Announcing IPv4 routes with an IPv6 next-hop in the Babel routing protocol

aka. draft-bastian-babel-v4ov6

Théophile Bastian, joint work with Juliusz Chroboczek

ENS Paris, IRIF, Nexedi
Babel: **routing protocol**. Only concern: **build the routing table**.

<table>
<thead>
<tr>
<th>Network Prefix (IPv6)</th>
<th>Next-Hop (IPv6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001::/64</td>
<td>fd80::42</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Prefix (IPv4)</th>
<th>Next-Hop (IPv4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0/24</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>10.0.0.0/16</td>
<td>10.0.1.1</td>
</tr>
<tr>
<td>10.0.0.0/8</td>
<td>fe80::f0</td>
</tr>
</tbody>
</table>
The router’s job

Payload

Destination MAC address

LAYER 2

Destination IP address

LAYER 3

Payload

Destination MAC address

LAYER 2

Destination IP address

LAYER 3

Payload

Destination MAC address

LAYER 2

Destination IP address

LAYER 3
What's under the hood?

Payload

Destination MAC address

LAYER 2

Destination IP address

LAYER 3

MAC addr

IPv4 addr
What's under the hood?

<table>
<thead>
<tr>
<th>Payload</th>
<th>Destination IP address</th>
<th>Destination MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LAYER 3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dest. prefix</strong></td>
<td></td>
<td><strong>Next hop</strong></td>
</tr>
<tr>
<td><strong>IPv4 addr</strong></td>
<td></td>
<td><strong>4</strong></td>
</tr>
<tr>
<td><strong>MAC addr</strong></td>
<td></td>
<td><strong>4</strong></td>
</tr>
</tbody>
</table>

4 IPv4 addr

4 MAC addr
What's under the hood?

Payload

Destination MAC address

LAYER 2

Destination IP address

LAYER 3

Routing table

<table>
<thead>
<tr>
<th>Dest. prefix</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

IPv4 addr

MAC addr
### What’s under the hood?

<table>
<thead>
<tr>
<th>Payload</th>
<th>LAYER 3</th>
<th>Destination IP address</th>
<th>MAC addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAYER 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination MAC address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Payload**: The data being transmitted.
- **Destination MAC address**: The address of the device receiving the data.
- **Destination IP address**: The IP address of the device receiving the data.
- **MAC addr**: The address of the device sending the data.
- **IPv4 addr**: The IP address of the device sending the data.

The data travels from the sender through the network to the destination, passing through different layers with the appropriate addresses.
What's under the hood?

Hey, what's your MAC address?
Hey 4, what's your MAC address?

Hey 4, my MAC is △.
What’s under the hood?

Payload
LAYER 3
Destination IP address
LAYER 2
Destination MAC address

Payload
LAYER 3
Destination IP address
LAYER 2
Destination MAC address

Payload
LAYER 3
Destination IP address
LAYER 2
Destination MAC address

Payload
LAYER 3
Destination IP address
LAYER 2
Destination MAC address

Destination MAC address
IPv4 addr
MAC addr
### What’s under the hood?

<table>
<thead>
<tr>
<th>Layer 2</th>
<th>Layer 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination MAC address</td>
<td>Destination IP address</td>
</tr>
</tbody>
</table>

The next-hop is only used to find the MAC!

![Diagram showing network flow](image-url)
What’s under the hood?

The next-hop is only used to find the MAC!

WAIT, but then...

Why can't we have a v6 next hop?
What's under the hood?

<table>
<thead>
<tr>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAYER 3</td>
</tr>
<tr>
<td>Destination IP address</td>
</tr>
<tr>
<td>LAYER 2</td>
</tr>
<tr>
<td>Destination MAC address</td>
</tr>
</tbody>
</table>

Routing table

<table>
<thead>
<tr>
<th>Dest. prefix</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
</tbody>
</table>

4 IPv4 addr

3 MAC addr

6 IPv6 addr
What’s under the hood?

Hey 6, what’s your MAC address?

Hey 6, my MAC is ▲.
What's under the hood?

payload

destination MAC address

Layer 2

destination IP address

Layer 3

MAC addr

IPv4 addr 4

IPv6 addr 6

Payload

Layer 2

destination MAC address

Payload

Layer 3

destination IP address

payload

Layer 2

destination MAC address

payload

Layer 3

destination IP address

payload

Layer 2

destination MAC address

payload

Layer 3

destination IP address

payload

Layer 2

destination MAC address

payload

Layer 3

destination IP address

payload

Layer 2

destination MAC address
The idea behind

New type of route! v4 prefix with v6 next-hop: v4-over-v6 route.

- Useful to go through a router with no v4 address configured
- Possible use: v6-only core, still serve v4 to clients without tunnel

Not an original idea! BGP had it first: draft-ietf-bess-rfc5549revision
Supported in Linux since 5.2 – July 2019!

```bash
# ip route add 10.42.0.0/16 via 10.40.0.42 dev eno1
```

- **Usually, add a v4 route with**

- **v4-over-v6 (recent kernel):**

  ```bash
  # ip route add 10.42.0.0/16 via inet6 fe80::a0de:baf:b39b dev eno1
  ```
Babel protocol extension
Advertising v4-over-v6 routes

**Interface with v4 address:** no changes; just as unextended babel.

**Interface with only v6 addresses:**

- **Receiving** a v4 route: install it anyway, the router’s address is irrelevant.
- **Announcing** a v4 route: use v4-over-v6; we need a next-hop and only have v6 at hand.
Backwards compatibility

Backwards compatibility: an unextended node must

- Be able to ignore v4-over-v6 routes
- Route correctly pure v4 and v6

In unextended Babel, Address Encodings (AEs) define the type of address/prefix contained in a TLV.

- IPv4 address
- IPv6 address
- link-local IPv6 address
Encoding v4-over-v6 routes: choices

Various encodings possible, among which:

1. [Toke] An IPv4 route announced without previously setting a valid next-hop is considered v4-over-v6.
   - Is backwards compatible – but not obviously so
   - Not clear an extension is being used

2. [Bastian + JCH] New specific AE for v4-over-v6 routes, and next-hop is...
   i. . . specific to v4-over-v6 routes
   ii. . . same as for v6 routes
      → more compact, just as clear
   - Backwards compatible and clean
   - A tiny bit more verbose
Encoding v4-over-v6 routes: choices

Various encodings possible, among which:

1. *[Toke]* An IPv4 route announced without previously setting a valid next-hop is considered v4-over-v6.
   - Is backwards compatible – but not obviously so
   - Not clear an extension is being used

2. *[Bastian + JCH]* New specific AE for v4-over-v6 routes, and next-hop is...
   i. . . .specific to v4-over-v6 routes
   ii. . . .same as for v6 routes
       → more compact, just as clear
   - Backwards compatible and clean
   - A tiny bit more verbose

⇝ 2.ii. Add a new AE. No need for new TLVs.
Conclusion

- New type of route: v4-over-v6, v4 destination, v6 next-hop
- Route IPv4 over an IPv6 network core. *Look, Ma! No tunnels!*
- Protocol described and drafted
- Production-ready implementation available on the babeld repository

- **Intended status:** experimental
- Opinions: should it be adopted by workgroup or carried alone?

**RFC draft**

huit.re/draft-v4ov6

**These slides**

huit.re/ietf108-v4ov6