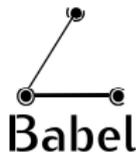


Proposed changes to the Babel routing protocol

Juliusz Chroboczek
IRIF
Université Paris-Diderot (Paris 7)

21 July 2016



Working group tradition

This is a **new working group**. It will exist for some time. (3 months? 3 years?)

We are establishing a **tradition** which will set the tone of things to come.

Consistent with both Babel and IETF traditions:

- **work happens on the mailing list**,
not at face-to-face meetings;
- **working code**:
if it's not implemented, it didn't happen;
- **think of the users**:
if it's being deployed, it did happen;
- Babel is **not a clone** of an existing routing protocol.

Work happens on the mailing list

Face-to-face meetings are fun and useful to get to know people.

But **real work happens on the mailing list**:

- chance to **read up** before replying;
- people who **cannot travel**;
- **non-native speakers** of English;
- **bad public speakers**;
- yields a **better record** of our work.

Working code

If it's **not implemented**, it **doesn't get into the spec**.

Equivalently, **if you want it in the spec**, you must **get it implemented**:

- **do it yourself** (best thing);
- **bribe** Toke (bird);
- **beg** Markus (pybabel);
- **bully** Juliusz and his crowd (babeld);
- **pay** a third party (discouraged).

Deployment experience

Babel has a **user community**.

Design and implementation are (to a great extent) guided by the users:

- if users rely on it, it's either **a good idea** or **works around a missing feature**;
- if users ignore it, it's **probably not needed**.

Babel is a new protocol

Babel is a **new protocol**:

- Babel is **not OSPF**;
- Babel is **not IS-IS**;
- Babel is **not EIGRP**.

Just because a feature has been found **useful in another routing protocol** doesn't mean it's **necessary in Babel**.

Does your feature solve a problem in Babel?

If it doesn't, it **doesn't go in**.

Compatibility decision

The first part of our charter (my wording):

- write a standards track protocol definition based on
 - RFC 6126 (The Babel Routing Protocol); and
 - RFC 7557.

RFC 6126 has a **major version** field:

- **set to 2**;
- packets with a different value are **silently ignored**.

We need to make a choice:

- **small**: fix bugs, tighten the spec, remain **strictly compatible** with RFCs 6126 and 7557;
- **medium**: retain version 2, remain **interoperable** with deployed implementations, but **not strictly compatible**;
- **large**: major version 3, **no interoperability**.

Compatibility decision (2)

Juliusz is in favour of option **medium**:

- **retain major version 2**;
- ensure that it is possible to **remain interoperable** with deployed implementations;
- **do make changes** that are technically incompatible:
 - e.g. forbid a given value;
 - e.g. add a new TLV;
 - e.g. obsolete a given TLV.

This decision can be **changed** if we're stuck:

- **becoming incompatible** is easy;
- **recovering compatibility** is difficult.

Compatibility decision (3)

By not bumping the major version number, **we give up** on some possibilities:

- cannot add a **mandatory bit** to sub-TLVs
 - **can be worked around** by using a new TLV or AE;
- **expanding the primary metric** beyond 16 bits
 - can be worked around by using a **secondary metric**;
- **cleaning up the packet format**
 - avoids endless bikeshedding?

Non-issues:

- **TLV length** is limited to 255 octets
 - this is a UDP-based unreliable protocol, it is not designed to carry massive amounts of data;
- **TLV type is unlimited**
 - use type 255 for signalling 16-bit types.

Proposed incompatible changes

Remaining interoperable doesn't prevent us from making **technically incompatible** changes:

Certain:

- forbid **router-ids 0 and all-ones**;
- **tighten compression requirements**.

More tentative (needs implementation experience):

- define **interval=0 in Hello?**
- define **unicast Hello?**
- remove **AE=0?**
- remove **non-seqno requests?**

Possible improvements

In addition to changes, some compatible **improvements** have been suggested:

- mechanism for detecting **router-id collisions**;
- specify **error handling**.

Extensions

What about **extensions**?

- **source-specific routing (SADR):**
 - **in-charter**;
 - planned **incompatible redesign**, already discussed on mailing-list, implementation planned for babeld-1.9;
- **RTT-based routing:**
 - **out-of-charter**;
 - mature, stable, deployed extension, **no changes planned**.
- **radio-interference aware routing:**
 - **out-of-charter**;
 - **not ready for standardisation yet**.

Security

What **security mechanism** should be standardised?

- **RFC 7298** (HMAC-based authentication)?
- **Stenberg-style** security (use unicast and TLS)?
- **statically keyed IPsec** with replay protection?
- **something else?**

We need an **independent opinion** on RFC 7298.

Conclusion

We're **ready to start work**:

- **WG** adoption of RFC 6126 bis;
- **integrate** RFC 7557;
- fix **obvious bugs**;
- **implement** the proposed changes, **discuss** on the mailing list, **integrate** in the document (in that order);
- decide **what to do about security**.