Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 July 2026

T. Przygienda
C. Barth
HPE Juniper Networking
R. Westphal
NetDEF

19 January 2026

Optional IS-IS Fragment Timestamping and Database Fingerprint
draft-rigatoni-lsr-isis-fragment-timestamping-03

Abstract

Many applications in today's networks rely on reliable and timely flooding of link-state information, such as, but not limited to Traffic Engineered networks. If such link-state information is delayed it can be difficult for those applications to adequately fulfill their intended functionality. This document describes extensions to ISIS supporting distribution of fragment origination time. The origination time can be used to aid troubleshooting and/or by the applications themselves to improve their behavior. Additionally, a mechanism is proposed by which the consistency of databases on all routers in the network can be easily verified and a rough metric of diffusion behavior derived.

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 https://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 23 July 2026.

Copyright Notice

Table of Contents

1. Introduction

Many applications in today's networks rely on reliable and timely flooding of link-state information, such as, but not limited to Traffic Engineered networks and advanced telemetry solutions. If such information is delayed during flooding it can be difficult for those applications to adequately fulfill their intended purpose. This document describes extensions to ISIS allowing it to carry the origination time on each fragment. The origination time can be used to aid troubleshooting of large domains and/or by the applications themselves to improve their behavior.

As an example, in the case of Traffic Engineered Networks synchronization of the Traffic Engineering Database (TED) enables the compute nodes to adapt to changes in the network state and/or react to network events in a timely manner. If link state information is delayed during the flooding process this can result in an unsynchronized TED and easily lead to service degradation due to substandard re-optimization of network load. More specifically, in RSVP-TE networks, a TE path computed using a specific snapshot of the TED may be rejected during signaling by a transit node because of bandwidth unavailability on a specific link (link bandwidth information in the snapshot of TED used during computation may not be "current"). When the ingress is subsequently notified of this "error" via RSVP signaling, the link in question is avoided in the subsequent path computation and an alternate path is sought. An

implementation may use a configurable “hold time” to determine how long this link needs to be avoided. The awareness of the distribution delay statistics can be used by implementations to dynamically adapt an appropriate “hold time” for a given TE link (instead of using a blanket topology-wide configuration). Therefore, the origination time proposed in this document is meant to be used by a compute node(s) or by an operator of Traffic Engineered Network to measure any delays incurred in TED synchronization. The awareness of delays in the distribution of information can be incorporated further into algorithms and network tooling to improve the responsiveness and quality of decisions taken.

In a similar vein it is important in large networks when they are in a stable state to measure whether all databases have synchronized properly and derive the delays involved in the propagation of information across the network. This document proposes an extension to the yang schema that allows easy access to such information.

2. Timestamp TLV

This section defines a new, optional TLV that can be present in any fragment. In case of multiple instances of the TLV in a fragment only the first occurrence MUST be used. The semantics of the TLV is the point in time the fragment with the current sequence number has been generated. Its absence signifies that such information is not available due to host of possible issues, one of them lack of clock with synchronization precise enough.

For practical purposes, although desirable, timestamping the moment a fragment is flooded would be preferable but beside practical implementation problems this could generate on different interfaces the same fragment with different content which breaks one of the fundamental tenants of link-state protocols. However, an implementation is free to choose to use, e.g. the moment the fragment is queued for flooding first time rather than the time the version is generated.

To save space the timestamp is following semantically NTP seconds epoch [RFC5905] with the exception of an extra bit in the seconds field to extend the wrap around and carrying only 2^-8 of a second as maximum resolution of the timestamp since this is considered sufficient for link-state purposes. The specification follows further guidelines of [RFC8877] as far as possible.

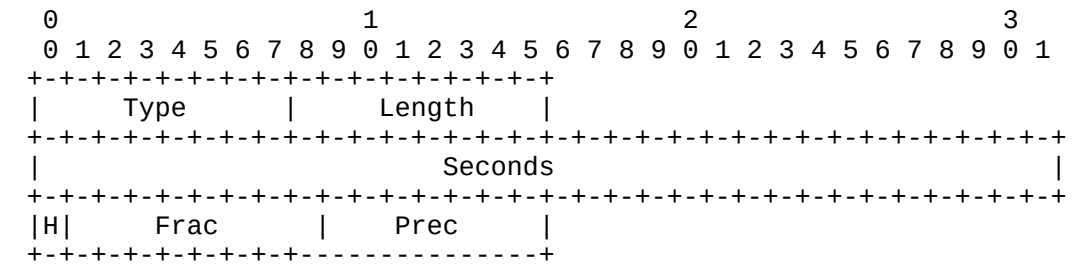

Figure 1

* Type: TBD1

* Length: ...

* Seconds: 4 bytes of number of seconds since the NTP [RFC5905] epoch.

* H(-Bit): 1 bit. Extra high order bit is used to prevent wrap-around in 2036 and pushes it out to 2242. The offset can be constructed in network order ‘HB‘ shifted to left without overflow by 32 bits and the ‘Seconds‘ field OR’ed into the according value.

* Fraction: 8-bits of fraction of the second in units of 2^-8 which is equivalent to 1/256 of a second or roughly 4 msecs resolution.

* Precision: 7 bits indicating the maximum possible slip (either in future or past) of the clock used to generate the timestamp (depending on the synchronization protocol) as 2^Precision where at minimum of the range signifies 2 msec or better precision and the maximum of the range amounts to 256 msec precision or less. A node that cannot achieve this minimum precision required SHOULD NOT advertise the fragment timestamp.

3. Operational and Deployment Considerations

A requirement for the correct interpretation of the additions proposed in this document is an infrastructure capable of synchronizing time across devices involved so the timestamps at the various points of interest become comparable. This could be accomplished by utilizing NTP [RFC5905], Precision Time Protocol (PTP) IEEE Std. 1588 [IEEEstd1588] or 802.1AS [IEEEstd8021AS] designed for bridged LANs. The achieved precision is carried in the timestamp of the fragment.

Though the timestamp can be very useful in deriving measurement of behavior in a deployed IS-IS network, e.g. maximum incurred flooding delays between any pair of nodes, it should not be used in any attempts to modify the behavior of protocol behavior itself such as e.g. influencing flooding rates. A single badly synchronized clock could otherwise change the behavior of parts or even the whole network in unpredictable or even detrimental way.

4. Yang Extensions for Database Fingerprint

To allow for quick validation of database synchronization across all nodes a node can implement yang extensions providing the checksum of all fragments per level. The extension includes the time difference from the last fingerprint change which facilitates measurement of flooding behavior across the network.

Aged out LSPs are excluded from the fingerprint calculation.

Reference checksum algorithm for a fragment is given in Appendix A. The overall fingerprint is the XOR of all fragment checksums.

4.1. Tree for the YANG Data Model

This document uses the graphical representation of data models per [RFC8340].

The following shows the tree diagram of the module:

```
      module: ietf-isis-fragment-timestamping

        augment /rt:routing/rt:control-plane-protocols
                  /rt:control-plane-protocol/isis:isis
                  /isis:database/isis:levels:
          +--rw fingerprint
             +--rw value?         uint64
             +--rw last-update?   uint32
        augment /rt:routing/rt:control-plane-protocols
                  /rt:control-plane-protocol/isis:isis
                  /isis:database/isis:levels/isis:lsp:
          +--rw fragment-origination-time?   yang:date-and-time
```

4.2. YANG Data Model

The following is the YANG module:

```
<CODE BEGINS> file "ietf-isis-fragment-timestamping@2026-01-14.yang"
module ietf-isis-fragment-timestamping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-isis-timestamping";
  prefix isis-fragment-timestamping;

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC 8349: A YANG Data Model for Routing
       Management (NMDA Version)";
  }
  import ietf-isis {
    prefix isis;
    reference
      "RFC 9130: YANG Data Model for the IS-IS Protocol";
  }

  organization
    "IETF LSR - LSR Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/lsr>
     WG List:  <mailto:mpls@ietf.org>

     Author:    Tony Przygienda
                <mailto:prz@juniper.net>
     Author:    Colby Barth
                <cbarth@juniper.net>
     Author:    Renato Westphal
                <renato@netdef.org>
    ";
  description
    "The YANG module augments the base IS-IS YANG data model with
     operational state related to IS-IS fragment timestamping and
     LSDB fingerprinting.

     Copyright (c) 2025 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Revised BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
```

```
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX
        (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
        for full legal notices.

        The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
        NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
        'MAY', and 'OPTIONAL' in this document are to be interpreted as
        described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
        they appear in all capitals, as shown here.";
    reference
      "RFC XXXX: Optional IS-IS Fragment Timestamping";

    revision 2026-01-14 {
      description
        "Initial Version";
      reference
        "RFC XXXX: Optional IS-IS Fragment Timestamping";
    }

    /* LSDB fingerprint */

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/isis:isis/isis:database/"
          + "isis:levels" {
      when "derived-from-or-self(../../../rt:type, 'isis:isis')" {
        description
          "This augment ISIS routing protocol when used";
      }
      container fingerprint {
        description
          "Information about the LSDB fingerprint for this level.";
        leaf value {
          type uint64;
          description
            "A 64-bit fingerprint derived from the set of LSPs at this
             level.

             The fingerprint is computed by XOR-combining a per-LSP
             component value for each LSP whose remaining lifetime is
             non-zero. The per-LSP component value is derived from the
             LSP identifier, the LSP checksum, and the LSP length, as
             specified in Appendix A of RFC XXXX.

             The fingerprint is intended for detecting potential LSDB
             synchronization differences. Different values indicate
```

```
              different LSDB contents; identical values do not guarantee
              equivalence due to possible collisions.";
          }
          leaf last-update {
            type uint32;
            units "seconds";
            description
              "Time elapsed since the fingerprint for this level's LSDB
               was last updated.";
          }
        }
      }

      /* Fragment Timestamp TLV */

      augment "/rt:routing/"
            + "rt:control-plane-protocols/rt:control-plane-protocol"
            + "/isis:isis/isis:database/isis:levels/isis:lsp" {
        when "derived-from-or-self(../../../../rt:type,"
            + "'isis:isis')" {
          description
            "This augment ISIS routing protocol when used";
        }
        description
          "This augments IS-IS protocol LSDB with Timestamp TLV.";
        leaf fragment-origination-time {
          type yang:date-and-time;
          description
            "The time at which this LSP fragment with the
             current sequence number was generated.";
        }
      }
    }
    <CODE ENDS>
```

5. Normative References

[IEEEstd1588]
              IEEE, "IEEE Standard for a Precision Clock Synchronization
              Protocol for Networked Measurement and Control Systems",
              IEEE Standard 1588,
              <https://ieeexplore.ieee.org/document/4579760/>.

[IEEEstd8021AS]
IEEE, "IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks", IEEE Standard 802.1AS, <https://ieeexplore.ieee.org/document/5741898/>.

6. Informative References

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8877] Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", RFC 8877, DOI 10.17487/RFC8877, September 2020, <https://www.rfc-editor.org/info/rfc8877>.

Appendix A. Checksum Reference Algorithm

```
<CODE BEGINS>
    pub(crate) fn lsdb_fingerprint_component(&self) -> u64 {
        let mut result: u64 = 0;
        let system_id: &[u8] = self.lsp_id.system_id.as_ref();
        for i in system_id.iter().chain(&[self.lsp_id.pseudonode]) {
            result <<= 8;
            result ^= *i as u64;
        }
        // fragment checksum
        result ^= (self.cksum as u64) << 48;
        // This is equal to the PDU length advertised by the fragment
        result ^= (self.raw.len() as u64) << 32;
        result
    }
<CODE ENDS>
```

Figure 2

Authors' Addresses

Tony Przygienda
HPE Juniper Networking
Email: antoni.przygienda@hpe.com

Colby Barth
HPE Juniper Networking
Email: colby.barth@hpe.com

Renato Westphal
NetDEF
Email: renato@netdef.org