

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: December 27, 2012

F. Templin, Ed.  
Boeing Research & Technology  
June 25, 2012

**Asymmetric Extended Route Optimization (AERO)**  
**draft-templin-aero-12.txt**

Abstract

Nodes attached to common multi-access link types (e.g., multicast-capable, shared media, non-broadcast multiple access (NBMA), etc.) can exchange packets as neighbors on the link, but may not always be provisioned with sufficient routing information for optimal neighbor selection. Such nodes should therefore be able to discover a trusted intermediate router on the link that provides both forwarding services to reach off-link destinations and redirection services to inform the node of an on-link neighbor that is closer to the final destination. This redirection can provide a useful route optimization, since the triangular path from the ingress link neighbor, to the intermediate router, and finally to the egress link neighbor may be considerably longer than the direct path from ingress to egress. However, ordinary redirection may lead to operational issues on certain link types and/or in certain deployment scenarios. This document therefore introduces an Asymmetric Extended Route Optimization (AERO) capability that addresses the issues.

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 December 27, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.



## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Terminology . . . . .</a>	<a href="#">6</a>
<a href="#">3.</a>	<a href="#">Motivation . . . . .</a>	<a href="#">7</a>
<a href="#">4.</a>	<a href="#">Example Use Cases . . . . .</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">Requirements . . . . .</a>	<a href="#">9</a>
<a href="#">6.</a>	<a href="#">Asymmetric Extended Route Optimization (AERO) . . . . .</a>	<a href="#">10</a>
<a href="#">6.1.</a>	<a href="#">AERO Link Dynamic Routing . . . . .</a>	<a href="#">10</a>
<a href="#">6.2.</a>	<a href="#">AERO Node Behavior . . . . .</a>	<a href="#">11</a>
<a href="#">6.2.1.</a>	<a href="#">AERO Node Types . . . . .</a>	<a href="#">11</a>
<a href="#">6.2.2.</a>	<a href="#">AERO Host Behavior . . . . .</a>	<a href="#">11</a>
<a href="#">6.2.3.</a>	<a href="#">Edge AERO Router Behavior . . . . .</a>	<a href="#">11</a>
<a href="#">6.2.4.</a>	<a href="#">Intermediate AERO Router Behavior . . . . .</a>	<a href="#">11</a>
<a href="#">6.3.</a>	<a href="#">AERO Reference Operational Scenario . . . . .</a>	<a href="#">12</a>
<a href="#">6.4.</a>	<a href="#">AERO Specification . . . . .</a>	<a href="#">14</a>
<a href="#">6.4.1.</a>	<a href="#">Classical Redirection Approaches . . . . .</a>	<a href="#">14</a>
<a href="#">6.4.2.</a>	<a href="#">AERO Concept of Operations . . . . .</a>	<a href="#">15</a>
<a href="#">6.4.3.</a>	<a href="#">Conceptual Data Structures and Protocol Constants . . . . .</a>	<a href="#">16</a>
<a href="#">6.4.4.</a>	<a href="#">Data Origin Authentication . . . . .</a>	<a href="#">17</a>
<a href="#">6.4.5.</a>	<a href="#">AERO Redirection Message Format . . . . .</a>	<a href="#">18</a>
<a href="#">6.4.6.</a>	<a href="#">Sending Redirects . . . . .</a>	<a href="#">20</a>
<a href="#">6.4.7.</a>	<a href="#">Processing Redirects and Sending Redirects . . . . .</a>	<a href="#">21</a>
<a href="#">6.4.8.</a>	<a href="#">Forwarding Redirects . . . . .</a>	<a href="#">22</a>
<a href="#">6.4.9.</a>	<a href="#">Processing Redirects . . . . .</a>	<a href="#">23</a>
<a href="#">6.4.10.</a>	<a href="#">Sending Periodic Redirect Keepalives . . . . .</a>	<a href="#">24</a>
<a href="#">6.4.11.</a>	<a href="#">Neighbor Reachability Considerations . . . . .</a>	<a href="#">26</a>
<a href="#">6.4.12.</a>	<a href="#">Mobility Considerations . . . . .</a>	<a href="#">26</a>
<a href="#">6.4.13.</a>	<a href="#">Link-Layer Address Change Considerations . . . . .</a>	<a href="#">27</a>
<a href="#">6.4.14.</a>	<a href="#">Prefix Re-Provisioning Considerations . . . . .</a>	<a href="#">28</a>
<a href="#">6.4.15.</a>	<a href="#">Backward Compatibility . . . . .</a>	<a href="#">28</a>
<a href="#">7.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">29</a>
<a href="#">8.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">29</a>
<a href="#">9.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">29</a>
<a href="#">10.</a>	<a href="#">References . . . . .</a>	<a href="#">30</a>
<a href="#">10.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">30</a>
<a href="#">10.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">30</a>
<a href="#">Appendix A.</a>	<a href="#">Intermediate Router Interworking . . . . .</a>	<a href="#">31</a>
	<a href="#">Author's Address . . . . .</a>	<a href="#">33</a>



## 1. Introduction

Nodes attached to common multi-access link types (e.g., multicast-capable, shared media, non-broadcast multiple access (NBMA), etc.) can exchange packets as neighbors on the link, but may not always be provisioned with sufficient routing information for optimal neighbor selection. Such nodes should therefore be able to discover a trusted intermediate router on the link that provides both default forwarding services to reach off-link destinations and redirection services to inform the node of an on-link neighbor that is closer to the final destination.

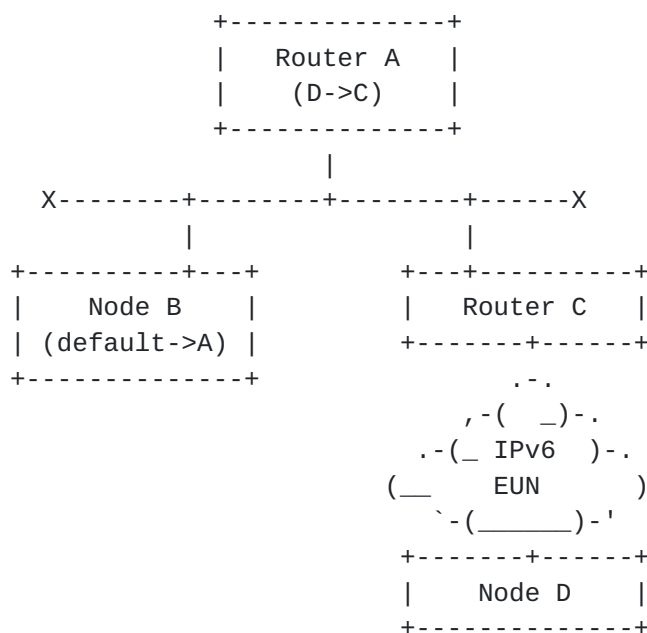


Figure 1: Classical Multi-Access Link Redirection

Figure 1 shows a classical multi-access link redirection scenario. In this figure, Node 'B' is provisioned with only a default route with Router 'A' as the next hop. Router 'A' in turn has a more-specific route that lists Router 'C' as the next hop neighbor on the link for Node 'D's attached End User Network (EUN).

If Node 'B' has a packet to send to Node 'D', 'B' is obliged to send its initial packets via Router 'A'. Router 'A' then forwards the packet to Router 'C' and also returns a redirection message to inform 'B' that 'C' is in fact an on-link neighbor that is closer to the final destination 'D'. After receiving the redirection message, 'B' can place a more-specific route in its forwarding table so that future packets destined to 'D' can be sent directly via Router 'C', as shown in Figure 2.



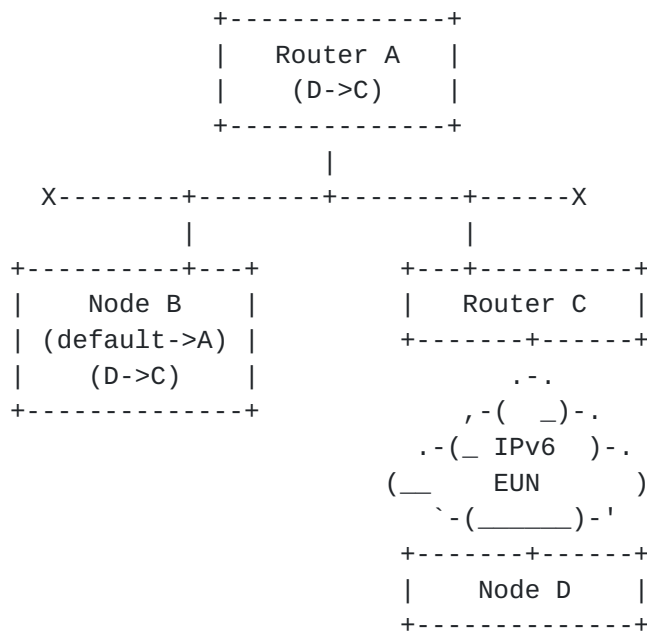


Figure 2: More-Specific Routes Following Redirection

This classical redirection can provide a useful route optimization, since the triangular path from the ingress link neighbor, to the intermediate router, and finally to the egress link neighbor may be considerably longer than the direct path from ingress to egress. However, ordinary redirection may lead to operational issues on certain link types and/or in certain deployment scenarios.

For example, when an ingress link neighbor accepts an ordinary redirection message, it has no way of knowing whether the egress link neighbor is ready and willing to accept packets directly without forwarding through an intermediate router. Likewise, the egress has no way of knowing that the ingress is authorized to forward packets from the claimed network-layer source address. (This is especially important for very large links, since any node on the link can spoof the network-layer source address with low probability of detection even if the link-layer source address cannot be spoofed.) Additionally, the ingress would have no way of knowing whether the direct path to the egress has failed, nor whether the final destination has moved away from the egress to some other network attachment point.

Therefore, a new approach is required that can enable redirection signaling from the egress to the ingress link node under the mediation of a trusted intermediate router. The mechanism is asymmetric (since only the forward direction from the ingress to the egress is optimized) and extended (since the redirection extends forward to the egress before reaching back to the ingress). This





document therefore introduces an Asymmetric Extended Route Optimization (AERO) capability that addresses the issues.

While the AERO mechanisms were initially designed for the specific purpose of NBMA tunnel virtual interfaces (e.g., see: [\[RFC2529\]](#)[\[RFC5214\]](#)[\[RFC5569\]](#)[\[I-D.templin-intarea-vet\]](#)) they can also be applied to any multiple access link types that support redirection. The AERO techniques are discussed herein with reference to IPv6 [\[RFC2460\]](#)[\[RFC4861\]](#)[\[RFC4862\]](#)[\[RFC3315\]](#), however they can also be applied to any other network layer protocol (e.g., IPv4 [\[RFC0791\]](#)[\[RFC0792\]](#)[\[RFC2131\]](#), etc.) that provides a redirection service (details of operation for other network layer protocols are out of scope.)

This document is intended for publication on the experimental track, and therefore does not seek to define a new standard for the Internet. Experimental instead of standards-track is requested since the document proposes a new and different dynamic routing mechanism. Experimentation will focus on candidate multiaccess link types that can connect large numbers of neighboring nodes where the use of existing dynamic routing protocols may be impractical. Examples include NBMA tunnel virtual links, large bridged campus LANs, etc.

## 2. Terminology

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

### AERO link

any link (either physical or virtual) over which the AERO mechanisms can be applied. (For example, a virtual overlay of tunnels can serve as an AERO link.)

### AERO interface

an node's attachment to an AERO link.

### AERO node

a router or host connected to an AERO link, and that participates in the AERO protocol on that link.

### intermediate AERO router ("intermediate router")

a router that configures an advertising router interface on an AERO link over which it can provide default forwarding and redirection services for other AERO nodes.



edge AERO router ("edge router")

a router 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 host

a simple host on an AERO link.

ingress AERO node ("ingress node")

a node that injects packets into an AERO link.

egress AERO node ("egress node")

a node that receives packets from an AERO link.

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [[RFC2119](#)].

### 3. Motivation

AERO was designed to operate as an on-demand route optimization function for nodes attached to a single multi-access link, i.e., similar to the standard ICMPv6 redirect mechanism. However, AERO differs in that the target of the redirection first receives a pre-authorization notification, after which it returns route optimization information to the source of the original packet. This scenario calls into question whether a standard dynamic routing protocol could be used instead of AERO, but a number of considerations indicate that standard routing protocols may be poorly suited for the use cases AERO was designed to address.

First, AERO is designed to work on very large multiple access links that may connect a mix of many thousands of routers and hosts. Classical proactive dynamic routing protocols such as OSPF, IS-IS, RIP, OLSR and TBRPF may be inefficient in such environments due to the control message overhead scaling when large numbers of routers are present and/or when link capacity is low.

Secondly, AERO is designed to work on-demand of data packet arrival, but only seeks to discover neighbors on the same link and not distant nodes that may be located many link hops away. Reactive dynamic routing protocols such as AODV and DSR also operate on-demand, however they flood specialized route discovery messages that reach all nodes on the link and may further traverse multiple link hops before a route reply is received. This requires a multicast-capable network and does not ensure delivery of the original data packet which may be dropped or delayed during route discovery.



Additionally, AERO is designed to override an existing route to a destination if the existing route directs traffic along a sub-optimal path via an extraneous router on the shared link. AERO nodes send data packets over a pre-existing working route, and may subsequently receive notification of a better route based on route optimization feedback from a trusted on-link neighbor. This stands in contrast to on-demand routing protocols that were designed to operate when no pre-existing working routes are present and that multicast explicit route request messages to receive a route reply rather than simply unicast forwarding the data packet via a pre-existing route.

Finally, AERO requires less control message and/or processing overhead than standard dynamic routing protocols on links for which the number of routes that must be maintained by each router is far smaller than the total number of routers on the link, and the routes maintained by each router may be changing over time. For example, on a link that connects  $N$  nodes it will often be the case that each node will only communicate with a small number link neighbors, and the set of neighbors may change dynamically over time. Therefore, the number of active neighbor pairs on the link is  $V*N$  (where  $V$  is a small variable number) instead of  $N^2$ . This is especially important on very large links, e.g., for values of  $N$  such as 1,000 or more.

#### **4. Example Use Cases**

AERO was designed to satisfy numerous operational use cases. As a first example, a hypothetical major airline has deployed an overlay network on top of the global Internet to track the aircraft in its fleet. The global Internet therefore acts as the "link" over which the overlay network is configured. Each aircraft acts as a mobile router that fronts for an internal network that includes various devices controlled and monitored by the airline. However, it would be impractical for each aircraft to track the changing locations of all other aircraft in the fleet due to control message overhead on limited capacity communication links.

In this example, an aircraft 'A' en route to its destination needs to report its ETA and communicate passenger itineraries to other en route aircraft that will be servicing passenger connections. 'A' knows the overlay network addresses of the other aircraft, but does not know the current underlay address mappings. 'A' sends its initial messages targeted to the other aircraft via an airline central dispatch router 'D', which may be located in a far away location. 'D' forwards the messages, but also initiates the AERO redirection procedure to step out of the triangular path and allow direct aircraft-to-aircraft communications.



In a second example, Mobile Ad-hoc Networks (MANETs) are often deployed in environments with a high degree of mobility, attrition, and very limited wireless communications link bandwidth. Such environments typically also require the use of network layer security mechanisms that view the MANET as a "link" over which encrypted messages are forwarded in an overlay network. In such environments, a dynamic routing protocol running in the overlay network may serve to add unacceptable additional congestion to the already overtaxed wireless links. In that case, the AERO route optimization mechanism can eliminate costly extraneous routing hops without imparting additional control message overhead.

In a further example, a large campus LAN that is joined by L2 bridges may connect many thousands of routers and hosts that appear to share a single common multi-access link. In that case, the AERO mechanisms can be applied to satisfy the necessary intra-link route optimization functions without employing an adjunct dynamic routing protocol that may be inefficient for reasons mentioned above.

## 5. Requirements

The route optimization mechanism must satisfy the following requirements:

Req 1: Off-load traffic from performance-critical gateways

The mechanism must offload sustained transit through an intermediate AERO router that would otherwise become a traffic concentrator.

Req 2: Support route optimization

The ingress AERO node should be able to send packets directly to the egress node without forwarding through an intermediate router for route optimization purposes.

Req 3: Support scaling

For scaling purposes, support interworking and control message forwarding between multiple intermediate routers (see [appendix A](#)).

Req 4: Do not circumvent ingress filtering

The mechanism must not open an attack vector where network-layer source address spoofing is enabled even when link-layer source address spoofing is disabled.

Req 5: Do not expose packets to loss due to filtering

The ingress AERO node must have a way of knowing that the egress AERO node will accept its forwarded packets.





Req 6: Do not expose packets to loss due to path failure

The ingress AERO node must have a way of discovering whether the AERO egress node has gone unreachable on the route optimized path.

Req 7: Do not introduce routing loops

Intermediate routers must not invoke a route optimization that would cause a routing loop to form.

Req 8: Support mobility

The mechanism must continue to work even if the final destination node/network moves from a first egress node and re-associates with a second egress node.

Req 9: Support link layer address changes

The mechanism must continue to work even if the Layer 2 addresses of ingress and/or egress AERO nodes change.

Req 10: Support network renumbering

The mechanism must provide graceful transition when an AERO node's attached EUN is renumbered.

## **6. Asymmetric Extended Route Optimization (AERO)**

The following sections specify an Asymmetric Extended Route Optimization (AERO) capability that fulfills the requirements specified in [Section 5](#).

### **6.1. AERO Link Dynamic Routing**

In many AERO link use case scenarios (e.g., small enterprise networks, small and stable MANETs, etc.), routers can engage in a classical dynamic routing protocol so that routing/forwarding tables can be populated and standard forwarding between routers can be used. In other scenarios (e.g., large enterprise/ISP networks, cellular service provider networks, dynamic MANETs, etc.), this might be impractical due to routing protocol control message scaling issues.

When a classical dynamic routing protocol cannot be used, the mechanisms specified in this section can provide a useful on-demand route discovery capability. When both classical dynamic routing protocols and the AERO mechanism are active on the same link, routes discovered by the dynamic routing protocol should take precedence over those discovered by AERO.



## **6.2. AERO Node Behavior**

The following sections discuss characteristics of nodes attached to links over which AERO can be used:

### **6.2.1. AERO Node Types**

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

Edge AERO routers configure their AERO link interfaces as non-advertising router interfaces.

AERO hosts configure their AERO link interfaces as simple host interfaces.

### **6.2.2. AERO Host Behavior**

AERO hosts observe the IPv6 host requirements defined in [\[RFC6434\]](#), except that AERO hosts also engage in the AERO route optimization procedure as specified in [Section 6.4](#).

### **6.2.3. Edge AERO Router Behavior**

Edge AERO routers observe the IPv6 router requirements defined in [\[RFC6434\]](#) except that they act as "hosts" on their non-advertising AERO link router interfaces in the same fashion as for IPv6 CPE routers [\[RFC6204\]](#). Edge routers can then acquire managed prefix delegations aggregated by an intermediate router through the use of, e.g., DHCPv6 Prefix Delegation [\[RFC3633\]](#), administrative configuration, etc.

After the edge router acquires prefixes, it can sub-delegate them to nodes and links within its attached EUNs, then can forward any outbound packets coming from its EUNs via the intermediate router. The edge router also engages in the AERO route optimization procedure as specified in [Section 6.4](#).

### **6.2.4. Intermediate AERO Router Behavior**

Intermediate AERO routers observe the IPv6 router requirements defined in [\[RFC6434\]](#) and respond to RS messages from AERO hosts and edge routers on their advertising AERO link router interfaces by returning an RA message. Intermediate routers further configure a DHCP relay/server function on their AERO links and/or provide an administrative interface for delegation of network-layer addresses



and prefixes.

When the intermediate router completes a stateful network-layer address or prefix delegation transaction (e.g., as a DHCPv6 relay/server, etc.), it establishes forwarding table entries that list the link-layer address of the client AERO node as the link-layer address of the next hop toward the delegated network-layer addresses/prefixes.

When the intermediate router forwards a packet out the same AERO interface it arrived on, it initiates an AERO route optimization procedure as specified in [Section 6.4](#).

### **6.3. AERO Reference Operational Scenario**

Figure 3 depicts the AERO reference operational scenario. The figure shows an intermediate AERO router ('A'), two edge AERO routers ('B', 'D'), an AERO host ('F'), and three ordinary IPv6 hosts ('C', 'E', 'G'):



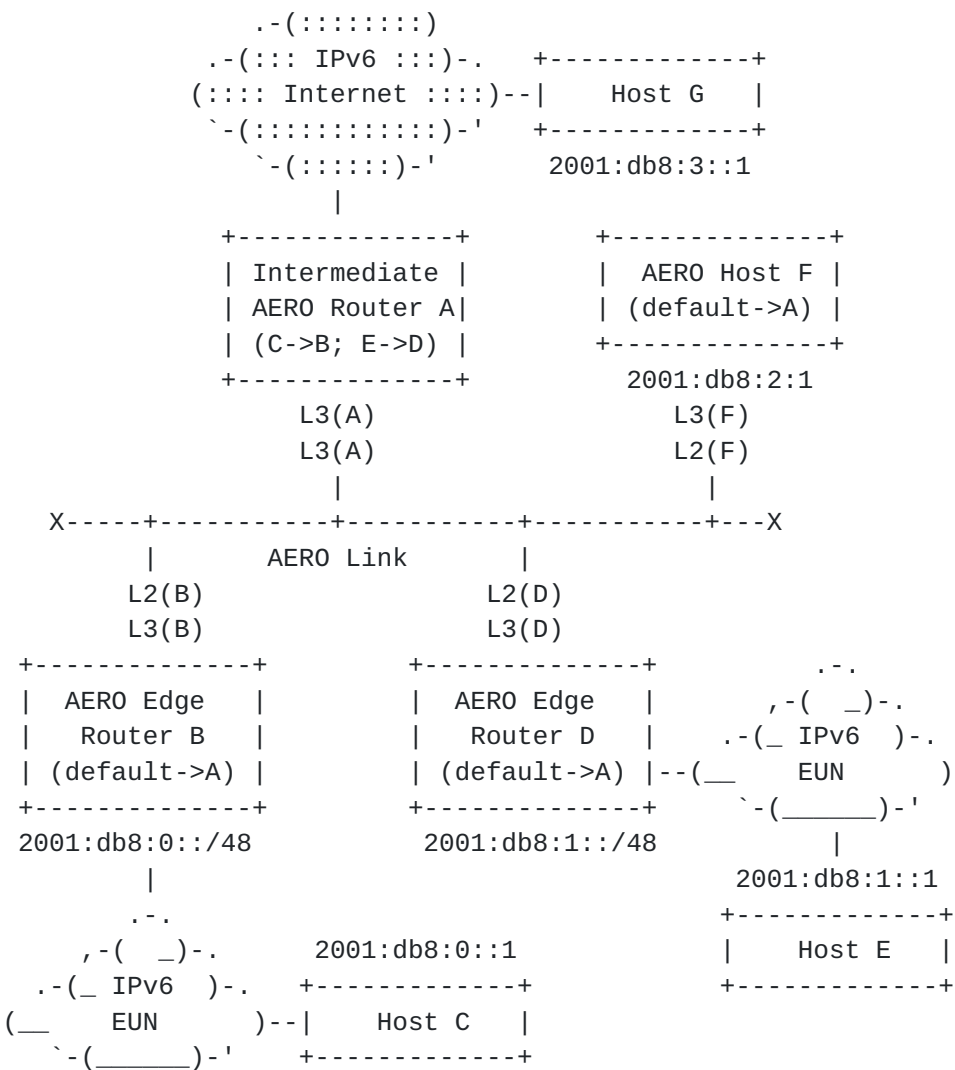


Figure 3: AERO Reference Operational Scenario

In Figure 3, intermediate AERO router ('A') connects to the AERO link and also connects to the IPv6 Internet, either directly or via other IPv6 routers (not shown). Intermediate router ('A') configures an AERO link interface with a link-local network-layer address L3(A) and with link-layer address L2(A). The intermediate router next arranges to add L2(A) to a published list of valid intermediate routers for the link.

AERO node ('B') is an AERO edge router that connects to the AERO link via an interface with link-local network-layer address L3(B) and with link-layer address L2(B). Node ('B') configures a default route with next-hop network-layer address L3(A) via the AERO interface, and also assigns the network-layer prefix 2001:db8:0::/48 to its attached EUN link. IPv6 host ('C') attaches to the EUN, and configures the network-layer address 2001:db8:0::1.





AERO node ('D') is an AERO edge router that connects to the AERO link via an interface with link-local network-layer address L3(D) and with link-layer address L2(D). Node ('D') configures a default route with next-hop network-layer address L3(A) via the AERO interface, and also assigns the network-layer prefix 2001:db8:1::/48 to its attached EUN link. IPv6 host ('E') attaches to the EUN, and configures the network-layer address 2001:db8:1::1.

AERO host ('F') connects to the AERO link via an interface with link-local network-layer address L3(F) and with link-layer address L2(F). Host ('F') configures a default route with next-hop network-layer address L3(A) via the AERO interface, and also assigns the network-layer address 2001:db8:2::1 to the AERO interface.

Finally, IPv6 host ('G') connects to an IPv6 network outside of the AERO link domain. Host ('G') 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.

In these arrangements, intermediate router ('A') must maintain state that associates the delegated network-layer addresses/prefixes with the link-local network-layer addresses of the correct edge routers and/or hosts on the AERO link. The nodes must in turn maintain at least a default route that points to intermediate router ('A'), and can discover more-specific routes either via a proactive dynamic routing protocol or via the AERO mechanisms specified in [Section 6.4](#).

## **6.4. AERO Specification**

[Section 6.3](#) describes the AERO reference operational scenario. We now discuss the operation and protocol details of AERO with respect to this reference scenario.

### **6.4.1. Classical Redirection Approaches**

With reference to Figure 3, when IPv6 source host ('C') sends a packet to an IPv6 destination host ('E'), the packet is first forwarded via the EUN to ingress AERO node ('B'). The ingress node ('B') then forwards the packet over its AERO interface to intermediate router ('A'), which then forwards the packet to egress AERO node ('D'), where the packet is finally forwarded to the IPv6 destination host ('E'). When intermediate router ('A') forwards the packet back out on its advertising AERO interface, it must arrange to redirect ingress node ('B') toward egress node ('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 the ingress and egress nodes are willing participants.



Consider a first alternative in which intermediate router ('A') informs ingress node ('B') only and does not inform egress node ('D') (i.e., "classic redirection"). In that case, the egress node has no way of knowing that the ingress is authorized to forward packets from their claimed source network-layer addresses, and may simply elect to drop the packets. Also, the ingress node has no way of knowing whether the egress is performing some form of source address filtering that would reject packets arriving from a node other than a trusted default router, nor whether the egress is even reachable via a direct path that does not involve the intermediate router. Finally, the ingress node has no way of knowing whether the final destination has moved away from egress node.

Consider also a second alternative in which intermediate router ('A') informs both ingress node ('B') and egress node ('D') separately via independent redirection messages (i.e., "augmented redirection"). In that case, several conditions can occur that could result in communication failures. First, if the ingress receives the redirection message but the egress does not, subsequent packets sent by the ingress could be dropped due to filtering since the egress would not have neighbor state to verify their source network-layer addresses. Second, if the egress receives the redirection message but the ingress does not, subsequent packets sent in the reverse direction by the egress would be lost. Finally, timing issues surrounding the establishment and garbage collection of neighbor state at the ingress and egress 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 neither of these alternatives can satisfy the requirements listed in [Section 5](#), a new redirection technique (i.e., "AERO redirection") is needed.

#### **6.4.2. AERO Concept of Operations**

AERO redirection is used on links for which the classical redirection approaches described in [Section 6.4.1](#) are insufficient to satisfy all requirements. We now discuss the concept of operations for this new approach.

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 ingress node ('B'), which then forwards the packet via its AERO interface to intermediate router ('A').



Using AERO redirection, intermediate router ('A') then forwards the packet out the same AERO interface toward egress node ('D') and also sends a "Predirect" message forward to the egress node as specified in [Section 6.4.6](#). The Predirect message includes the identity of ingress node ('B') as well as information that egress node ('D') can use to determine the longest-match prefixes that cover the source and destination network-layer addresses of the packet that triggered the Predirect. After egress node ('D') receives the Predirect, it processes the message and returns a Redirect message to the intermediate router ('A') as specified in [Section 6.4.7](#). (During the process, it also creates or updates neighbor state for ingress node ('B'), and retains this (src, dst) "prefix pair" as ingress filtering information to accept future packets using addresses matched by the prefixes from ingress node ('B').)

When the intermediate router ('A') receives the Redirect message, it processes the message and forwards it on to ingress node ('B') as specified in [Section 6.4.8](#). The Redirect message includes the identity of egress node ('D') as well as information that ingress node ('B') can use to determine the longest-match prefixes that cover the source and destination network-layer addresses of the packet that triggered the Redirect. After ingress node ('B') receives the Redirect, it processes the message as specified in [Section 6.4.9](#). (During the process, it also creates or updates neighbor state for egress node ('D'), and retains this prefix pair as forwarding information to forward future packets using addresses matched by the prefixes to the egress node ('D').)

Following the above Predirect/Redirect message exchange, forwarding of packets with source and destination network-layer addresses covered by the longest-match prefix pair is enabled in the forward direction from ingress node ('B') to egress node ('D'). The mechanisms that enable this exchange are specified in the following sections.

#### **[6.4.3](#). Conceptual Data Structures and Protocol Constants**

Each AERO node maintains a per AERO interface conceptual neighbor cache that includes an entry for each neighbor it communicates with on the AERO link the same as for any IPv6 interface (see: [\[RFC4861\]](#)).

Each AERO interface neighbor cache entry further maintains two lists of (src, dst) prefix pairs. The AERO node adds a prefix pair to the ACCEPT list if it has been informed by a trusted intermediate router that it is safe to accept packets from the neighbor using network-layer source and destination addresses covered by the prefix pair. The AERO node adds a prefix pair to the FORWARD list if it has been informed by a trusted intermediate router that it is permitted to



forward packets to the neighbor using network-layer addresses covered by the prefix pair.

When the node adds a prefix pair to a neighbor cache entry ACCEPT list, it also sets an expiration timer for the prefix pair to ACCEPT\_TIME seconds. When the node adds a prefix pair to a neighbor cache entry FORWARD list, it also sets an expiration timer for the prefix pair to FORWARD\_TIME seconds. The node further maintains a keepalive interval KEEPALIVE\_TIME used to limit the number of keepalive control messages. Finally, the node maintains a constant value MAX\_RETRY to limit the number of keepalives sent when a neighbor has gone unreachable.

It is RECOMMENDED that FORWARD\_TIME be set to the default constant value 30 seconds to match the default REACHABLE\_TIME value specified for IPv6 neighbor discovery [[RFC4861](#)].

It is RECOMMENDED that ACCEPT\_TIME be set to the default constant value 40 seconds to allow a 10 second window so that the AERO redirection procedure can converge before the ACCEPT\_TIME timer decrements below FORWARD\_TIME.

It is RECOMMENDED that KEEPALIVE\_TIME be set to the default constant value 5 seconds to providing timely reachability verification without causing excessive control message overhead.

It is RECOMMENDED that MAX\_RETRY be set to 3 the same as described for IPv6 neighbor discovery address resolution in [Section 7.3.3 of \[RFC4861\]](#).

Different values for FORWARD\_TIME, ACCEPT\_TIME, KEEPALIVE\_TIME and MAX\_RETRY MAY be administratively set if necessary to better match the AERO link's performance characteristics; however, if different values are chosen all nodes on the link MUST consistently configure the same values. ACCEPT\_TIME SHOULD further be set to a value that is sufficiently longer than FORWARD time to allow the AERO redirection procedure to converge.

#### **[6.4.4.](#) Data Origin Authentication**

AERO nodes MUST employ a data origin authentication check for the packets they receive on an AERO interface. In particular, the node considers the network-layer source address correct for the link-layer source address if at least one of the following is true:

- o the network-layer source address is an on-link address that embeds the link-layer source address, or





- o the network-layer source address is explicitly linked to the link-layer source address through per-neighbor state, or
- o the link-layer source address is the address of a trusted intermediate AERO router.

When the AERO node receives a packet on an AERO interface, it processed the packet further if it satisfies one of these data origin authentication conditions; otherwise it drops the packet.

Note that on links in which link-layer address spoofing is possible, AERO nodes may require additional securing mechanisms. To address this, future work will define a strong data origin authentication scheme such as the use of digital signatures.

#### **6.4.5. AERO Redirection Message Format**

AERO redirection messages use the same format as for ICMPv6 Redirect messages depicted in [Section 4.5 of \[RFC4861\]](#), however the messages are encapsulated in a UDP header [\[RFC0768\]](#) to distinguish them from ordinary ICMPv6 Redirect messages. AERO Redirect messages therefore require a new UDP service port number 'AERO\_PORT'.

The AERO redirection message is formatted as shown in Figure 4:



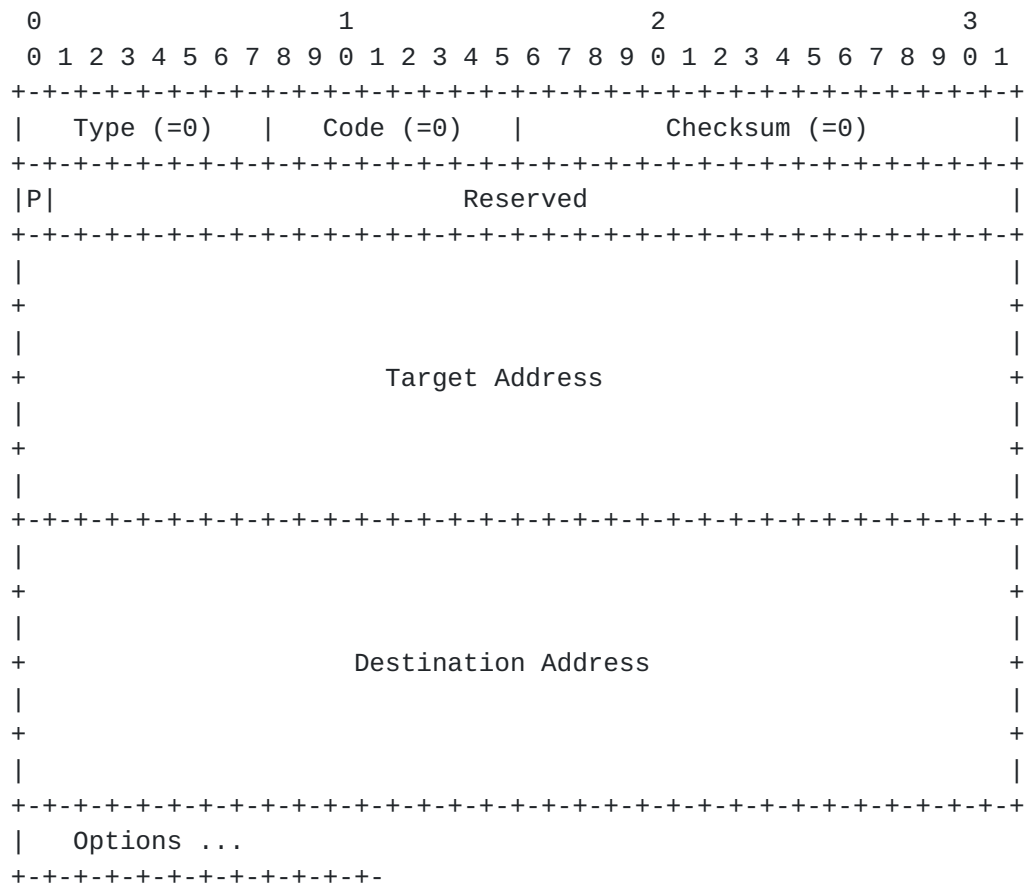


Figure 4: AERO Redirection Message Format

The AERO redirection message sender sets the 'Type' field to 0 (since this is not an actual ICMPv6 message), and also sets the 'Checksum' field to 0 (since the UDP checksum will provide protection for the entire packet). The sender further sets the 'P' bit to 1 if this is a 'Predirect' message and sets the 'P' bit to 0 if this is a 'Redirect' message (as described below).

The sender then encapsulates the AERO Redirect message in IP/UDP headers as shown in Figure 5:



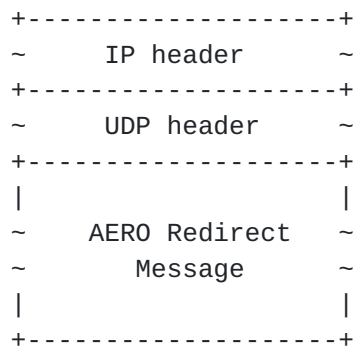


Figure 5: AERO Message UDP Encapsulation Format

The AERO redirection message sender sets the UDP destination port number to 'AERO\_PORT' and sets the UDP source port number to a (pseudo-)random value. The sender next sets the UDP length field to the length of the UDP message, then calculates the checksum across the message and writes the value into the UDP checksum field. Next, the sender sets the IP TTL/Hop-limit field to a small integer value chosen to provide a quick exit from any temporal routing loops. It is RECOMMENDED that the sender set IP TTL/Hop-limit to the value 8 unless it has better knowledge of the AERO link characteristics.

#### 6.4.6. Sending Predirects

When an intermediate AERO router forwards a packet out the same AERO interface that it arrived on, the router sends an AERO Predirect message forward toward the egress AERO node instead of sending an ICMPv6 Redirect message back to the ingress AERO node.

In the reference operational scenario, when the intermediate router ('A') forwards a packet sent by the ingress node ('B') toward the egress node ('D'), it also sends an AERO Predirect message forward toward the egress, subject to rate limiting (see [Section 8.2 of \[RFC4861\]](#)). The intermediate router ('A') prepares the AERO Predirect message as follows:

- o the link-layer source address is set to 'L2(A)' (i.e., the link-layer address of the intermediate router).
- o the link-layer destination address is set to 'L2(D)' (i.e., the link-layer address of the egress node).
- o the network-layer source address is set to 'L3(A)' (i.e., the link-local network-layer address of the intermediate router).
- o the network-layer destination address is set to 'L3(D)'. (i.e., the link-local network-layer address of the egress node).



- o the UDP destination port is set to 'AERO\_PORT'.
- o the Target and Destination Addresses are both set to 'L3(B)' (i.e., the link-local network-layer address of the ingress node).
- o on links that require stateful address mapping, the message includes a Target Link Layer Address Option (TLLO) set to 'L2(B)' (i.e., the link-layer address of the ingress node).
- o the message includes a Route Information Option (RIO) [[RFC4191](#)] that encodes the ingress node's network-layer address/prefix delegation that covers the network-layer source address of the originating packet.
- o 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.
- o the 'P' bit is set to P=1.

The intermediate router ('A') then sends the message forward to the egress node ('D').

#### **6.4.7. Processing Predirects and Sending Redirects**

When the egress node ('D') receives an AERO Redirect message, it accepts the message only if it satisfies the data origin authentication requirements specified in [Section 6.4.4](#). The egress further accepts the message only if it is willing to serve as a redirection target.

Next, the egress node ('D') validates the message according to the ICMPv6 Redirect message validation rules in [Section 8.1 of \[RFC4861\]](#) with the exception that the message includes a Type value of 0, a Checksum value of 0 and a link-local address in the ICMP destination field that differs from the destination address of the packet header encapsulated in the RHO.

In the reference operational scenario, when the egress node ('D') receives a valid AERO Redirect message it either creates or updates a neighbor cache entry that stores the Target address of the message (i.e., the link-local network-layer address of the ingress node ('B')). The egress node ('D') then records the prefix found in the RIO along with its own prefix that matches the network-layer destination address in the packet header found in the RHO with the neighbor cache entry as an acceptable (src, dst) prefix pair. The egress node ('D') then adds the prefix pair to the neighbor cache





entry ACCEPT list, and sets/resets an expiration timer for the prefix pair to ACCEPT\_TIME seconds. If the timer later expires, the egress node ('D') deletes the prefix pair.

After processing the message, the egress node ('D') prepares an AERO Redirect message response as follows:

- o the link-layer source address is set to 'L2(D)' (i.e., the link-layer address of the egress node).
- o the link-layer destination address is set to 'L2(A)' (i.e., the link-layer address of the intermediate router).
- o the network-layer source address is set to 'L3(D)' (i.e., the link-local network-layer address of the egress node).
- o the network-layer destination address is set to 'L3(B)' (i.e., the link-local network-layer address of the ingress node).
- o the UDP destination port is set to 'AERO\_PORT'.
- o the Target and the Destination Addresses are both set to 'L3(D)' (i.e., the link-local network-layer address of the egress node).
- o on links that require stateful address mapping, the message includes a Target Link Layer Address Option (TLLO) set to 'L2(D)'.
- o the message includes an RIO that encodes the egress node's network-layer address/prefix delegation that covers the network-layer destination address of the originating packet.
- o the message includes as much of the RHO copied from the corresponding AERO Redirect message as possible such that at least the network-layer header is included but the size of the message does not exceed 1280 bytes.
- o the 'P' bit is set to P=0.

After the egress node ('D') prepares the AERO Redirect message, it sends the message to the intermediate router ('A').

#### **6.4.8. Forwarding Redirects**

When the intermediate router ('A') receives an AERO Redirect message, it accepts the message only if it satisfies the data origin authentication requirements specified in [Section 6.4.4](#). Next, the intermediate router ('A') validates the message the same as described



in [Section 6.4.7](#). Following validation, the intermediate router ('A') processes the Redirect, and then forwards a corresponding Redirect on to the ingress node ('B') as follows.

In the reference operational scenario, the intermediate router ('A') receives the AERO Redirect message from the egress node ('D') and prepares to forward a corresponding Redirect message to the ingress node ('B'). The intermediate router ('A') then verifies that the RIO encodes a network-layer address/prefix that the egress node ('D') is authorized to use, and discards the message if verification fails. Otherwise, the intermediate router ('A') changes the link-layer source address of the message to 'L2(A)', changes the network-layer source address of the message to the link-local network-layer address 'L3(A)', and changes the link-layer destination address to 'L2(B)'. The intermediate router ('A') finally decrements the IP TTL/Hop-limit and forwards the message to the ingress node ('B').

#### **6.4.9. Processing Redirects**

When the ingress node ('B') receives an AERO Redirect message (i.e., one with P=0), it accepts the message only if it satisfies the data origin authentication requirements specified in [Section 6.4.4](#). Next, the ingress node ('B') validates the message the same as described in [Section 6.4.6](#). Following validation, the ingress node ('B') then processes the message as follows.

In the reference operational scenario, when the ingress node ('B') receives the AERO Redirect message it either creates or updates a neighbor cache entry that stores the Target address of the message (i.e., the link-local network-layer address of the egress node 'L3(D)'). The ingress node ('B') then records the (src, dst) prefix pair associated with the triggering packet in the neighbor cache entry FORWARD list, i.e., it records its prefix that matches the redirected packet's network-layer source address and the prefix listed in the RIO as the prefix pair. The ingress node ('B') then sets/resets an expiration timer for the prefix pair to FORWARD\_TIME seconds. If the timer later expires, the ingress node ('B') deletes the entry.

Now, the ingress node ('B') has a neighbor cache FORWARD list entry for the prefix pair, and the egress node ('D') has a neighbor cache ACCEPT list entry for the prefix pair. Therefore, the ingress node ('B') may forward ordinary network-layer data packets with network-layer source and destination addresses that match the prefix pair directly to the egress node ('D') without forwarding through the intermediate router ('A'). Note that the ingress node must have a way of informing the network layer of a route that associates the destination prefix with this neighbor cache entry. The manner of



establishing such a route (and deleting it when it is no longer necessary) is left to the implementation.

To enable packet forwarding in the reverse direction, a separate AERO redirection operation is required which 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.

#### **6.4.10. Sending Periodic Redirect Keepalives**

In order to prevent prefix pairs from expiring while data packets are actively flowing, the ingress node ('B') can send AERO Redirect keepalive messages directly to the egress node ('D') to solicit AERO Redirect messages. The node should send a keepalive message only when a data packet covered by the prefix pair has been sent recently, and should wait for at least KEEPALIVE\_TIME seconds before sending each successive keepalive message in order to limit control message overhead.

In the reference operational scenario, when the ingress node ('B') needs to refresh the FORWARD timer for a specific prefix pair it can send an AERO Redirect keepalive message directly to the egress node ('D') prepared as follows:

- o the link-layer source address is set to 'L2(B)' (i.e., the link-layer address of the ingress node).
- o the link-layer destination address is set to 'L2(D)' (i.e., the link-layer address of the egress node).
- o the network-layer source address is set to 'L3(B)' (i.e., the link-local network-layer address of the ingress node).
- o the network-layer destination address is set to 'L3(D)' (i.e., the link-local network-layer address of the egress node).
- o the UDP destination port is set to 'AERO\_PORT'.
- o the Redirect Target and Destination Addresses are both set to 'L3(B)' (i.e., the link-local network-layer address of the ingress node).
- o the Redirect message includes an 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.



- o the 'P' bit is set to P=1.

When the egress node ('D') receives the AERO Redirect message, it validates the message the same as described in [Section 6.4.6](#).

Following validation, the egress node ('D') then resets its ACCEPT timer for the prefix pair that matches the originating packet's network-layer source and destination addresses to ACCEPT\_TIME seconds, and sends an AERO Redirect message directly to the ingress node ('B') prepared as follows:

- o the link-layer source address is set to 'L2(D)' (i.e., the link-layer address of the egress node).
- o the link-layer destination address is set to 'L2(B)' (i.e., the link-layer address of the ingress node).
- o the network-layer source address is set to 'L3(D)' (i.e., the link-local network-layer address of the egress node).
- o the network-layer destination address is set to 'L3(B)' (i.e., the link-local network-layer address of the ingress node).
- o the UDP destination port is set to 'AERO\_PORT'.
- o the Redirect Target and Destination Addresses are both set to 'L3(D)' (i.e., the link-local network-layer address of the egress node).
- o the message includes as much of the RHO copied from the corresponding AERO Redirect message as possible such that at least the network-layer header is included but the size of the message does not exceed 1280 bytes.
- o the 'P' bit is set to P=0.

When the ingress node ('B') receives the AERO Redirect message, it validates the message the same as described in [Section 6.4.6](#).

Following validation, the ingress node ('B') then resets its FORWARD timer for the prefix pair that matches the originating packet's network-layer source and destination addresses to FORWARD\_TIME seconds.

In this process, if the ingress node sends MAX\_RETRY Redirect keepalive messages without receiving a Redirect reply it can either declare the prefix pair unreachable immediately or allow the pair to expire after FORWARD\_TIME seconds.





#### **6.4.11. Neighbor Reachability Considerations**

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

On AERO links in which an optimistic assumption of transitive reachability may be unreasonable, however, the ingress node ('B') can defer the redirection until it tests the direct path to the egress node ('D'), e.g., by sending an IPv6 Neighbor Solicitation to elicit an IPv6 Neighbor Advertisement response. If the ingress node ('B') is unable to elicit a response after MAX\_RETRY attempts, it should consider the direct path to the egress node ('D') as unusable.

In either case, the ingress node ('B') can process any link errors corresponding to the data packets sent directly to the egress node ('D') as a hint that the direct path has either failed or has become intermittent. Conversely, the ingress node ('B') can further process any Redirect messages received as evidence of neighbor reachability.

#### **6.4.12. Mobility Considerations**

Again with reference to Figure 3, egress node ('D') can configure both a non-advertising router interface on a provider AERO link and advertising router interfaces on its connected EUN links. When an EUN node ('E') in one of the egress node's connected EUNs moves to a different network point of attachment, however, it can release its network-layer address/prefix delegations that were registered with egress node ('D') and re-establish them via a different router.

When the EUN node ('E') releases its network-layer address/prefix delegations, the egress node ('D') marks its forwarding table entries corresponding to the network-layer addresses/prefixes as "departed" and no longer responds to AERO Redirect keepalive messages for the departed addresses/prefixes. When egress node ('D') receives packets from an ingress node ('B') with network-layer source and destination addresses that match a prefix pair on the ACCEPT list, it forwards them to the last-known link-layer address of EUN node ('E') as a means for avoiding mobility-related packet loss during routing changes. Egress node ('D') also returns a NULL AERO Redirect message to inform the ingress node ('B') of the departure. The message is prepared as follows:



- o the link-layer source address is set to 'L2(D)'.
- o the link-layer destination address is set to 'L2(B)'.
- o the network-layer source address is set to the link-local address 'L3(D)'.
- o the network-layer destination address is set to the link-local address 'L3(B)'.
- o the UDP destination port is set to 'AERO\_PORT'.
- o the Redirect Target and Destination Addresses are both set to NULL.
- o the message includes an RHO that contains as much of the original packet as possible such that at least the network-layer header is included but the size of the message does not exceed 1280 bytes.
- o the 'P' bit is set to P=0.

When ingress node ('B') receives the NULL AERO Redirect message, it deletes the prefix pair associated with the packet in the RHO from its list of forwarding entries corresponding to egress node ('D'). When egress node ('D')s ACCEPT\_TIME timer for the prefix pair corresponding to the departed prefix expires, it deletes the prefix pairs from its list of ingress filtering entries corresponding to ingress node ('B').

Eventually, any such correspondent AERO nodes will receive a NULL AERO Redirect message and will cease to use the egress node ('D') as a next hop. They will then revert to sending packets destined to the EUN node ('E') via a trusted intermediate router and may subsequently receive new AERO Redirect messages to discover that the EUN node ('E') is now associated with a new AERO edge router.

Note that any packets forwarded by the egress node ('D') via a departed forwarding table entry may be lost if the (mobile) EUN node ('E') moves off-link with respect to its previous EUN point of attachment. This should not be a problem for large links (e.g., large cellular network deployments, large ISP networks, etc.) in which all/most mobility events are intra-link.

#### **6.4.13. Link-Layer Address Change Considerations**

When an ingress node needs to change its link-layer address, it deletes each FORWARD list entry that was established under the old link layer address, changes the link layer address, then allows



packets to again flow through an intermediate router. Any egress node that receives the packets will also receive new Redirect messages from the intermediate router. The egress node then deletes the ACCEPT entry that included the ingress node's old link-layer address and installs a new ACCEPT entry that includes the ingress node's new link-layer address. The egress then returns a new Redirect message to the ingress node via the intermediate router, which the ingress node uses to establish a new FORWARD list entry.

When an egress node needs to change its link-layer address, it deletes each entry in the ACCEPT list and SHOULD also send NULL AERO Redirect messages to the corresponding ingress node (i.e., the same as described for mobility operations in [Section 6.4.12](#)) before changing the link-layer address. Any ingress node that receives the NULL Redirect messages will delete any corresponding FORWARD list entries and again allow packets to flow through an intermediate router. The egress then changes the link-layer address, and sends new Redirect messages in response to any Redirect messages it receives from the intermediate router while using the new link-layer address.

#### **[6.4.14.](#) Prefix Re-Provisioning Considerations**

When an AERO node configures one or more FORWARD/ACCEPT list prefix pair entries, and the prefixes associated with the pair are somehow re-configured or renumbered, the stale FORWARD/ACCEPT list information must be deleted.

When an ingress node ('B') re-configures its network-layer source prefix in such a way that the ACCEPT list entry in the egress node ('D') would no longer be valid (e.g., the prefix length of the source prefix changes), the ingress node ('B') simply deletes the prefix pair from its FORWARD list and allows subsequent packets to again flow through an intermediate router ('A').

When the egress node ('D') re-configures its network-layer destination prefix in such a way that the FORWARD list entry in the ingress node ('B') would no longer be valid, the egress node ('D') sends a NULL AERO Redirect message to the ingress node ('B') the same as described for mobility and link-layer address change considerations when it receives either an AERO Redirect message or a data packet (subject to rate limiting) from the ingress node ('B').

#### **[6.4.15.](#) Backward Compatibility**

There are no backward compatibility considerations since AERO redirection messages use a new UDP port number that distinguishes them from other kinds of control messages. Therefore, legacy nodes



will simply discard any AERO redirection messages they may accidentally receive.

Note however that AERO redirection requires that all three of the ingress, intermediate router and egress participate in the protocol. Additionally, the intermediate router SHOULD disable ordinary ICMPv6 Redirects when AERO redirection is enabled.

## **7. IANA Considerations**

IANA has assigned a UDP user port number TBD for this protocol via the expert review process [[RFC5226](#)].

## **8. 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 edge nodes and intermediate routers, where the edge nodes must only engage in the AERO mechanism when it is facilitated by a trusted intermediate router.

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, sufficient assurances against link-layer address spoofing attacks are possible if the source can digitally sign its messages through means outside the scope of this document.

## **9. 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.

## **10. References**





### **10.1. Normative References**

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), August 1980.
- [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.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", [RFC 4191](#), November 2005.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", [RFC 4443](#), March 2006.
- [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.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", [RFC 6434](#), December 2011.

### **10.2. Informative References**

- [I-D.templin-intarea-vet]  
Templin, F., "Virtual Enterprise Traversal (VET)", [draft-templin-intarea-vet-33](#) (work in progress), December 2011.
- [I-D.templin-ironbis]  
Templin, F., "The Internet Routing Overlay Network (IRON)", [draft-templin-ironbis-10](#) (work in progress), December 2011.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5,



[RFC 792](#), September 1981.

- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [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.
- [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., Beebe, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", [RFC 6204](#), April 2011.

## **[Appendix A](#). Intermediate Router Interworking**

Figure 3 depicts a reference AERO operational scenario with a single intermediate router on the AERO link. In order to support scaling to larger numbers of nodes, the AERO link can deploy multiple intermediate routers, e.g., as shown in Figure 6



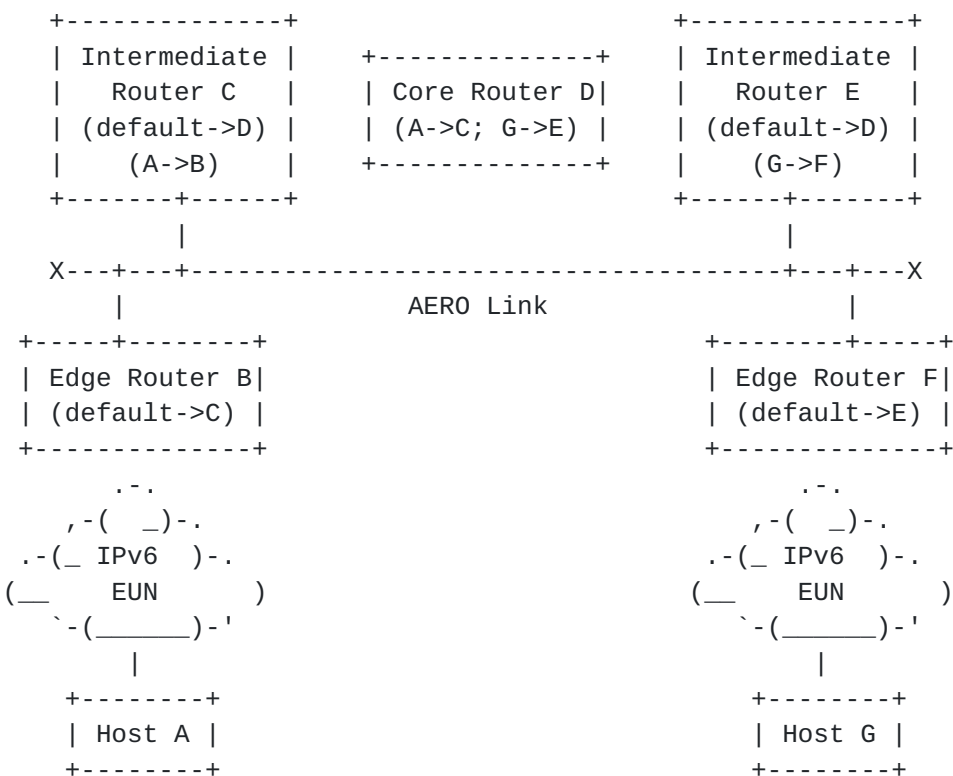


Figure 6: Multiple Intermediate Routers

In this example, the ingress AERO node ('B') (in this case an edge router, but could also be a host) associates with intermediate AERO router ('C'), while the egress AERO node ('F') (in this case an edge router, but could also be a host) associates with intermediate AERO router ('E'). Furthermore, intermediate routers ('C') and ('E') do not associate with each other directly, but rather have an association with a "core" router ('D') (i.e., a router that has full topology information concerning its associated intermediate routers). Core router 'D' may connect to either the AERO link, or to other physical or virtual links (not shown) to which intermediate routers 'C' and 'E' also connect.

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

After processing the AERO Redirect message, egress router ('F') sends an AERO Redirect message to intermediate router ('E').



Intermediate router ('E') in turn forwards the message through core router ('D') to intermediate router ('C'). When intermediate router ('C') receives the Redirect, it forwards the message to ingress edge router ('B') informing it that host 'G's EUN can be reached via egress router 'F', thus completing the AERO redirection.

The interworkings between intermediate and core routers (including the conveyance of pseudo Predirects and Redirects) must be carefully coordinated in a manner outside the scope of this document. In particular, the intermediate and core routers must ensure that any routing loops that may be formed are temporal in nature. See [[I-D.templin-ironbis](#)] for an architectural discussion of coordinations between intermediate and core routers.

#### Author's Address

Fred L. Templin (editor)  
Boeing Research & Technology  
P.O. Box 3707 MC 7L-49  
Seattle, WA 98124  
USA

Email: [fltemplin@acm.org](mailto:fltemplin@acm.org)



