

# Source-specific routing with a mandatory sub-TLV

*(draft-boutier-babel-source-specific-03)*

Matthieu Boutier, Juliusz Chroboczek

IRIF (ex Laboratoire PPS) - Université Paris Diderot  
boutier@irif.fr, jch@irif.fr

July 2017

IETF 97 — Prague  
Babel

# Background: source-specific routing

The routing decision depends both on the [destination and source of the packet](#).

Routing [tables map pairs of prefixes](#) (destination, source) to next-hop.

The main use case is for [host-centric multihoming](#) (with PA addresses).

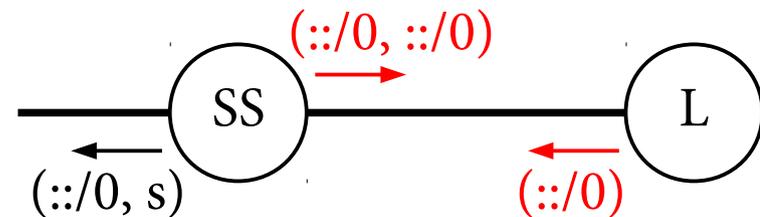
# Source-specific routing in Babel

A natural **solution** for Babel is to **add a source prefix**:

- to data **structures** (source table, route table, etc.)
- to **messages**: Update, Route Request, Seqno Request.

→ **the whole message MUST be ignored by legacy routers**

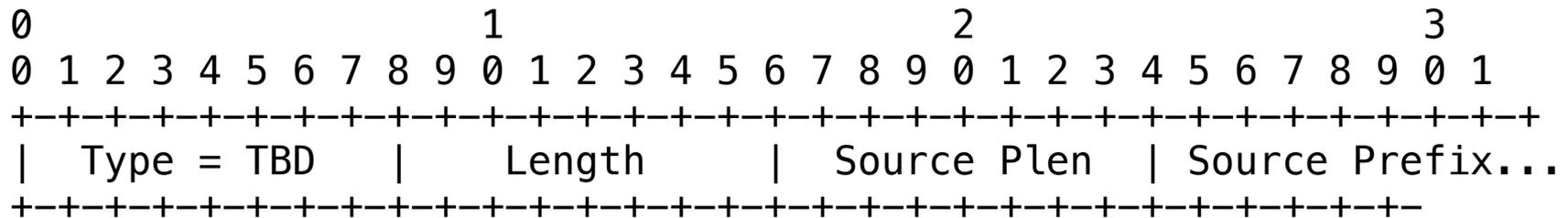
Example: persistent routing loop  
with partially understood update.



# From three TLV to one sub-TLV

Instead of defining three new TLVs, we define only one mandatory sub-TLV

## The Source Prefix sub-TLV



Source-Specific Update  
 Source-Specific Route Request  
 Source-Specific Seqno Request

draft-boutier-babel-source-specific-01

→

Update  
 Route Request      +      Source Prefix  
 Seqno Request                      sub-TLV

draft-boutier-babel-source-specific-03

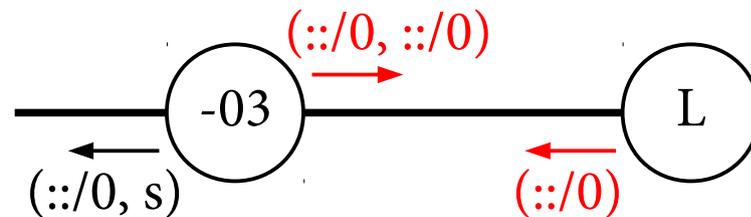
# Incompatibility with 6126

RFC 6126 doesn't handle mandatory sub-TLVs

A 6126 router will:

- ignore the sub-TLV,
- install the route (as a legacy route),
- announce the installed route.

persistent routing loop due to partially understood update.



-03: *implement this draft*  
L: *6126 router (with or without -01 source-specific routing)*

# Implementation Status

It's **implemented**.

It **works**.

It uses an **experimental sub-TLV type**.

# Wildcard requests

6126 says: « AE == 0 requests a full routing table dump »

**Problems:** a legacy router asks only for legacy routes.

- Does sending all routes break the **semantics**?
- Sending more routes is **waste**.
- If each extension define its requests, **how to combine extensions**?

# Wildcard requests (2)

6126 says: « AE == 0 requests a full routing table dump »

Proposals overview (*detailed in the draft*):

- request a **full dump**, reply with a **full dump**,
- request for **each extension and combination** of extensions, reply with the **requested routes**,
- request for **each extension**, reply with the **requested routes and combinations**,
- deprecate wildcard route requests.

# Remaining proposals

1. Put one Wildcard Route Request (WRR).
- ~~2. Put one WRR with all sub-TLVs you know but without mandatory bit.~~
3. Put one WRR per extension and per combinations.
- ~~4. Deprecate WRR.~~
5. Define a new sub-TLV with one field per extension. Send understood combinations.
6. Put one WRR per extension. Send understood combinations.

wasted routes  
hard to define  
wasted TLVs  
parser state

Red	Green	Green	Green
Green	Green	Green	Red
Green	Green	Red	Green
Green	Green	Green	Green
Orange	Green	Green	Green
Orange	Red	Green	Green

# Wildcard updates

A wildcard update is, in fact, a **wildcard retraction**.

As Juliusz wrote:

*Think of a wildcard retraction as saying "I'm shutting down really soon now, please route around me."*

of course, you will also retract source-specific routes

→ **no source-specific wildcard retraction**

# Conclusion

- Thanks to mandatory sub-TLVs.
- It's **implemented**, it **works**...
- Choosing a **sub-TLV number** for the Source Prefix sub-TLV:  
→ **128?**
- Choosing a proposal for source-specific **requests**.

Working group **adoption** ?