Network Working Group   M. Boutier
Internet-Draft   J. Chroboczek
Intended status: Standards Track   IRIF, University of Paris-Diderot
Expires: August 4, 2018   January 31, 2018

Source-Specific Routing in Babel
draft-ietf-babel-source-specific-03

Abstract

Source-specific routing (also known as Source-Address Dependent Routing, SADR) is an extension to traditional next-hop routing where packets are forwarded according to both their destination and their source address. This document describes an extension for source-specific routing to the Babel routing protocol.

Table of Contents

1. Introduction and background

The Babel routing protocol [BABEL] is a distance vector routing protocol for next-hop routing. In next-hop routing, each node maintains a forwarding table which maps destination prefixes to next hops. The forwarding decision is a per-packet operation which depends on the destination address of the packets and on the entries of the forwarding table. When a packet is about to be routed, its destination address is compared to the prefixes of the routing table: the entry with the most specific prefix containing the destination address of the packet is chosen, and the packet is forwarded to the associated next-hop. Next-hop routing is a simple, well understood paradigm that works satisfactorily in a large number of cases.

Source-specific routing [SS-ROUTING], or Source Address Dependent

Routing (SADR) [DSR], is a modest extension to next-hop routing where the forwarding decision depends not only on the destination address but also on the source address of the packet being routed, which makes it possible for two packets with the same destination but different source addresses to be routed following different paths.

The forwarding tables are extended to map pairs of prefixes (destination, source) to next hops. When multiple entries match a given packet, the one with the most specific destination prefix is chosen, and, in case of equality, the one with the most specific source prefix.

The main application of source-specific routing is a form of multihoming known as multihoming with multiple addresses. When using this technique in a network connected to multiple providers, every host is assigned multiple addresses, one per provider. When a host sources a packet, it picks one of its addresses as the source address, and source-specific routing is used to route the packet to an edge router that is connected to the corresponding provider, which is compatible with [BCP84]. Unlike classical multihoming, this technique is applicable to small networks, as it does not require the use of provider-independent addresses, or cause excessive growth of the global routing table. More details are given in [SS-ROUTING] and [DSR].

This document describes a source-specific routing extension for the Babel routing protocol [BABEL]. This involves minor changes to the data structures, which must include a source prefix in addition to the destination prefix already present, and some changes to the Update, Route Request and Seqno Request TLVs, which are extended with a source prefix. The source prefix is encoded using a mandatory sub-TLV ([BABEL] Section 4.4).

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Data Structures

A number of the conceptual data structures described in Section 3.2

of [BABEL] contain a destination prefix. This specification extends these data structures with a source prefix. Data from the original protocol, which do not specify a source prefix, are stored with a zero length source prefix, which matches exactly the same set of packets as the original, non-source-specific data.

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 specifying the source address of packets to which this entry applies.

The source table is now indexed by triples of the form (prefix, source prefix, router-id).

Note that the route entry contains a source which itself contains a source prefix. These are two very different concepts that should not be confused.

3.2. The Route Table

Every Babel node maintains a route table, as described in [BABEL] Section 3.2.6. Each route table entry contains, among other data, a source, which this specification extends with a source prefix as described above. The route table is now indexed by triples of the form (prefix, source prefix, neighbour), where the prefix and source prefix are obtained from the source.

3.3. The Table of Pending Seqno Requests

Every Babel node maintains a table of pending seqno 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.

The table of pending seqno requests is now indexed by triples of the form (prefix, source prefix, router-id).

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 entry: two routing table entries might match a given packet without one necessarily being more specific than the other. Consider for example the following routing table:

destination	source	next-hop
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 all packets with 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. More details are given in [SS-ROUTING] Section IV.C.

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.) This is consistent with the behaviour described in Section 3.3 of [DSR].

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] Section V.B for such an algorithm);

- If the lower layers cannot hold source-specific routes, then a Babel implementation MUST silently ignore (drop) any source-specific routes.

5. Protocol Operation

This extension does not fundamentally change the operation of the Babel protocol, and we therefore only describe differences between the original protocol and the extended protocol.

In the original protocol, three TLVs carry a destination prefix: Updates, Route Requests and Seqno Requests. This specification extends these messages to optionally carry a source prefix sub-TLV, as described in Section 7 below. The sub-TLV is marked as mandatory, so that an unextended implementation will silently ignore the whole enclising TLV. A node obeying this specification MUST NOT send a TLV with a zero-length source prefix: instead, it sends a TLV with no source prefix sub-TLV. Conversely, an extended implementation MUST interpret an unextended TLV as carrying a source prefix of zero

length. Taken together, these properties ensure interoperability between the original and extended protocols (see Section 6 below).

5.1. Protocol Messages

This extension allows three TLVs of the original Babel protocol to carry a source prefix: Update TLVs, Route Request TLVs and Seqno Request TLVs.

In order to advertise a route with a non-zero-length source prefix, a node sends a source-specific Update, i.e., an Update with a source prefix sub-TLV. When a node receives a source-specific Update (prefix, source prefix, router-id, seqno, metric) from a neighbour neigh, it behaves as described in [BABEL] Section 3.5.4, except that the entry under consideration is indexed by (prefix, source prefix,

neigh) rather than just (prefix, neigh).

Similarly, when a node needs to send a Request of either kind that applies to a route with a non-zero length source prefix, it sends a source-specific Request, i.e., a Request with a source prefix sub-TLV. When a node receives a source-specific Request, it behaves as described in Section 3.8 of [BABEL], except that the request applies to the Route Table entry carrying the source prefix indicated by the sub-TLV.

5.2. Wildcard Messages

In the original protocol, the Address Encoding value 0 is used for wildcard messages: messages that apply to all routes, of any address family and with any destination prefix. Wildcard messages are allowed in two places in the protocol: wildcard retractions are used to retract all of the routes previously advertised by a node on a given interface, and wildcard Route Requests are used to request a full dump of the Route Table from a given node. Wildcard messages are intended to apply to all routes, including routes decorated with additional data and AE values to be defined by future extensions, and hence this specification extends wildcard operations to apply to all routes, whatever the value of the source prefix.

More precisely, a node receiving an Update with the AE field set to 0 and the Metric field set to infinity (a wildcard retraction) MUST apply the route acquisition procedure described in Section 3.5.4 of [BABEL] to all of the routes that is has learned from the sending node, whatever the value of the source prefix. A node MUST NOT send a wildcard retraction with an attached source prefix, and a node that receives a wildcard retraction with a source prefix MUST silently ignore it.

Similarly, a node that receives a route request with the AE field set to 0 (a wildcard route request) SHOULD send a full routing table dump, including routes with a non-zero-length source prefix. A node MUST NOT send a wildcard request that carries a source prefix, and a node receiving a wildcard request with a with a source prefix MUST silently ignore it.

6. Compatibility with the base protocol

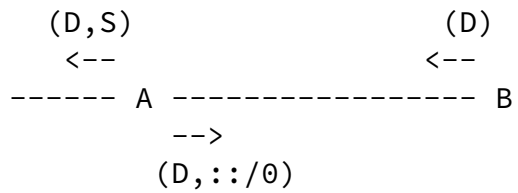
The protocol extension defined in this document is, to a great extent, interoperable with the base protocol defined in [BABEL] (and all of its extensions). More precisely, if non-source-specific routers and source-specific routers are mixed in a single routing domain, Babel's loop-avoidance properties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

However, this extension is encoded using mandatory sub-TLVs, introduced in [BABEL], and therefore is not compatible with the older version of the Babel Routing Protocol [RFC6126]. Consequently, this extension MUST NOT be used with routers implementing RFC 6126, otherwise persistent routing loops may occur.

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 (non source-specific) merely ignores the source prefix information when it receives the update rather than ignoring the whole TLV, and re-announces the route as D. This re-announcement reaches 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 source-specific routes by non-source-specific routers will cause route 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.

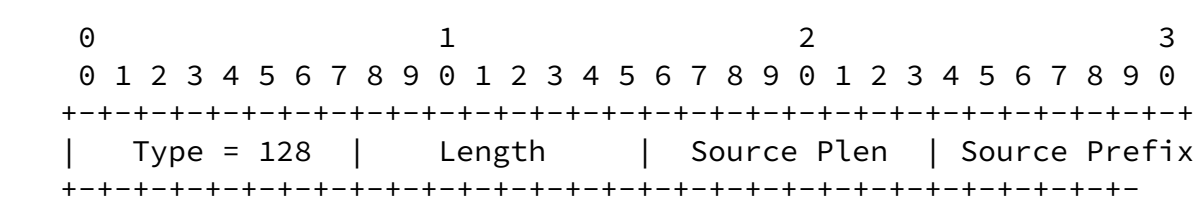
A simple yet sufficient condition for avoiding starvation is to build a connected source-specific backbone that includes all of the edge routers, and announce a (non-source-specific) default route towards the backbone.

7. Protocol Encoding

This extension defines a new sub-TLV used to carry a source prefix: the Source Prefix sub-TLV. It can be used within an Update, a Route Request or a Seqno Request TLV to match a source-specific entry of the Route Table, in conjunction with the destination prefix natively carried by these TLVs.

Since a source-specific routing entry is characterized by a single destination prefix and a single source prefix, a source-specific message contains exactly one Source Prefix sub-TLV. A node MUST NOT send more that one Source Prefix sub-TLV in a TLV, and a node receiving more than one Source Prefix sub-TLV in a single TLV SHOULD ignore this TLV. It MAY ignore the whole packet.

7.1. Source Prefix sub-TLV



Fields:

Type Set to 128 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 contents of the source prefix sub-TLV are interpreted according to the AE of the enclosing TLV.

Note that this sub-TLV is a mandatory sub-TLV. Threfore, as described in Section 4.4 of [BABEL], the whole TLV MUST be ignored if that sub-TLV is not understood (or malformed). Otherwise, routing loops may occur (see Section 6.1).

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]). A wildcard retraction (Update with AE equals to 0) MUST NOT carry a Source Prefix sub-TLV.

Contrary to the destination prefix, this extension does not compress the source prefix attached to Updates. However, compression is allowed for the destination prefix of source-specific routes. As described in Section 4.5 of [BABEL], unextended implementations will correctly update their parser state while otherwise ignoring the whole TLV.

7.3. Source-specific (Route) Request

A source-specific Route Request is a Route Request TLV with a Source Prefix sub-TLV. It prompts the receiver to send an update for a given pair of destination and source prefixes, as described in Section 3.8.1.1 of [BABEL]. A wildcard request (Route Request with AE equals to 0) MUST NOT carry a Source Prefix sub-TLV.

7.4. Source-Specific Seqno Request

A source-specific Seqno Request is a Seqno Request TLV with a Source Prefix sub-TLV. It requests the receiving node to perform the procedure described in Section 3.8.1.2 of [BABEL], but applied to a pair of a destination and source prefix.

8. IANA Considerations

IANA has allocated sub-TLV number 128 for the Source Prefix sub-TLV in the Babel Sub-TLV Numbers registry.

9. Security considerations

The extension defined in this document adds a new sub-TLV to three

TLVs already present in the original Babel protocol, and does not in itself change the security properties of the protocol. However,

source-specific routing gives more control over routing to the sending hosts, which might have security implications (see Section 8 of [DSR]).

10. Acknowledgments

The authors are grateful to Joel Halpern for his help with this document.

11. References

11.1. Normative References

[BABEL] Chroboczek, J., "The Babel Routing Protocol", Internet Draft draft-ietf-babel-rfc6126bis-04, May 2017.

[BCP84] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

11.2. Informative References

[DSR] Lamparter, D. and A. Smirnov, "Destination/Source Routing", Internet Draft draft-ietf-rtgwg-dst-src-routing-06, May 2018.

[RFC6126] Chroboczek, J., "The Babel Routing Protocol (Experimental)", RFC 6126, February 2011.

[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