Internet Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

J. Chroboczek
IRIF, University of Paris
W. Kumari
Google, LLC
T. H淡iland-J淡rgensen
Red Hat
7 July 2025

# IPv4 routes with an IPv6 next hop
draft-ietf-intarea-v4-via-v6-01

## Abstract

This document proposes "v4-via-v6" routing, a technique that uses IPv6 next-hop addresses for routing IPv4 packets, thus making it possible to route IPv4 packets across a network where routers have not been assigned IPv4 addresses. The document both describes the technique, as well as discussing its operational implications.

## About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://wkumari.github.io/draft-chroboczek-intarea-v4-via-v6/draft-ietf-intarea-v4-via-v6.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-intarea-v4-via-v6/.

Discussion of this document takes place on the Internet Area Working Group Working Group mailing list (mailto:int-area@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/int-area/. Subscribe at https://www.ietf.org/mailman/listinfo/int-area/.

Source for this draft and an issue tracker can be found at https://github.com/wkumari/draft-chroboczek-intarea-v4-via-v6.

## Status of This Memo

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

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

## Copyright Notice



## Table of Contents

## 1. Introduction

The dominant form of routing in the Internet is next-hop routing, where a routing protocol constructs a routing table which is used by a forwarding process to forward packets. The routing table is a data structure that maps network prefixes in a given family (IPv4 or IPv6) to next hops, pairs of an outgoing interface and a neighbor's network address, for example:

| destination | next hop |
|---|---|
| 2001:db8:0:1::/64 | eth0, fe80::1234:5678 |
| 203.0.113.0/24 | eth0, 192.0.2.1 |

When a packet is routed according to a given routing table entry, the forwarding plane uses a neighbor discovery protocol (the Neighbor Discovery protocol (ND) [RFC4861] in the case of IPv6, the Address Resolution Protocol (ARP) [RFC0826] in the case of IPv4) to map the next-hop address to a link-layer address (a "MAC address"), which is then used to construct the link-layer frames that encapsulate forwarded packets.

It is apparent from the description above that there is no fundamental reason why the destination prefix and the next-hop address should be in the same address family: there is nothing preventing an IPv6 packet from being routed through a next hop with an IPv4 address (in which case the next hop's MAC address will be obtained using ARP), or, conversely, an IPv4 packet from being routed through a next hop with an IPv6 address. (In fact, it is even possible to store link-layer addresses directly in the next-hop entry of the routing table, thus avoiding the use of an address resolution protocol altogether, which is commonly done in networks using the OSI protocol suite.)

This document focuses on the specific case of routing IPv4 packets through an IPv6 next-hop. This case is particularly interesting, since it makes it possible to build networks that have no IPv4 addresses except at the edges and still provide IPv4 connectivity to edge hosts. In addition, since an IPv6 next hop can use a link-local address that is autonomously configured, the use of such routes enables a mode of operation where the network core has no statically assigned IP addresses of either family, which significantly reduces the amount of manual configuration required. (See also [RFC7404] for a discussion of the issues involved with such an approach.)

We call a route towards an IPv4 prefix that uses an IPv6 next hop a "v4-via-v6" route. V4-via-v6 routing is not restricted to routers, and could usefully be applied to hosts, although doing so would require solving the issue of host configuration, for example by extending either DHCPv4 or DHCPv6 to publish an IPv4 default route with an IPv6 next hop.

[RFC8950] discusses advertising of IPv4 NLRI with a next-hop address that belongs to the IPv6 protocol, but confines itself to how this is carried and advertised in the BGP protocol. This document, on the other hand, discusses the concept of v4-via-v6 routes independently of any specific routing protocol, their design and operational considerations, and the implications of using them.

{ Editor note, to be removed before publication. This document is heavily based on draft-ietf-babel-v4viav6. When draft-ietf-babel-v4viav6 was going through IESG eval, Warren raised concerns that something this fundamental deserved to be documented in a separate, standalone document, so that it can be more fully discussed, and, more importantly, referenced cleanly in the future.}

## 2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Operation

Next-hop routing is implemented by two separate components, the routing protocol and the forwarding plane, that communicate through a shared data structure, the routing table.

### 3.1. Structure of the routing table

The routing table is a data structure that maps address prefixes to next-hops, pairs of the form (interface, address). In traditional next-hop routing, the routing table maps IPv4 prefixes to IPv4 next hops, and IPv6 addresses to IPv6 next hops. With v4-via-v6 routing, the routing table is extended so that an IPv4 prefix may map to an IPv6 next hop.

Resolution may be recursive: the next-hop may itself be a prefix that requires further resolution to map to the outgoing interface and L2 address. V4-via-v6 routing does not prevent recursive resolution.

### 3.2. Operation of the forwarding plane

The forwarding plane is the part of the routing implementation that is executed for every forwarded packet. As a packet arrives, the forwarding plane consults the routing table, selects a single route matching the packet, determines the next-hop address, and forwards the packet to the next-hop address.

With v4-via-v6 routing, the address family of the next-hop address is no longer determined by the address family of the prefix: since the routing table may map an IPv4 prefix to either an IPv4 or an IPv6 next-hop, the forwarding plane must be able to determine, on a per-packet basis, whether the next-hop address is an IPv4 or an IPv6 address, and to use that information in order to choose the right address resolution protocol to use (ARP for IP4, ND for IPv6).

### 3.3. Operation of routing protocols

The routing protocol is the part of the routing implementation that is executed asynchronously from the forwarding plane, and whose role is to build the routing table. Since v4-via-v6 routing is a generalization of traditional next-hop routing, v4-via-v6 can interoperate with existing routing protocols: a traditional routing protocol produces a traditional next-hop routing table, which can be used by an implementation supporting v4-via-v6 routing.

However, in order to use the additional flexibility provided by v4-via-v6 routing, routing protocols need to be extended with the ability to populate the routing table with v4-via-v6 routes when an IPv4 address is not available or when the available IPv4 addresses are not suitable for use as a next-hop (for example not stable enough).

Some protocols already support the advertisement of IPv4 routes with an IPv6 next-hop, including Babel [RFC8966] and BGP [RFC8950]. Other protocol advertise both IPv4 and IPv6 prefixes over a single neighbor; these include: * Multi-Topology (MT) Routing in OSPF ([RFC4915]) * Multi-Topology (MT) Routing in IS-IS ([RFC5120]) While both of these employ a common control plane, they use separate data planes, and therefore don’t implement v4-via-v6 routing.

## 4. ICMP Considerations

The Internet Control Message Protocol (ICMPv4, or simply ICMP) [RFC0792] is a protocol related to IPv4 that is primarily used to carry diagnostic and debugging information. ICMPv4 packets may be originated by end hosts (e.g., the "destination unreachable, port unreachable" ICMPv4 packet), but they may also be originated by intermediate routers (e.g., most other kinds of "destination unreachable" packets).

Some protocols deployed in the Internet rely on ICMPv4 packets sent by intermediate routers. Most notably, path MTU Discovery (PMTUd) [RFC1191] is an algorithm executed by end hosts to discover the maximum packet size that a route is able to carry. While there exist variants of PMTUd that are purely end-to-end [RFC4821], the variant most commonly deployed in the Internet has a hard dependency on ICMPv4 packets originated by intermediate routers: if intermediate routers are unable to send ICMPv4 packets, PMTUd may lead to persistent black-holing of IPv4 traffic.

Due to this kind of dependency, every router that is able to forward IPv4 traffic SHOULD be able originate ICMPv4 traffic. Since the extension described in this document enables routers to forward IPv4 traffic received over an interface that has not been assigned an IPv4 address, a router implementing this extension MUST be able to originate ICMPv4 packets even when the outgoing interface has not been assigned an IPv4 address.

In such a situation, if the router has an interface that has been assigned a publicly routable IPv4 address (other than the loopback address), or if an IPv4 address has been assigned to the router itself (to the "loopback interface"), then that IPv4 address may be used as the source of originated ICMPv4 packets. If no IPv4 address is available, the router should use the mechanism described in Requirement R-22 of Section 4.8 [RFC7600], which consists of using the dummy address 192.0.0.8 as the source address of originated ICMPv4 packets. Note however that using the same address on multiple routers may hamper debugging and fault isolation, e.g., when using the "traceroute" utility. This mirrors the behavior in Section 3 of [RFC9229].

[I-D.draft-ietf-intarea-extended-icmp-nodeid] provides a possible solution to this issue, by allowing the ICMP packet to carry a "host identifier" that can be used to identify the router that originated the ICMP by providing a unique IP address and/or a textual name for the node, in the case where each node may not have a unique IP address.

## 5. Implementation Status

( This section to be removed before publication. )

As this document does not really define a protocol, this implementation status section is much less formal. Instead, it is being used as a place to list implementations that are known to support this functionality, examples, notes, etc. This information is provided as a guide to the reader, and is not intended to be a complete list, nor endorsement, etc. If you know of an implementation which is not listed, please let the authors know.

### 5.1. Arista EOS

Arista has supported static IPv4 routes with IPv6 nexthops since EOS-4.30.1.

### 5.2. The Babel routing protocol

As noted above, this document is heavily based on RFC9229 (nee draft-ietf-babel-v4viav6), and this functionality is supported by babeld.

Pasted below is email sent to the babel mailing list (archived at https://mailarchive.ietf.org/arch/msg/babel/QtFi3F4TFfF7fXXlkHSpEnuT44Y/)

A route across three IPv6-only nodes:

```
$ ip route show 10.0.0.2
10.0.0.2 via inet6 fe80::216:3eff:fe00:1 dev lxcbr0 proto babel onlink
```

Here’s how it’s logged by babeld:

```
10.0.0.2/32 from 0.0.0.0/0 metric 384 (384) refmetric 288 id
02:16:3e:ff:fe:9a:5e:22 seqno 36425 chan (255) age 15 via lxcbr0 neigh
fe80::216:3eff:fe00:1 (installed)
```

Traceroute is a little confusing:

```
$ traceroute 10.0.0.2
traceroute to 10.0.0.2 (10.0.0.2), 30 hops max, 60 byte packets
 1  192.0.0.8 (192.0.0.8)  0.079 ms  0.019 ms  0.014 ms
 2  192.0.0.8 (192.0.0.8)  0.040 ms  0.023 ms  0.042 ms
 3  192.0.0.8 (192.0.0.8)  0.061 ms  0.030 ms  0.030 ms
 4  10.0.0.2 (10.0.0.2)  0.060 ms  0.040 ms  0.039 ms
```

PMTUD works fine (thanks to Toke):

```
19:58:47.402871 IP 192.168.0.27.60046 > 10.0.0.2.22: Flags [.],\
seq 33:1481, ack 33, win 502, options [nop,nop,TS val 917354570\
ecr 1849974691], length 1448
19:58:47.402874 IP 192.168.0.27.60046 > 10.0.0.2.22: Flags [P.],\
seq 1481:1537, ack 33, win 502, options [nop,nop,TS val 917354570\
ecr 1849974691], length 56
19:58:47.402906 IP 192.0.0.8 > 192.168.0.27: ICMP 10.0.0.2 \
unreachable- need to frag (mtu 1420), length 556
19:58:47.402919 IP 10.0.0.2.22 > 192.168.0.27.60046: Flags [.],\
ack 33, win 509, options [nop,nop,TS val 1849974692 \
ecr 917354569,nop,nop,sac 1 {1481:1537}], length 0
19:58:47.402934 IP 192.168.0.27.60046 > 10.0.0.2.22: Flags [.], \
seq 33:1401, ack 33, win 502, options [nop,nop,TS val 917354570 \
ecr 1849974692], length 1368

-- Juliusz
```

### 5.3. Linux

Linux has supported v4-via-v6 routes since kernel version 5.2, released on 2019-07-07.

#### 5.3.1. Example:

```
rincewind ~ #
ip -4 r a 192.0.2.23/32 via inet6 2001:db8::2342

rincewind ~ # ip r s 192.0.2.23/32
192.0.2.23 via inet6 2001:db8::2342 dev wlp36s0.25
```

### 5.4. Mikrotik RouterOS

Mikrotik RouterOS has supported v4-via-v6 routes since (at least) version 7.11beta2

{Editor note: I’m not sure when support was added. I tested this in Version 7.11beta2, and it worked there, but I believe that this functionality has existed for a while. I’ll try to find out when it was added.}

#### 5.4.1. Example

```
[wkumari@Dulles-CCR] /ip/route> print
Flags: D - DYNAMIC; I - INACTIVE, A - ACTIVE; c - CONNECT, s - STATIC,
d -DHCP, v - VPN; H - HW-OFFLOADED
Columns: DST-ADDRESS, GATEWAY, DISTANCE
#      DST-ADDRESS       GATEWAY                              DISTANCE
0  As  192.0.2.0/24      fe80::201:5cff:feb2:1646%1_Comcast         1
```

### 5.5. Cisco NX-OS

Cisco NX-OS has supported v4-via-v6 routes "for more than 8 years" -- Krishnaswamy Ananthamurthy

## 6. Security Considerations

The techniques described in this document make routing more flexible by allowing IPv4 routes to propagate across a section of a network that has only been assigned IPv6 addresses. This additional flexibility might invalidate otherwise reasonable assumptions made by network administrators, which could potentially cause security issues.

For example, if an island of IPv4-only hosts is separated from the IPv4 Internet by routers that have not been assigned IPv4 addresses, a network administrator might reasonably assume that the IPv4-only hosts are unreachable from the IPv4 Internet. This assumption is broken if the intermediary routers implement v4-via-v6 routing, which might make the IPv4-only hosts reachable from the IPv4 Internet. If this is not desirable, then the network administrator must filter out the undesirable traffic in the forwarding plane by implementing suitable packet filtering rules.

## 7. IANA Considerations

This document has no IANA actions.

## 8. References

### 8.1. Normative References

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

[RFC7600] Despres, R., Jiang, S., Ed., Penno, R., Lee, Y., Chen, G., and M. Chen, "IPv4 Residual Deployment via IPv6 - A Stateless Solution (4rd)", RFC 7600, DOI 10.17487/RFC7600, July 2015, <https://www.rfc-editor.org/rfc/rfc7600>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

### 8.2. Informative References

[I-D.draft-ietf-intarea-extended-icmp-nodeid]
Fenner, B. and R. Thomas, "Adding Extensions to ICMP Errors for Originating Node Identification", Work in Progress, Internet-Draft, draft-ietf-intarea-extended-icmp-nodeid-02, 26 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-intarea-extended-icmp-nodeid-02>.

[IANA-IPV4-REGISTRY]
"IANA IPv4 Address Registry", Web https://www.iana.org/assignments/iana-ipv4-special-registry/.

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

[RFC0826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <https://www.rfc-editor.org/rfc/rfc826>.

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

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/rfc/rfc4821>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/rfc/rfc4861>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <https://www.rfc-editor.org/rfc/rfc4915>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/rfc/rfc5120>.

[RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <https://www.rfc-editor.org/rfc/rfc7404>.

[RFC8950] Litkowski, S., Agrawal, S., Ananthamurthy, K., and K. Patel, "Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop", RFC 8950, DOI 10.17487/RFC8950, November 2020, <https://www.rfc-editor.org/rfc/rfc8950>.

[RFC8966] Chroboczek, J. and D. Schinazi, "The Babel Routing Protocol", RFC 8966, DOI 10.17487/RFC8966, January 2021, <https://www.rfc-editor.org/rfc/rfc8966>.

[RFC9229] Chroboczek, J., "IPv4 Routes with an IPv6 Next Hop in the Babel Routing Protocol", RFC 9229, DOI 10.17487/RFC9229, May 2022, <https://www.rfc-editor.org/rfc/rfc9229>.

## Acknowledgments

We are grateful to nnJoe Abley, Krishnaswamy Ananthamurthy, Bill Fenner, Tobias Fiebig, John Gilmore, Bob Hinden, David Lamparter, Gyan Mishra, tom petch, Herbie Robinson, Behcet Sarikaya, David Schinazi, Ole Troan, and テ詠ic Vyncke, for their helpful comments and suggestions on this document. We are also indebted to the members of the Babel community for the discussions that led to the creation of this document.

## Changes

This section is to be removed before publication, and the primary change log is the git repository. This is just a place to note some of the more substantive changes.

### Version 00-01

* Added note that this works just as well for IPv6 routes with an IPv4 next hop. (テ詠ic Vyncke)

* Cisco NX-OS has supported v4-via-v6 routes "for more than 8 years" (Krishnaswamy Ananthamurthy)

* Mention recursive next hops, and that the next hop may be a prefix. (Krishnaswamy Ananthamurthy)

* Hosts are routers too! (David Lamparter)

* Removed the claim that it’s mainly a UI issue.

## Authors’ Addresses

Juliusz Chroboczek
IRIF, University of Paris
Case 7014
75205 Paris Cedex 13
France
Email: jch@irif.fr

Warren Kumari
Google, LLC
Email: warren@kumari.net

Toke Hテクiland-Jテクrgensen
Red Hat
Email: toke@toke.dk