Applicability of the Babel routing protocol
draft-ietf-babel-applicability-03

Abstract

   Where we argue that, although OSPF and IS-IS are fine protocols,
   there exists a space where the Babel routing protocol (RFC 6126bis)
   is useful.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction and background

   Babel [RFC6126bis] is a routing protocol based on the familiar
   distance-vector algorithm (sometimes known as distributed Bellman-
   Ford) augmented with mechanisms for loop avoidance (there is no
   "counting to infinity") and starvation avoidance. In this document,
   we argue that there exist niches where Babel is useful and that are
   not adequately served by more mature protocols such as OSPF [RFC5340]
   and IS-IS [RFC1195].

1.1. Technical overview of the Babel protocol

   At its core, Babel is a traditional distance-vector protocol based on
   the distributed Bellman-Ford algorithm, similar in principle to RIP
   [RFC2453], but with two obvious extensions: provisions for sensing of
   neighbour reachability, bidirectional reachability and link quality,
   and support for multiple address families (e.g., IPv6 and IPv4) in a
   single protocol instance.

   Algorithms of this class are simple to understand and simple to
   implement, but unfortunately they do not work very well -- they
   suffer from "counting to infinity", a case of pathologically slow
   convergence in some topologies after a link failure. Babel uses a
   mechanism pioneered by EIGRP [DUAL] [RFC7868], known as
   "feasibility", which avoids routing loops and therefore makes
   counting to infinity impossible.

   Feasibility is a conservative mechanism, one that not only avoids all
   looping routes but also rejects some loop-free routes. Thus, it can

lead to a situation known as "starvation", where a router rejects all
routes to a given destination, even those that are loop-free. In
order to recover from starvation, Babel uses a mechanism pioneered by
DSDV [DSDV] and known as "sequenced routes". In Babel, this
mechanism is generalised to deal with prefixes of arbitrary length
and routes announced at multiple points in a single routing domain
(DSDV was a pure mesh protocol, and only dealt with host routes).

In DSDV, the sequenced routes algorithm is slow to react to a
starvation episode. In Babel, starvation recovery is accelerated by
using explicit requests (known as "seqno requests" in the protocol)
that signal a starvation episode and cause a new sequenced route to
be propagated in a timely manner. In the absence of packet loss,
this mechanism is provably complete and clears the starvation in time
proportional to the diameter of the network, at the cost of some
additional signalling traffic.

2. Properties of the Babel protocol

In this section, we describe the properties of the Babel protocol as
well as its known limitations.

2.1. Simplicity and implementability

Babel is a conceptually simple protocol. It consists of a familiar
algorithm (distributed Bellman-Ford) augmented with three simple and
well-defined mechanisms (feasibility, sequenced routes and explicit
requests). Given a sufficiently friendly audience, the principles
behind Babel can be explained in 15 minutes, and a full description
of the protocol can be done in 52 minutes (one microcentury).

An important consequence is that Babel is easy to implement. While
Babel is a young protocol, there exist four independent
implementations, including one that was reportedly written and
debugged in just two nights.

2.2. Robustness

The fairly strong properties of the Babel protocol (convergence, loop
avoidance, starvation avoidance) rely on some rather weak properties
of the network and the metric being used. The most significant are:

o causality: a control message is not received before it has been
   sent (more precisely, the "happens-before" relation is acyclic);

o strict monotonicity of the metric: $M < C + M$;

o left-distributivity of the metric: if M <= M', then
   C + M <= C + M'.

In particular, Babel does not assume a reliable transport, it does
not assume ordered delivery, it does not assume that communication is
transitive, and it does not require that the metric be discrete
(continuous metrics are possible, reflecting for example packet loss
rates). This is in contrast to traditional link-state routing
protocols such as OSPF [RFC5340] or IS-IS [RFC1195], which are
layered over a reliable flooding algorithm and make stronger
requirements on the underlying network and metric.

These weak requirements make Babel a robust protocol:

o robust with respect to bugs: an implementation bug does most
   probably not violate the properties on which Babel relies; in our
   (extensive) experience, bugs tend to slow down convergence or
   cause sub-optimal routing, but do not cause the network to
   collapse;

o robust with respect to unusual networks: an unusual network (non-
   transitive links, unstable metrics, etc.) does most probably not
   violate the assumptions of the protocol;

o robust with respect to novel metrics: no matter how strange your
   metric (continuous, constantly fluctuating, etc.), it does most
   probably not violate the assumptions of the protocol.

These robustness properties have important consequences for the
applicability of the protocol: Babel works (more or less efficiently)
in a wide range of circumstances where traditional routing protocols
give up.

2.3. Extensibility

Babel's packet format has a number of features that make the protocol
extensible (see Appendix C of [RFC6126bis]), and a number of
extensions have been designed to make Babel work better in situations
that were not envisioned when the protocol was initially designed.
The ease of extensibility is not an accident, but a consequence of
the design of the protocol: it is reasonably easy to check whether a
given extension violates the assumptions on which Babel relies.

Remarkably enough, all of the extensions designed to date
interoperate with the base protocol and with each other. This,
again, is a consequence of the protocol design: in order to check the
interoperability of two implementations of Babel, it is enough to

   verify that the interaction of the two does not violate the
   protocol's assumptions.

   Notable extensions deployed to date include:

   o source-specific routing (SADR) [BABEL-SS] allows forwarding to
      take a packet's source address into account, thus enabling a cheap
      form of multihoming [SS-ROUTING];

   o RTT-based routing [BABEL-RTT] minimises link delay, which is
      useful in overlay network (where both hop count and packet loss
      are poor metrics).

   Some other extensions have been designed, but have not seen
   deployment yet (and their usefulness is yet to be demonstrated):

   o frequency-aware routing [BABEL-Z] aims to minimise radio
      interference in wireless networks;

   o ToS-aware routing [BABEL-TOS] allows routing to take a packet's
      ToS marking into account for selected routes without incurring the
      full cost of a multi-topology routing protocol.

2.4. Limitations

   Babel has some undesirable properties that make it suboptimal or even
   unusable in some deployments.

2.4.1. Periodic updates

   The main mechanisms used by Babel to reconverge after a topology
   change are reactive: triggered updates, triggered retractions and
   explicit requests. However, in the presence of heavy packet loss,
   Babel relies on periodic updates to clear pathologies. This reliance
   on periodic updates makes Babel unsuitable in at least two kinds of
   deployments:

   o large, stable networks: since Babel sends periodic updates even in
      the absence of topology changes, in well-managed, large, stable
      networks the amount of control traffic will be reduced by using a
      protocol that relies on a reliable transport (such as OSPF, IS-IS
      or EIGRP);

   o low-power networks: the periodic updates use up battery power even
      when there are no topology changes and no user traffic, which
      makes Babel wasteful in low-power networks.

2.4.2. Full routing table

While there exist techniques that allow a Babel speaker to function
with a partial routing table (e.g., by learning just a default route
or, more generally, performing route aggregation), Babel is designed
around the assumption that every router has a full routing table. In
networks where some nodes are too constrained to hold a full routing
table, it might be preferable to use a protocol that was designed
from the outset to work with a partial routing table (such as AODVv2
[AODVv2], RPL [RFC6550] or LOADng [LOADng]).

2.4.3. Slow aggregation

Babel's loop-avoidance mechanism relies on making a route unreachable
after a retraction until all neighbours have been guaranteed to have
acted upon the retraction, even in the presence of packet loss.
Unless the optional algorithm described in Section 3.5.5 of
[RFC6126bis] is implemented, this entails that a node is unreachable
for a few minutes after the most specific route to it has been
retracted. This delay may make Babel slow to recover from a topology
change in networks that perform automatic route aggregation.

3. Successful deployments of Babel

In this section, we give a few examples of environments where Babel
has been successfully deployed.

3.1. Hybrid networks

Babel is able to deal with both classical, prefix-based ("Internet-
style") routing and flat ("mesh-style") routing over non-transitive
link technologies. Because of that, it has seen a number of
succesful deployments in medium-sized hybrid networks, networks that
combine a wired, aggregated backbone with meshy wireless bits at the
edges. No other routing protocol known to us is similarly robust and
efficient in this particular kind of topology.

Efficient operation in hybrid networks requires the implementation to
distinguish wired and wireless links, and to perform link quality
estimation on wireless links.

3.2. Large scale overlay networks

The algorithms used by Babel (loop avoidance, hysteresis, delayed
updates) allow it to remain stable and efficient in the presence of
unstable metrics, even in the presence of a feedback loop. For this
reason, it has been successfully deployed in large scale overlay

networks, built out of thousands of tunnels spanning continents, where it is used with a metric computed from links' latencies.

This particular application depends on the extension for RTT-sensitive routing [DELAY-BASED].

3.3. Pure mesh networks

While Babel is a general-purpose routing protocol, it has been repeatedly shown to be competitive with dedicated routing protocols for wireless mesh networks [REAL-WORLD] [BRIDGING-LAYERS]. Although this particular niche is already served by a number of mature protocols, notably OLSR-ETX and OLSRv2 [RFC7181] (equipped e.g. with the DAT metric [RFC7779]), Babel has seen a moderate amount of successful deployment in pure mesh networks.

3.4. Small unmanaged networks

Because of its small size and simple configuration, Babel has been deployed in small, unmanaged networks (e.g., home and small office networks), where it serves as a more efficient replacement for RIP [RFC2453], over which it has two significant advantages: the ability to route multiple address families (IPv6 and IPv4) in a single protocol instance, and good support for using wireless links for transit.

4. IANA Considerations

This document requires no IANA actions. [RFC Editor: please remove this section before publication.]

5. Security Considerations

As is the case in all distance-vector routing protocols, a Babel speaker receives reachability information from its neighbours, which by default is trusted. A number of attacks are possible if this information is not suitably protected, either by a lower-layer mechanism or by an extension to the protocol itself (e.g. [RFC7298]).

Implementors and deployers must be aware of the insecure nature of the base protocol, and must take suitable measures to ensure that the protocol is deployed as securely as required by the application.

6. Informational References

   [AODVv2]   Perkins, C., Ratliff, S., Dowdell, J., Steenbrink, L., and
              V. Mercieca, "Ad Hoc On-demand Distance Vector Version 2
              (AODVv2) Routing", draft-ietf-manet-aodvv2-16 (work in
              progress), May 2016.

   [BABEL-RTT]
              Jonglez, B. and J. Chroboczek, "Delay-based Metric
              Extension for the Babel Routing Protocol", draft-jonglez-
              babel-rtt-extension-01 (work in progress), May 2015.

   [BABEL-SS]
              Boutier, M. and J. Chroboczek, "Source-Specific Routing in
              Babel", draft-ietf-babel-source-specific-03 (work in
              progress), August 2018.

   [BABEL-TOS]
              Chouasne, G. and J. Chroboczek, "TOS-Specific Routing in
              Babel", draft-chouasne-babel-tos-specific-00 (work in
              progress), July 2017.

   [BABEL-Z] Chroboczek, J., "Diversity Routing for the Babel Routing
              Protocol", draft-chroboczek-babel-diversity-routing-01
              (work in progress), February 2016.

   [BRIDGING-LAYERS]
              Murray, D., Dixon, M., and T. Koziniec, "An Experimental
              Comparison of Routing Protocols in Multi Hop Ad Hoc
              Networks", Proc. ATNAC 2010, 2010.

   [DELAY-BASED]
              Jonglez, B. and J. Chroboczek, "A delay-based routing
              metric", March 2014, <http://arxiv.org/abs/1403.3488>.

   [DSDV]   Perkins, C. and P. Bhagwat, "Highly Dynamic Destination-
              Sequenced Distance-Vector Routing (DSDV) for Mobile
              Computers", ACM SIGCOMM'94 Conference on Communications
              Architectures, Protocols and Applications 234-244, 1994.

   [DUAL]   Garcia Luna Aceves, J., "Loop-Free Routing Using Diffusing
              Computations", IEEE/ACM Transactions on Networking 1:1,
              February 1993.

   [LOADng]   Clausen, T., Verdiere, A., Yi, J., Niktash, A., Igarashi,
              Y., Satoh, H., Herberg, U., Lavenu, C., Lys, T., and J.
              Dean, "The Lightweight On-demand Ad hoc Distance-vector
              Routing Protocol - Next Generation (LOADng)", draft-
              clausen-lln-loadng-15 (work in progress), January 2017.

[REAL-WORLD]
          Abolhasan, M., Hagelstein, B., and J. Wang, "Real-world
          performance of current proactive multi-hop mesh
          protocols", Asia-Pacific Conference on Communication 2009,
          2009.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and
          dual environments", RFC 1195, December 1990.

[RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, November
          1998.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
          for IPv6", RFC 5340, July 2008.

[RFC6126bis]
          Chroboczek, J. and D. Schinazi, "The Babel Routing
          Protocol", Internet Draft draft-ietf-babel-rfc6126bis-04,
          October 2017.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J.,
          Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur,
          JP., and R. Alexander, "RPL: IPv6 Routing Protocol for
          Low-Power and Lossy Networks", RFC 6550, March 2012.

[RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg,
          "The Optimized Link State Routing Protocol Version 2",
          RFC 7181, April 2014.

[RFC7298] Ovsienko, D., "Babel Hashed Message Authentication Code
          (HMAC) Cryptographic Authentication", RFC 7298,
          DOI 10.17487/RFC7298, July 2014,
          <http://www.rfc-editor.org/info/rfc7298>.

[RFC7779] Rogge, H. and E. Baccelli, "Directional Airtime Metric
          Based on Packet Sequence Numbers for Optimized Link State
          Routing Version 2 (OLSRv2)", RFC 7779,
          DOI 10.17487/RFC7779, April 2016.

[RFC7868] Savage, D., Ng, J., Moore, S., Slice, D., Paluch, P., and
          R. White, "Cisco's Enhanced Interior Gateway Routing
          Protocol (EIGRP)", RFC 7868, DOI 10.17487/RFC7868, May
          2016.

[SS-ROUTING]
          Boutier, M. and J. Chroboczek, "Source-Specific Routing",
          August 2014, <http://arxiv.org/pdf/1403.0445>.

              In Proc. IFIP Networking 2015.

Author's Address

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

   Email: jch@irif.fr