Network Working Group
Internet-Draft
Updates: 6126bis
(if approved)
Intended status: Standards Track
Expires: December 17, 2017

M. Boutier
J. Chroboczek
IRIF, University of Paris-Diderot

June 15, 2017

Source-Specific Routing in Babel
draft-boutier-babel-source-specific-02

Abstract

This document describes an extension to the Babel routing protocol to support source-specific routing.

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 http://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 December 17, 2017.

Copyright Notice

Table of Contents

1. TODOs

- Source Prefix sub-TLV type: TBD
- check references (Section) for BABEL in 6126bis

2. Introduction and background

Source-specific routing (other known as Source Address Dependant Routing, SAD Routing or SADR) is an extension to traditional next-hop routing where packets are routed according to both their destination and their source address. This document describes the source-specific routing extension to the Babel routing protocol as defined in 6126bis [BABEL]. It notably requires the sub-TLV mandatory bit.

Background information about source-specific routing is provided in [SS-ROUTING].

3. Data Structures

This extension adds some data to the data structures maintained by a Babel node.

3.1. The Source Table

Every Babel node maintains a source table, as described in [BABEL], Section 3.2.5. A source-specific Babel node extends this table with the following field:

- the source prefix (sprefix, splen) specifying the source address of packets to which this entry applies.

If a source table entry has a zero length source prefix (splen equals to 0), then the entry is a non-source-specific entry, and is treated just like a source table entry defined by the original Babel protocol.

With this extension the route entry contains a source which itself contains a source prefix. Notwithstanding the accidental similarity in their names, these are two very different concepts, and should not be confused.

3.2. The Route Table

Every Babel node maintains a route table, as described in [BABEL], Section 3.2.6. With this extension, this table is indexed by the 5-tuple (prefix, plen, source prefix, source plen, router-id)

obtained from the associated source table entry.

If a route table entry has a zero length source prefix, then the entry is a non-source-specific entry, and is treated just like a route table entry defined by the original Babel protocol.

3.3. The Table of Pending Requests

Every Babel node maintains a table of pending requests, as described in [BABEL], Section 3.2.7. A source-specific Babel node extends this table with the following entry:

- the source prefix being requested.

4. Data Forwarding

In next-hop routing, if two routing table entries overlap, then one is necessarily more specific than the other; the "longest prefix rule" specifies that the most specific applicable routing table entry is chosen.

With source-specific routing, there might no longer be a most specific applicable prefix: two routing table entries might match a given packet without one necessarily being more specific than the other. Consider for example the following fragment of a routing table:

```
(2001:DB8:0:1::/64, ::/0, A)
(::/0, 2001:DB8:0:2::/64, B)
```

This specifies that all packets with destination in 2001:DB8:0:1::/64 are to be routed through A, while packets with a source in 2001:DB8:0:2::/64 are to be routed through B. A packet with source 2001:DB8:0:2::42 and destination 2001:DB8:0:1::57 matches both rules, although neither is more specific than the other. A choice is necessary, and unless the choice being made is the same on all routers in a routing domain, persistent routing loops may occur.

A Babel implementation MUST choose routing table entries by using the so-called destination-first ordering, where a routing table entry R1 is preferred to a routing table entry R2 when either R1's destination prefix is more specific than R2's, or the destination prefixes are equal and R1's source prefix is more specific than R2's. (In more formal terms, routing table entries are compared using the lexicographic product of the destination prefix ordering by the source prefix ordering.)

In practice, this means that a source-specific Babel implementation must take care that any lower layer that performs packet forwarding obey this semantics. In particular:

- If the lower layers implement the destination-first ordering, then the Babel implementation MAY use them directly;
- If the lower layers can hold source-specific routes, but not with the right semantics, then the Babel implementation MUST disambiguate the routing table by using a suitable disambiguation algorithm (see [SS-ROUTING] for such an algorithm);
- If the lower layers cannot hold source-specific routes, then a Babel implementation MUST silently ignore any source-specific routes.

5. Protocol Operation

This extension does not fundamentally change the operation of the Babel protocol. We only described the fundamental differences between the original protocol and the extension in this section. The other mechanisms described in [BABEL] (Section 3) may be infered by using pairs of (destination, source) prefixes instead of just (destination) prefixes.

5.1. Source-specific messages

A route of this extension with a zero-length source prefix is the same than a route without source prefix (a route of the classical Babel). In both of the cases, packets are accepted independantly of their source address. Thus, a route is said source-specific only if its source prefix has a non-zero length.

Three messages are used to communicate informations on routes: Updates, Route Requests and Seqno Requests. With this extension, these messages carry an additionnal source prefix if (and only if) the corresponding route is source-specific. More formally, an Update, a Route Request and a Seqno Request MUST carry a source prefix if they concern a source-specific route (non-zero length source prefix) and MUST NOT carry a source prefix otherwise (zero length source prefix). A message which carry a source prefix is said source-specific.

5.2. Route Acquisition

When a non-source-specific Babel node receives a source-specific update, it just ignores it.

On receipt of a source-specific update (id, prefix, source prefix,

seqno, metric), a source-specific Babel node behaves as described in [BABEL] Section 3.5.4 though indexing entries by (neigh, id, prefix, source prefix). When a source-specific Babel node receives a non-source-specific update, it MUST consider this update as carrying a zero length source prefix.

5.3. Wildcard requests

TODO: behaviour to be defined.

5.3.1. Proposal 1

The original Babel protocol states that when a node receives a wildcard route request, it SHOULD send a full routing table dump. This extension does not change this statement: a source-specific node SHOULD send a full routing table dump when receiving a wildcard request.

Source-specific wildcard requests does not exist: a wildcard request SHOULD NOT carry a source prefix.

5.3.2. Proposal 2

We assume that a mandatory sub-TLV has a corresponding non-mandatory sub-TLV. This proposal is like Proposal 3 but instead of having multiple wildcard request TLVs, one for each kind of routes understood, we use one wildcard request with sub-TLVs corresponding to the extension. To have a full routing table dump, a node sends a wildcard requests with a non-mandatory Source sub-TLV.

A source-specific node SHOULD always attach a non-mandatory Source sub-TLV to its wildcard requests.

This proposal has been rejected because it implies to share the space of non-mandatory and mandatory sub-TLVs.

5.3.3. Proposal 3

The Babel protocol provides the ability to request a full routing table dump by sending a "wildcard request", a route request with the AE field set to 0. As the original protocol has no source-specific routes, such a request may only concern non-source-specific routes. This extension does not modify the semantics of wildcard requests in that sense: a wildcard request prompts the receiver to send its non-source-specific routes only, and a Babel node SHOULD NOT send any source-specific updates in reply to a wildcard request.

To obtain a dump of the source-specific routes, a source-specific

wildcard request MUST be used. A source-specific wildcard request is a wildcard request carrying a zero length source prefix.

When a node receives a source-specific wildcard request, it SHOULD send a dump of its routes which are source-specific "only". It SHOULD NOT send any non-source-specific routes in reply to a source-specific wildcard request. It SHOULD NOT send any source-specific routes which are under the effect of a future extension. Such extension should detail how to handle the possible combinations.

In consequence, a node requiring a full routing table dump must send both a non-source-specific wildcard request and a source-specific wildcard request.

5.3.4. Proposal 4

Wildcard requests are deprecated. Either deprecate it in 6126bis, or say the following.

A node receiving a wildcard request SHOULD ignore it.

This proposal has been rejected because wildcard requests speeds up the convergence of the network on boot. This is considered important.

5.3.5. Note on Overhead between (1) and (3)

Sending one wildcard request (1) instead of a few something-specific wildcard requests (3) in a negligible gain.

Non-source-specific nodes sending requests to source-specific nodes may reduce the global overhead with (3). But, if the network has no source-specific route, there is no overhead to reduce; if there is only a few source-specific routes (like in a home network), the overhead would be negligible. Thus, the interesting case is when there is a lot of source-specific routes.

We can imagine a network with a source-specific backbone announcing a default route and catching all trafic. Good old routers not supporting this extensions would be put at some backbone leafs. Is sbabeld part of that use case ?

Couldn’t we just send a Route Request for *default* ?

6. Backwards compatibility

The protocol extension defined in this document is, to a great

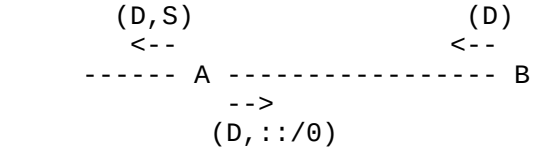
extent, interoperable with the base protocol defined in [BABEL] (and all its known extensions). More precisely, if non-source-specific routers and source-specific routers are mixed in a single routing domain, Babel's loop-avoidance properties are preserved, and, in particular, no persistent routing loops will occur.

TODO: Should we put a warning to say it's not the case with the Experimental Track Babel ?

6.1. Loop-avoidance

The extension defined in this protocol uses a new Mandatory sub-TLV to carry the source prefix information. As discussed in Section 4.4 of [BABEL], this encoding ensures that non-source-specific routers will silently ignore the whole TLV, which is necessary to avoid persistent routing loops in hybrid networks.

Consider two nodes A and B, with A source-specific announcing a route to (D, S). Suppose that B ignores the source prefix information when it receives the update, and reannounces it as D. This is reannounced to A, which treats it as (D, ::/0). Packets destined to D but not sourced in S will be forwarded by A to B, and by B to A, causing a persistent routing loop:



6.2. Starvation and Blackholes

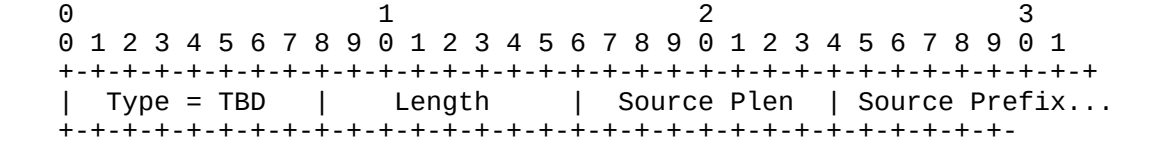
In general, discarding of source-specific routes by non-source-specific routers will cause routing starvation. Intuitively, unless there are enough non-source-specific routes in the network, non-source-specific routers will suffer starvation, and discard packets for destinations that are only announced by source-specific routers.

A simple yet sufficient condition for avoiding starvation is to build a connected source-specific backbone that includes all of the edge routers, and announce a (non-source-specific) default route towards the backbone. However, introducing such a default route in the network may in the same time introduce a blackhole. This tradeoff is let to the administrator.

7. Protocol Encoding

This extension defines a new sub-TLV used to carry a source prefix by the three following existing messages: Update, Route Request and Seqno Request.

7.1. Source Prefix sub-TLV



Fields:

Type Set to TBD to indicate a Source Prefix sub-TLV.

Length The length of the body, exclusive of the Type and Length fields.

Source Plen The length of the advertised source prefix. This MUST NOT be 0.

Source Prefix The source prefix being advertised. This field's size is (Source Plen)/8 rounded upwards.

The Source Prefix field's encoding is the same than the Prefix's. It is defined by the AE field of the corresponding TLV.

Remark that this sub-TLV is a Mandatory sub-TLV. The whole TLV MUST be ignored if that TLV is not recognized. Otherwise, routing loops may occur.

7.2. Source-specific Update

The source-specific Update is an Update TLV with a Source Prefix sub-TLV. It advertises or retracts source-specific routes in the same manner than routes with non-source-specific Updates (see [BABEL]) except for wildcard updates.

Wildcard updates MUST NOT carry any source prefix. Wildcard updates (in fact, wildcard retraction) are used when a Babel node stops: a receiver retracts all routes announced by the announcing node. There is no use case for source-specific wildcard updates. A source-specific Babel node receiving a (legacy) wildcard update MUST retracts all routes it learns from this node (including source-specific ones).

Contrary to the destination prefix, this extension does not compress the source prefix attached to Updates. The destination prefix uses compression as defined in [BABEL] for Updates with Mandatory

extensions.

However, as defined in [BABEL] (Section 4.5), the compression is allowed for the destination prefix of source-specific routes. Legacy implementation will correctly update their parser state, while ignoring the whole TLV afterwards.

7.3. Source-specific (Route) Request

TODO: A source-specific Route Request prompts the receiver to send an update for a given pair of destination and source prefixes. It MUST NOT be used to request a full routing table dump. The Source Prefix sub-TLV of a wildcard source-specific Route Request (Request with AE equals to 0 and a Source Prefix sub-TLV) MIGHT be ignored: a receiver MIGHT reply by a full routing table dump.

7.4. Source-Specific Seqno Request

A source-specific Seqno Request is just like a Seqno Request for a source-specific route. It uses the same mechanisms described in [BABEL].

8. IANA Considerations

IANA is instructed to add the following entry to the "Babel sub-TLV Types" registry:

Type	Name	Reference
TBD	Source Prefix	(this document)

9. Security considerations

The extension defined in this document adds a new sub-TLV to three TLVs already present in the original Babel protocol. It does not by itself change the security properties of the protocol.

10. References

10.1. Normative References

[BABEL] Chroboczek, J., "The Babel Routing Protocol", Internet Draft draft-ietf-babel-rfc6126bis-02, May 2017.

10.2. Informative References

[SS-ROUTING]
Boutier, M. and J. Chroboczek, "Source-Specific Routing", August 2014.

In Proc. IFIP Networking 2015. A slightly earlier version is available online from http://arxiv.org/pdf/1403.0445.

Authors' Addresses

Matthieu Boutier
IRIF, University of Paris-Diderot
Case 7014
75205 Paris Cedex 13,
France

Email: boutier@irif.fr

Juliusz Chroboczek
IRIF, University of Paris-Diderot
Case 7014
75205 Paris Cedex 13,
France

Email: jch@irif.fr