Published documents:

- RFC 8965, applicability;
- RFC 8966, base specification;
- RFC 8967, MAC authentication;
- RFC 8968, Babel over DTLS.

_The current procedural rules [...] can lead to situations where WGs or document authors believe that one or two ADs are deliberately blocking the progress of a WG document [...] Appeal processes in these circumstances are limited — RFC 3774, May 2004_

Unpublished documents:

- information model and YANG;
- source-specific;
- v4-via-v6;
- diversity (not planning to publish).
Source-specific


draft-ietf-babel-source-specific:
- technically correct;
- technically complete;
- difficult to understand without reading the paper.

The document is badly written:
- no point addressing individual IESG comments;
- rewrite, make better, resubmit.
draft-ietf-babel-v4-via-v6: IPv4 via IPv6 next hops.

No encapsulation, no translation: IPv6 is only used for neighbour discovery.
v4-via-v6

- Configure an IPv6 Babel network;
- IPv4 routing works out of the box:
  - no configuration of intermediate routers;
  - no encapsulation;
  - no translation.

Encourages the IPv4→IPv6 transition:
- makes it worthwhile to build an IPv6 network.
v4-via-v6

draft-ietf-babel-v4viav6-00:
- implemented (Théophile Bastian) in spring 2021;
- some testing (not enough);
- presented (as “v4-in-v6”) to Babel WG.

Margaret Cullen (thanks!) noticed two problems:
- the name was misleading, v4-via-v6 is better;
- what about ICMP.
v4-via-v6 and ICMP

Modern IPv4 relies on ICMP from intermediate routers:
– PMTU discovery relies on fragmentation required.

Three distinct paths:
– data path;
– ack path;
– ICMP path (for every router).

There is no fate sharing between the data/ack paths and the ICMP path. End-to-end is broken!
v4-via-v6 and ICMP (2)

Three distinct paths:
- data path;
- ack path;
- ICMP path (for every router).

With v4-via-v6, the ICMP path is not necessarily operational.

Since there is no fate sharing, mysterious failures.
v4-via-v6 and ICMP: solutions

ICMP needs a source address on each router.

Possible solutions:
1. require that every router have an IPv4 address; or
2. use a single IPv4 address for all routers; or
3. define ICMPv4-via-v6 for unnumbered hosts;
4. give up on ICMP.
Solution 1: IPv4 address on each router

To send ICMPv4, a router requires an IPv4 address. Need not be assigned to the outgoing interface. (Weak host model.)

Require an IPv4 address on each router:
- loopback address (Cisco-style);
- borrowed from another interface (Linux-style).

Consequences:
- requires manual per-router configuration;
- only one address per router, not per interface.

Mitigation:
- Donald suggests that the IPv4 address could be autogenerated (drawn randomly) à la Zeroconf.
Solution 2: a single IPv4 address for all routers

Require an IPv4 address on each router. Use the same address on all routers.

Consequences:
- no per-router configuration;
- debugging made more difficult.

Issues:
- should this “fake” address be reserved, or locally assigned?
- what experience is there with sharing addresses (not NAT, not anycast — only used as source).
Solution 3: define ICMPv4-via-v6

Define a way to send **ICMPv4 without an IPv4 address**.

Compounds the **third path** problem: lack of fate sharing made worse (IPv4 vs. IPv6).

**Out of scope** for this working group.
Solution 4: give up on ICMPv4

Original sin: **ICMPv4** as an integral part of the protocol. (As opposed to debugging and fault isolation.)

Solution: **avoid relying on ICMPv4.**

Example: RFC 4821,

*Packetization Layer Path MTU Discovery.*

(More explicit datapath signalling might be better.)

**Out of scope** for this working group.
Proposal and conclusion

Proposed wording:
- every IPv4 router **MUST** have at least one IPv4 address;
  - giving up on ICMPv4 is not currently an option;
- sharing a single address between routers is a tempting option, but **more experimentation is needed**;
  - that’s why we’re aiming for experimental.

Only mention in passing:
- ICMPv4-via-v6;
- packetisation-layer pMTUd.

Agreed?