

Babel routing protocol
Internet-Draft
Intended status: Informational
Expires: September 14, 2017

B. Stark
AT&T
March 13, 2017

Babel Information Model
draft-stark-babel-information-model-01

Abstract

This Babel Information Model can be used to create data models under various data modeling regimes (e.g., YANG). It allows a Babel implementation (via a management protocol such as netconf) to report on its current state and may allow some limited configuration of protocol constants.

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 September 14, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Requirements Language	2
1.2.	Notation	3
2.	The Information Model	3
2.1.	Definition of babel-information	3
2.2.	Definition of babel-constants	4
2.3.	Definition of babel-interfaces	5
2.4.	Definition of babel-neighbors	5
2.5.	Definition of babel-csa	6
2.6.	Definition of babel-sources	6
2.7.	Definition of babel-routes	7
3.	References	8
3.1.	Normative References	8
3.2.	Informative References	8
	Author's Address	8

1. Introduction

Babel is a loop-avoiding distance-vector routing protocol defined in RFC 6126 [RFC6126]. Babel Hashed Message Authentication Code (HMAC) Cryptographic Authentication, defined in RFC 7298 [RFC7298], describes a cryptographic authentication mechanism for the Babel routing protocol. This document describes an information model for Babel (including HMAC) that can be used to create management protocol data models (such as a netconf [RFC6241] YANG data model). Other Babel extensions may be included in this document when they become working group drafts.

Due to the simplicity of the Babel protocol and the fact that it is designed to be used in non-professionally administered environments (such as home networks), most of the information model is focused on reporting status of the Babel protocol, and very little of that is considered mandatory to implement (conditional on a management protocol with Babel support being implemented). Some parameters may be configurable; however, it is up to the Babel implementation whether to allow any of these to be configured within its implementation. Where the implementation does not allow configuration of these parameters, it may still choose to expose them as read-only.

1.1. Requirements Language

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 RFC 2119 [RFC2119].

1.2. Notation

This document uses a programming language-like notation to define the properties of the objects of the information model. An optional property is enclosed by square brackets, [], and a list property is indicated by two numbers in angle brackets, <m..n>, where m indicates the minimal number of values, and n is the maximum. The symbol * for n means no upper bound.

2. The Information Model

2.1. Definition of babel-information

```
object {
  string          babel-version;
  int             babel-self-router-id;
  [int           babel-self-seqno;]
  string         babel-cost-comp-algorithms<1..*>;
  babel-constants-obj  babel-constants;
  babel-interfaces-obj  babel-interfaces<0..*>;
  babel-sources-obj     babel-sources<0..*>;
  babel-routes-obj     babel-routes<0..*>;
}babel-information-obj;
```

babel-version: the version of this implementation of the Babel protocol

babel-self-router-id: the router-id used by this instance of the Babel protocol to identify itself

babel-self-seqno: the current sequence number included in route updates for routes originated by this node

babel-cost-comp-algorithm: a set of names of supported cost computation algorithms; possible values include "k-out-of-j", "ETX"

babel-constants: a babel-constants object

babel-interfaces: a set of babel-interface objects

babel-sources: a set of babel-source objects

babel-routes: a set of babel-route objects

2.2. Definition of babel-constants

```
object {
    int          babel-udp-port;
    [int         babel-multicast-group;]
}babel-constants-obj;
```

babel-udp-port: UDP port for sending and listening for Babel messages; MAY be configurable

babel-hello-interval-lossy: Hello Interval default for lossy links in milliseconds; MAY be configurable

babel-hello-interval-lossless: Hello Interval default for lossless links in milliseconds; MAY be configurable

babel-ihu-interval: IHU Interval default as multiples of Hello interval

babel-update-interval: Update Interval default as multiples of Hello interval

babel-ihu-hold-time: IHU Hold Time default as multiples of Hello interval

babel-route-expiry-time: IHU Interval default as multiples of Hello interval

babel-garbage-collection-time: Garbage Collection time default as multiples of Update interval

babel-max-trigger-delay: Maximum delay to wait before sending a triggered update in milliseconds

babel-max-normal-delay: Maximum delay to wait before sending a non-triggered message in milliseconds

babel-ack-limit: Threshold for requesting acknowledgements on an interface (do not request acknowledgements if there are more than this many neighbors on the interface); MAY be configurable

babel-resend-trigger-lossy-limit: Resend limit of triggered updates on lossy links (can this be the same, whether or not acknowledgements are requested?)

babel-resend-trigger-lossless-limit: Resend limit of triggered updates on lossless links (can this be the same, whether or not acknowledgements are requested?)

babel-resend-normal-lossy-limit: Resend limit of normal messages on lossy links

babel-resend-normal-lossless-limit: Resend limit of normal messages on lossless links

2.3. Definition of babel-interfaces

```

object {
  uri                babel-interface-reference;
  [int               babel-interface-seqno;]
  [int               babel-interface-hello-interval;]
  [int               babel-interface-update-interval;]
  boolean            babel-request-trigger-ack;
  boolean            babel-lossy-link;
  [int               babel-external-cost;]
  babel-neighbors-obj babel-neighbors<1..*>;
  [babel-csa-obj     babel-csa<1..*>;]
}babel-interfaces-obj;

```

babel-interface-reference: reference to an interface object as defined by the data model

babel-interface-seqno: the current sequence number in use for this interface

babel-interface-hello-interval: the current hello interval in use for this interface

babel-interface-update-interval: the current update interval in use for this interface

babel-request-trigger-ack: requests acknowledgement of triggered updates (if number of neighbors less than babel-ack-limit); MAY be configurable

babel-lossy-link: indicates (if true) that the link of this interface is considered lossy; MAY be configurable

babel-external-cost: external input to cost of link of this interface (need to determine how to express this);MUST be configurable if implemented

2.4. Definition of babel-neighbors

```

object {
    some address format  babel-neighbor-address;
    string               babel-hello-history;
    int                  babel-txcost;
    int                  babel-hello-seqno;
    int                  babel-neighbor-ihu-interval;
    [int                 babel-rxcost]
}babel-neighbors-obj;

```

babel-neighbor-address: (IPv4 or v6) address the neighbor sends messages from

babel-hello-history: the Hello history (do we want a human readable format?)

babel-txcost: transmission cost value from the last IHU packet received from this neighbor, or FFFF hexadecimal (infinity) if the IHU hold timer for this neighbor has expired

babel-hello-seqno: expected Hello sequence number

babel-neighbor-ihu-interval: current IHU interval for this neighbor

babel-router-id: router-id of the neighbor

babel-rxcost: reception cost calculated for this neighbor

2.5. Definition of babel-csa

```

object {
    string               placeholder;
}babel-csa-obj;

```

placeholder: this section to be filled in, in the future

2.6. Definition of babel-sources

```

object {
    (prefix, plen)      babel-source-prefix;
    int                  babel-source-router-id;
    int                  babel-source-seqno;
    int                  babel-source-metric;
    [int                 babel-source-garbage-collection-time;]
}babel-sources-obj;

```

babel-source-prefix: Prefix (with prefix length)

babel-source-router-id: router-id of the router originating this prefix

babel-source-seqno: last sequence number used by this source

babel-source-metric: this source's feasibility distance

babel-source-garbage-collection-time: garbage-collection timer for this source

2.7. Definition of babel-routes

```
object {
  (prefix, plen)      babel-route-prefix;
  int                 babel-route-router-id;
  int                 babel-route-neighbor;
  int                 babel-route-metric;
  int                 babel-route-seqno;
  ip address          babel-route-next-hop;
  boolean             babel-route-selected;
}babel-routes-obj;
```

babel-route-prefix: Prefix (with prefix length) for which this route is advertised

babel-route-router-id: router-id of the router originating this prefix

babel-route-neighbor: neighbor that advertised this route (is this a router-id ?)

babel-route-metric: the metric with which this route was advertised by the neighbor, or FFFF hexadecimal (infinity) for a recently retracted route

babel-route-seqno: the sequence number with which this route was advertised

babel-route-next-hop: the next-hop address of this route

babel-route-selected: a boolean flag indicating whether this route is selected, i.e., whether it is currently being used for forwarding and is being advertised

3. References

3.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

3.2. Informative References

- [RFC6126] Chroboczek, J., "The Babel Routing Protocol", RFC 6126, DOI 10.17487/RFC6126, April 2011, <<http://www.rfc-editor.org/info/rfc6126>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [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>>.

Author's Address

Barbara Stark
AT&T
Atlanta, GA
US

Email: barbara.stark@att.com