packets using addresses matched by Client ('B')'s prefix.

When Server ('A') receives the Redirect message, it processes the message and forwards it on to Client ('B') as specified in Section 3.7.6. The message includes Client ('D')'s network- and link-layer addresses as well as information that Client ('B') can use to determine the IPv6 prefix used by Client ('D'). After Client ('B') receives the Redirect message, it processes the message as specified in Section 3.7.7. During the process, Client ('B') also creates or updates a neighbor cache entry for Client ('D'), and creates an IPv6 route for forwarding future packets using addresses matched by the prefixes to Client ('D').

Following the above Predirect/Redirect message exchange, forwarding of packets from Client ('B') to Client ('D') without involving Server ('A) as an intermediary is enabled. The mechanisms that support this exchange are specified in the following sections.

3.7.3. AERO Redirection Message Format

AERO Redirect/Predirect messages use the same format as for ICMPv6 Redirect messages depicted in Section 4.5 of [RFC4861], but also include a new field (the "Prefix Length" field) taken from the Redirect message Reserved field. The Redirect/Prediect messages are formatted as shown in Figure 4:

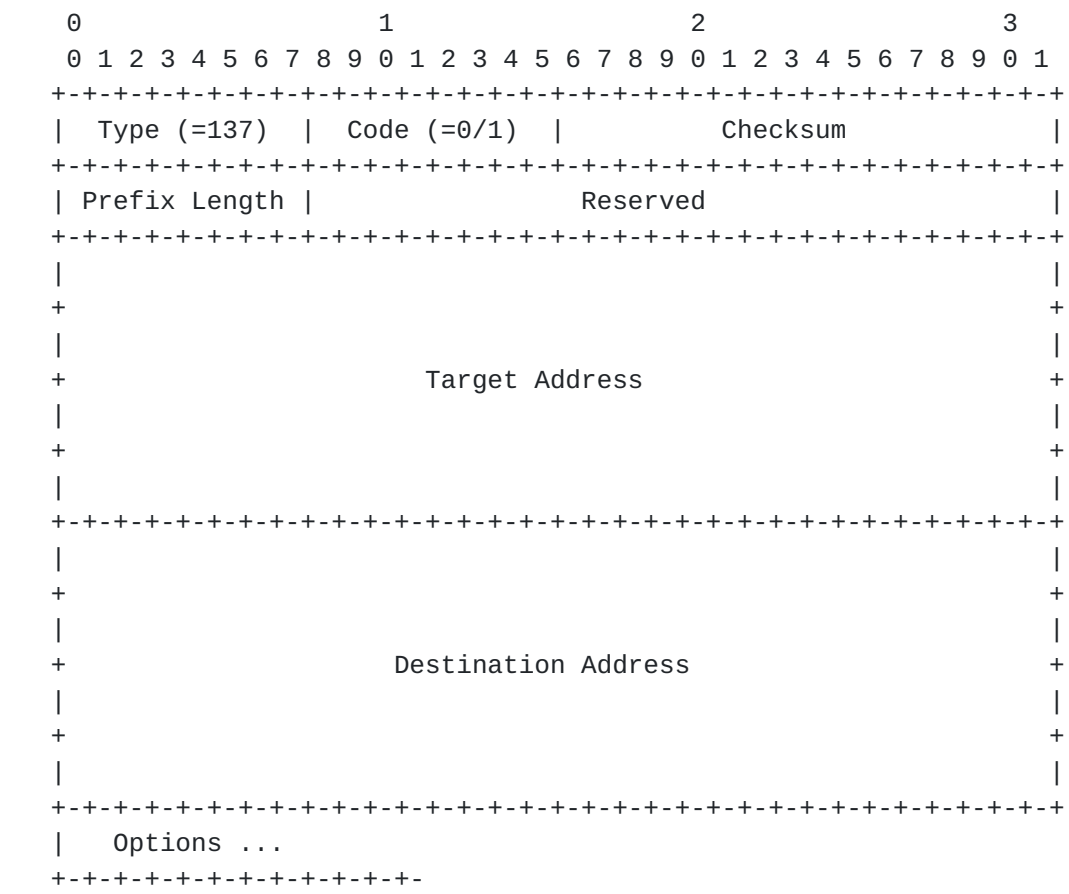

Figure 4: AERO Redirect/Predirect Message Format

3.7.4. Sending Predirects

When an AERO Server forwards a packet out the same AERO interface that it arrived on, the Server sends a Predirect message forward toward the AERO Client nearest the destination instead of sending a Redirect message back to AERO Client nearest the source.

In the reference operational scenario, when Server ('A') forwards a packet sent by Client ('B') toward Client ('D'), it also sends a Predirect message forward toward Client ('D'), subject to rate limiting (see Section 8.2 of [RFC4861]). Server ('A') prepares the Predirect message as follows:

- the link-layer source address is set to 'L2(A)' (i.e., the underlying address of Server ('A')).

- the link-layer destination address is set to 'L2(D)' (i.e., the underlying address of Client ('D')).

- the network-layer source address is set to fe80::1 (i.e., the AERO address of Server ('A')).
- the network-layer destination address is set to fe80::2001:db8:1:0 (i.e., the AERO address of Client ('D')).
- the Type is set to 137.
- the Code is set to 1 to indicate "Predirect".
- the Prefix Length is set to the length of the prefix to be applied to Target address.
- the Target Address is set to fe80::2001:db8:0:0 (i.e., the AERO address of Client ('B')).
- the Destination Address is set to the IPv6 source address of the packet that triggered the Predirection event.
- the message includes a TLLAO set to 'L2(B)' (i.e., the underlying address of Client ('B')).
- the message includes a Redirected Header Option (RHO) that contains the originating packet truncated to ensure that at least the network-layer header is included but the size of the message does not exceed 1280 bytes.

Server ('A') then sends the message forward to Client ('D').

3.7.5. Processing Predirects and Sending Redirects

When Client ('D') receives a Predirect message, it accepts the message only if it has a link-layer source address of the Server, i.e. 'L2(A)'. Client ('D') further accepts the message only if it is willing to serve as a redirection target. Next, Client ('D') validates the message according to the ICMPv6 Redirect message validation rules in Section 8.1 of [RFC4861].

In the reference operational scenario, when the Client ('D') receives a valid Predirect message, it either creates or updates a neighbor cache entry that stores the Target Address of the message as the network-layer address of Client ('B') and stores the link-layer address found in the TLLAO as the link-layer address of Client ('B'). Client ('D') then applies the Preflx Length to the Interface Identifier portion of the Target Address and records the resulting IPv6 prefix in its IPv6 forwarding table. Client ('D') then marks the forwarding table entry as "FILTERING", i.e., the entry is to be used by the network layer for ingress filtering purposes only and not

for forwarding purposes.

After processing the message, Client ('D') prepares a Redirect message response as follows:

- the link-layer source address is set to 'L2(D)' (i.e., the link-layer address of Client ('D')).
- the link-layer destination address is set to 'L2(A)' (i.e., the link-layer address of Server ('A')).
- the network-layer source address is set to 'L3(D)' (i.e., the AERO address of Client ('D')).
- the network-layer destination address is set to 'L3(B)' (i.e., the AERO address of Client ('B')).
- the Type is set to 137.
- the Code is set to 0 to indicate "Redirect".
- the Prefix Length is set to the length of the prefix to be applied to the Target and Destination address.
- the Target Address is set to fe80::2001:db8:1:1 (i.e., the AERO address of Client ('D')).
- the Destination Address is set to the IPv6 destination address of the packet that triggered the Redirection event.
- the message includes a TLLAO set to 'L2(D)' (i.e., the underlying address of Client ('D')).
- the message includes as much of the RHO copied from the corresponding AERO Predirect message as possible such that at least the network-layer header is included but the size of the message does not exceed 1280 bytes.

After Client ('D') prepares the Redirect message, it sends the message to Server ('A').

3.7.6. Forwarding Redirects

When Server ('A') receives a Redirect message, it accepts the message only if it has a neighbor cache entry that associates the message's link-layer source address with the network-layer source address. Next, Server ('A') validates the message according to the ICMPv6 Redirect message validation rules in Section 8.1 of [RFC4861].

Following validation, Server ('A') processes the Redirect, and then forwards a corresponding Redirect on to Client ('B') as follows.

In the reference operational scenario, Server ('A') receives the Redirect message from Client ('D') and prepares to forward a corresponding Redirect message to Client ('B'). Server ('A') then verifies that Client ('D" is authorized to use the Prefix Length in the Redirect message when applied to the AERO address in the network-layer source of the Redirect message, and it discards the message if verification fails. Otherwise, Server ('A') changes the link-layer source address of the message to 'L2(A)', changes the network-layer source address of the message to fe80::1, and changes the link-layer destination address to 'L2(B)' . Server ('A') finally forwards the message to the ingress node ('B') without decrementing the network-layer IPv6 header Hop Limit field.

While not shown in Figure 3, AERO Relays forward Redirect and Predirect messages in exactly this same fashion described above. See Figure 5 in Appendix A for an extension of the reference operational scenario that includes Relays.

3.7.7. Processing Redirects

When Client ('B') receives the Redirect message, it accepts the message only if it has a link-layer source address of the Server, i.e. 'L2(A)'. Next, Client ('B') validates the message according to the ICMPv6 Redirect message validation rules in Section 8.1 of [RFC4861]. Following validation, Client ('B') then processes the message as follows.

In the reference operational scenario, when Client ('B') receives the Redirect message, it either creates or updates a neighbor cache entry that stores the Target Address of the message as the network-layer address of Client ('D') and stores the link-layer address found in the TLLAO as the link-layer address of Client ('D'). Client ('B') then applies the Preflx Length to the Interface Identifier portion of the Target Address and records the resulting IPv6 prefix in its IPv6 forwarding table. Client ('B') then marks the forwarding table entry as "FORWARDING", i.e., the entry is to be used by the network layer for forwarding purposes only and not for ingress filtering purposes..

Now, Client ('B') has an IPv6 forwarding table entry for Client('D')'s prefix in the "FORWARDING" state, and Client ('D') has an IPv6 forwarding table entry for Client ('B')'s prefix in the "FILTERING" state. Therefore, Client ('B') may forward ordinary network-layer data packets directly to the egress node ('D') without forwarding through Server ('A').

To enable packet forwarding in the reverse direction, a separate AERO redirection operation is required that is the mirror-image of the forward operation described above but the link segments traversed in the forward and reverse directions may be different, i.e., the operations are asymmetric.

3.7.8. Neighbor Reachability Considerations

When Client ('B') receives a Redirect message informing it of a direct path to Client ('D'), there is a question in point as to whether Client ('D') can be reached directly without forwarding through Server ('A'). On some AERO links, it may be reasonable for Client ('B') to (optimistically) assume that reachability is transitive, and to immediately begin forwarding data packets to Client ('D') without testing reachability.

On AERO links in which an optimistic assumption of transitive reachability may be unreasonable, however, Client ('B') can defer the redirection until it tests the direct path to the egress node ('D'), e.g., by sending an initial NS message to elicit an NA response. If Client ('B') is unable to elicit a response after MAX_RETRY attempts, it should consider the direct path to Client ('D') to be unusable.

In still other instances, Client ('B') may connect only to an IPvX underlying network, while Client ('D') connects only to an IPvY underlying network. In that case, Client ('B') has no means for reaching Client ('D') directly (since they connect to underlying networks of different IP protocol versions) and so must ignore any Redirects and continue to send packets via Server ('A').

If a direct path between Client ('B') and Client ('D') can be established, the clients can thereafter process any link-layer errors as a hint that the direct path has either failed or has become intermittent.

3.7.9. Neighbor Cache Entry Mainenance

While Client ('B') is actively sending packets to Client ('D'), it should also send NS messages (subject to rate limiting) to keep neighbor cache entries alive and to keep link-layer addresses up to date. If Client ('B') ceases to send packets to Client ('D') for longer than the standard neighbor discover reachability timer, it considers Client ('D') as "unreachable". It then deletes the neighbor cache and IPv6 routing table entries, and allows future packets destined to Client (D')'s EUNs to once again flow through Server ('A') (after which it may eventually receive additional Redirects).

3.7.10. Mobility and Link-Layer Address Change Considerations

When Client ('B') needs to change its link-layer address (e.g., due to a mobility event, due to a change in underlying network interface, etc.), it sends an immediate NS message forward to Client ('D'), which then discovers a new link-layer address.

If both Client ('B') and Client ('D') change their link-layer addresses simultaneously, the NS/NA exchanges between the two neighbors may fail. In that case, the Clients follow the same neighbor unreachability procedures specified in Section 3.7.9.

4. IANA Considerations

There are no IANA actions required for this document.

5. Security Considerations

AERO link security considerations are the same as for standard IPv6 Neighbor Discovery [RFC4861] except that AERO improves on some aspects. In particular, AERO is dependent on a trust basis between AERO Clients and Servers, where the Clients must only engage in the AERO mechanism when it is facilitated by a trusted Server.

AERO links must be protected against link-layer address spoofing attacks in which an attacker on the link pretends to be a trusted neighbor. Links that provide link-layer securing mechanisms (e.g., WiFi networks) and links that provide physical security (e.g., enterprise network LANs) provide a first line of defense that is often sufficient. In other instances, securing mechanisms such as Secure Neighbor Discovery (SeND) [RFC3971] or IPsec [RFC4301] must be used.

6. Acknowledgements

Discussions both on the v6ops list and in private exchanges helped shape some of the concepts in this work. Individuals who contributed insights include Mikael Abrahamsson, Fred Baker, Stewart Bryant, Brian Carpenter, Brian Haberman, Joel Halpern, and Lee Howard. 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.

Earlier works on NBMA tunneling approaches are found in

[RFC2529][RFC5214][RFC5569].

7. References

7.1. Normative References

[I-D.templin-intarea-seal]
Templin, F., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", draft-templin-intarea-seal-65 (work in progress), October 2013.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, December 2011.

7.2. Informative References

[IRON] Templin, F., "The Internet Routing Overlay Network (IRON)", Work in Progress, June 2012.

[RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.

[RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.

Appendix A. AERO Server and Relay Interworking

Figure 3 depicts a reference AERO operational scenario with a single Server on the AERO link. In order to support scaling to larger numbers of nodes, the AERO link can deploy multiple Servers and Relays, e.g., as shown in Figure 5.

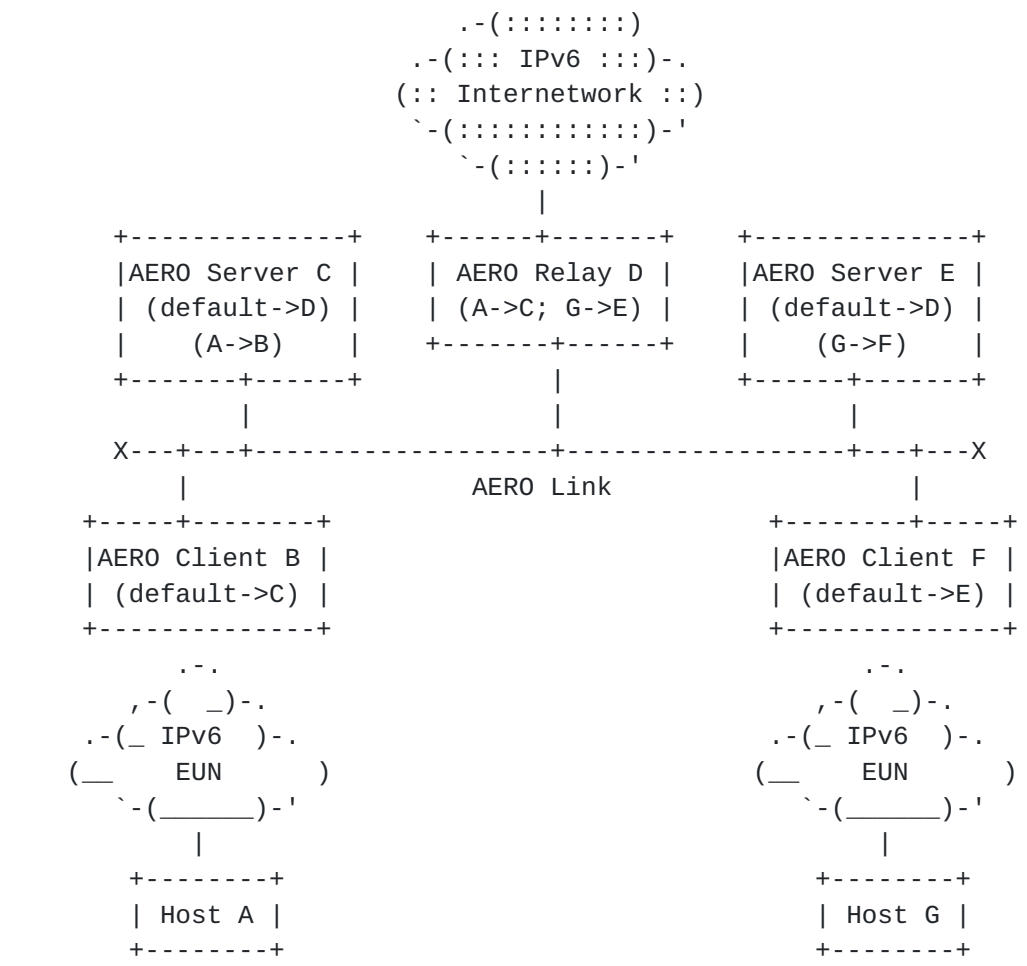

Figure 5: AERO Server/Relay Interworking

In this example, AERO Client ('B') associates with AERO Server ('C'), while AERO Client ('F') associates with AERO Server ('E'). Furthermore, AERO Servers ('C') and ('E') do not associate with each other directly, but rather have an association with AERO Relay ('D') (i.e., a router that has full topology information concerning its associated Servers and their Clients). Relay ('D') connects to the AERO link, and also connects to the native IPv6 Internetwork.

When host ('A') sends a packet toward destination host ('G'), IPv6 forwarding directs the packet through the EUN to Client ('B'), which forwards the packet to Server ('C') in absence of more-specific forwarding information. Server ('C') forwards the packet, and it also generates an AERO Predirect message that is then forwarded through Relay ('D') to Server ('E'). When Server ('E') receives the message, it forwards the message to Client ('F').

After processing the AERO Predirect message, Client ('F') sends an AERO Redirect message to Server ('E'). Server ('E'), in turn,

forwards the message through Relay ('D') to Server ('C'). When Server ('C') receives the message, it forwards the message to Client ('B') informing it that host 'G's EUN can be reached via Client ('F'), thus completing the AERO redirection.

The network layer routing information shared between Servers and Relays must be carefully coordinated in a manner outside the scope of this document. In particular, Relays require full topology information, while individual Servers only require partial topology information (i.e., they only need to know the EUN prefixes associated with their current set of Clients). See [IRON] for an architectural discussion of routing coordination between Relays and Servers..

Author's Address

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

Email: fltemplin@acm.org