Workgroup: Internet Area
Internet-Draft:
draft-chroboczek-int-v4-via-v6-01
Published: 7 March 2022
Intended Status: Standards Track
Expires: 8 September 2022
Authors: J. Chroboczek
IRIF, University of Paris
T. Høiland-Jørgensen
Red Hat
W. Kumari
Google, LLC

IPv4 routes with an IPv6 next hop

Abstract

We propose "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. We describe the technique, and discuss 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-int-v4-via-v6/draft-chroboczek-int-v4-via-v6.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-chroboczek-int-v4-via-v6/.

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

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 8 September 2022.

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).

The case of routing IPv4 packets through an IPv6 next hop 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.

This document discusses the protocol design and operations implications of such routes and is designed to be used as a reference for future documents.

{ 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 either an IPv4 or an IPv6 next hop.

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 dermined 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 generalisation 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 will 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 (e.g., not stable enough).

3.3.1. Distance-vector routing protocols

3.3.2. Link-state routing protocols

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 an 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 could use the experimental 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.

{Editor note: It would be surprising to many operators to see something like:

```
> $ traceroute -n 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 64 hops max, 52 byte packets
 1  192.168.0.1  1.894 ms  1.953 ms  1.463 ms
 2  192.0.0.8  9.012 ms  8.852 ms  12.211 ms
 3  192.0.0.8  8.445 ms  9.426 ms  9.781 ms
 4  192.0.0.8  9.984 ms  10.282 ms  10.763 ms
 5  192.0.0.8  13.994 ms  13.031 ms  12.948 ms
 6  192.0.0.8  27.502 ms  26.895 ms
 7  8.8.8.8  26.509 ms
```

Is this a problem though? If this becomes common practice, will operators just come to understand that the repeated 192.0.0.8 is not actually a looping packet, but rather that the packet is (probably!) making forward progress? What if it goes: 192.168.0.1 -> 192.0.0.8 -> 10.10.10.10 -> 192.0.0.8 -> 172.16.14.2 -> dest? }

{ Editor note / question: 192.0.0.8 is assigned in the [IANA-IPV4-REGISTRY] as the "IPv4 dummy address". It may be used as a Source Address, and was requested in [rfc7600] to be used as the IPv4 source address when synthesizing an ICMPv4 packet to mirror an ICMPv6 error message. This is all fine and good - but, 192.0.0.0/24 is commonly considered a bogon or martian

Example (from a Juniper router):

```
wkumari@rtr2.pao> show route martians

inet.0:
             0.0.0.0/0 exact -- allowed
             0.0.0.0/8 orlonger -- disallowed
             127.0.0.0/8 orlonger -- disallowed
             192.0.0.0/24 orlonger -- disallowed
             240.0.0.0/4 orlonger -- disallowed
             224.0.0.0/4 exact -- disallowed
             224.0.0.0/24 exact -- disallowed
```

This means that these packets are likely to be filtered in many places, and so ICMP packets with this source address are likely to be dropped. Is this a major issue? Would requesting another address be a better solution? Would it help? If it were to be allocated from some more global pool, it would still likely require "magic" to allow it to pass BCP38 filters. }

5. 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.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. 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>.

7.2. Informative References

[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>.

[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>.

Acknowledgments

TODO acknowledge.

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 Høiland-Jørgensen
Red Hat

Email: toke@toke.dk