Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

A. Wang
China Telecom
H. Chen
Individual
J. Dong
Huawei Technologies
C. Lin
New H3C Technologies
G. Mishra
Verizon

7 July 2025

IS-IS Extension for Big TLV
draft-wang-lsr-isis-big-tlv-02

Abstract

The IS-IS routing protocol uses TLV (Type-Length-Value) encoding in a variety of protocol messages. The original IS-IS TLV definition allows for 255 octets of value in maximum. This document proposes a solution to IS-IS extension for encoding the TLV whose value is bigger than 255 octets.

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Type-Length-Value (TLV) encoding of information is widely used in Intermediate System to Intermediate System (IS-IS) routing protocol messages including Link State Protocol Data Units (LSPs). Each TLV defined in [ISO10589] allows for maximum of 255 octets of value (or say payload). This is because the length field of the TLV is one octet, which has 255 as its maximum value. When the size of the value of a TLV of type T (such as the Extended IS Reachability TLV of type 22) is bigger than 255 octets, this TLV is called a Big TLV of type T (or Big-TLV for short). There is no general mechanism for encoding and distributing this Big-TLV in classic IS-IS.

IS-IS has been optionally extended by which permits larger TLV value, in principle up to 65,535 octets due to a two-octet length field. However, the[RFC7356] extensions are not widely deployed, are not backward compatible in the sense that they use a new Protocol Data Unit (PDU) and new LSP types that un-extended implementations will ignore, and in any case do not support values so large they do not fit into a single packet.

This document proposes a simple IS-IS extension for encoding and distributing the Big-TLV whose value parts are bigger and can’t be accommodated in a single TLV. This extension uses a "Container TLV".

2. General Procedure Description

This document describes a method that segments an original Big-TLV into fragmented contents, encapsulates the fragmented contents into one newly defined container TLV, which is transmitted via normal IS-IS PDU to the receiver.

The fragment procedure happen only when the length of original Big-TLV exceeds the normal boundary value(255 bytes), and the results segmented contents should be encapsulated within the newly defined container TLV.

The original Big-TLV includes sub-TLVs, or sub-sub-TLVs. The segment process occurs only at the boundary of sub-TLV, or the boundary of sub-sub-TLV. That is to say, one normal sub-TLV, or sub-sub-TLV shouldn’t be splitted into several segments.

The container TLV is defined in Section 3, which contains a length of fragmented content(normally less than 255 bytes); a packet identifier to identify the original Big-TLV that the fragmented content belongs to; a packet length of the original data packet; a flag field indicating whether the original Big-TLV includes the sub-TLV and/or the sub-sub TLV and the TLV/sub-TLV and sub-sub TLV types of original Big-TLV.

When the receiver receives the segmented container TLV, it will concatenate these container TLVs to reconstruct the orginal Big TLV.

The receiver concatenates these container TLVs only when the above mentioned packet identifier, TLV type, sub-TLV type and sub-sub-TLV type are the same.

The concatenates procedure will end when the receiver receives all the segmented contents, that is to say, the length of concatenate packet match the original packet length that encapsulated in the header of the container TLV.

3. IS-IS Extension for Big TLV

A new TLV, called the Container TLV, is defined. Figure 1 shows the format of the new TLV in the classic [ISO10589] case. This new TLV is used to carry a piece of the value of a Big-TLV of type T.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+        -----------+
|  Type (TBD1)  |    Length     |                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  --+              |
|   Flag   |T2T1|    Type T0    |    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
| Sub-Type T1   |Sub-Sub-Type T2|    |        Container TLV
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |       of type TBD1
|    Length of Type T0/T1/T2    |    |              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |              |
|       Identification          |    |              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  Value of         |
| Piece of value of Big TLV of  |  Container TLV    |
~ type T (less than 247 octets) ~    |              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  --+   -----------+
```

Figure 1: Format of Container TLV

Type (TBD1) field: 1 octet. The type of the Container TLV, its value is assigned by IANA.

Length field: 1 octet. The length of the Value field of the Container TLV.

Value field: contains a Flag field, a Type T0, T1 and T2 field (indicates the top TLV, sub-TLV, sub-sub-TLV type of original Big TLV belongs to respectively), Length of original Type T0/T1/T2, a Identification field and a Piece of value of Big TLV. (Piece field for short).

Flag field: 1 octet. Defines flag for the container TLV. T1(bit 7), T2(bit 6) are defined now. when T1/T2 are all unset, it indicates the Big-TLV belong to one top IS-IS TLV; when only T1 bit is set, it indicates the Big-TLV belong to sub-TLV; when both T1 and T2 bit are set, i indicates the Big-TLV belongs to sub-sub-TLV. T2 bit can be set only when T1 bit is set at the same time.

Type T0、T1、T2 fields: All are 1 octet, Indicates the Type/sub-Type/ sub-sub-Type of the Big-TLV that is being transported in this Container TLV.

Length of Type T0/T1/T2: 2 octets. Actual lengths of Big-TLV.

Identification field: 2 octets. Together with Type(T0/T1/T2), can identify unique Big-TLV that each container TLV belongs to. The sender of Big TLV should keep the identification filed same when it fragments the Big-TLV into several pieces which encapsulated within the container TLV. The receiver of the Big-TLV should assemble the container TLV with the same identification field and Type(T0/T1/T2) into the original Big-TLV.

Piece field: A piece of the value of the Big- TLV of type T0/T1/T2 that is being transported in this Container TLV.

When a node has a Big-TLV of type T0/T1/T2 to be originated, it splits the value of the Big-TLV into a number of pieces, from Piece 1 to Piece n. Each piece from Piece 1 to Piece n is less than 247 octets.

This is illustrated in Figure 2.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type (T)     |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  --+
| Piece 1 (less than 252 octets)|    |
~                               ~    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
| Piece 2 (less than 252 octets)|    |
~                               ~   Bigger than
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   255 octets
~               :               ~    |
~               .               ~    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
| Piece n (less than 252 octets)|    |
~                               ~    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  --+
```

Figure 2: Big-TLV of type T with value field bigger than 255 octets

Each piece carries a subset of entries in the Big-TLV. An entry is an existing sub-TLV or structure. One entry MUST NOT be split across pieces. The Big-TLV is split in entries boundaries, not octets boundaries.

The node originates n TLVs for the Big-TLV of type T. These TLVs are the n new TLVs of type TBD1, each of which has a normal payload. The node advertises each of these TLVs to its neighbors according to the normal IS-IS procedure. Figure 3 shows the encoding of the Big TLV with type T in Figure 2.

```
      0                   1
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+        -----------+
     |  Type (TBD1)  |    Length     |                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  --+              |
     |   Flag   |T2T1|    Type T0    |    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
     | Sub-Type T1   |Sub-Sub-Type T2|    |        Container TLV
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |       of type TBD1
     |    Length of Type T0/T1/T2    |    |              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |              |
     |       Identification          |    |              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  Value of         |
     | Piece 1(less than 247 octets) |  Container TLV    |
     ~                               ~    |              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  --+   -----------+

     ~               :               ~
     ~               .               ~

      0                   1
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+        -----------+
     |  Type (TBD1)  |    Length     |                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  --+              |
     |   Flag   |T2T1|    Type T0    |    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
     | Sub-Type T1   |Sub-Sub-Type T2|    |        Container TLV
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |       of type TBD1
     |    Length of Type T0/T1/T2    |    |              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |              |
     |       Identification          |    |              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  Value of         |
     | Piece n(less than 247 octets) |  Container TLV    |
     ~                               ~    |              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  --+   -----------+
```

Figure 3: Encoding value bigger than 255 octets

For each of the n pieces of the value of the Big-TLV, a Container TLV of type TBD1 carries the piece. Container TLV 1 contains Piece 1 of the value of the Big-TLV; Container TLV 2 contains Piece 2 of the value of the Big-TLV; ...; Container TLV n contains Piece n of the value of the Big-TLV.

If a node supports the extension (i.e., Container TLV), the node understands each piece of the value of the Big-TLV received. Each of the n Container TLVs having Type (T) contains a piece of the Big-TLV value in its Piece field.

4. Split and Glue

This section discusses a couple of ways in which a Big-TLV is split into pieces at sending and the pieces are glued at receiving.

When a TLV of type T is too big at an originating node, this Big-TLV is split into a sequence of pieces. Each piece carries a subset of entries in the Big-TLV. An entry is an existing Sub-TLV or structure.

if there is only one Big-TLV of type T, the node originates container TLVs with type T containing the pieces, and set the identification field the same value in each container TLV.

When there are multiple Big-TLVs of type T, the node originates multiple sequences of container TLVs, with each sequence of the container TLVs has unique different identification field.

For example, suppose that a node has a Big-TLV of type T = 22 as shown in Figure 4. This TLV is too Big and split into two pieces piece 1 and piece 2 at boundary between Sub-TLV K and K+1.

```
         0                   1
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |  Type (T=22)  |    Length     |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ----+-----------+
        |system ID for neighbor 10.2.2.2|      |           |
        +         (6 octets)            +      |           |
        |                               |      |           |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      |           |
        |        Metric (continue)      |    Piece 1       |
        +               +-+-+-+-+-+-+-+-+    < 247 octets  |
        |               |sub-TLVs-length|      |           |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      |       Bigger than
        |          sub-TLV 1            |      |       255 octets
        :             :                 :      |           |
        :          sub-TLV K            :      |           |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ----+           |
        |          sub-TLV K+1          |      |           |
        :             :                 :    Piece 2       |
        :          sub-TLV N            :      |           |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ----+-----------+
```

Figure 4: Example Big TLV of type T=22 with Value Field > 255 Octets

For this Big-TLV of type T = 22, the node originates two container TLV with type T = 22 containing the two pieces (i.e., piece 1 and piece 2) directly. The container TLV is illustrated in Figure 5.

```
    0                   1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  -----------------------+
   |  Type (TBD1)  |    Length     |                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ------------+           |
   |   Flag    |0|0|    Type 22    |             |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+             |           |
   | Sub-Type 0    |Sub-Sub-Type 0 |             |      Container TLV
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+             |     of type TBD1
   |    Length of Type T0/T1/T2    |             |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+             |           |
   |      Identification(ID X)     |             |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -----+      |           |
   |system ID for neighbor 10.2.2.2|      |      |           |
   +         (6 octets)            +      |      |           |
   |                               |      |      |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      |      |           |
   |        Metric (continue)      |      | Value of         |
   +               +-+-+-+-+-+-+-+-+      | Container TLV    |
   |               |sub-TLVs-length|      |      |           |
```

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      |      |           |
   |          sub-TLV 1            |   Piece 1   |           |
   :             :                 :      |      |           |
   :          sub-TLV K            :      |      |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ----+------------------+

    0                   1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  -----------------------+
   |  Type (TBD1)  |    Length     |                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ------------+           |
   |   Flag    |0|0|    Type 22    |             |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+             |           |
   | Sub-Type 0    |Sub-Sub-Type 0 |             |        Container TLV
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+             |       of type TBD1
   |    Length of Type T0/T1/T2    |             |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+             |           |
   |      Identification(ID X)     |             |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -----+      |           |
   |system ID for neighbor 10.2.2.2|      |      |           |
   +         (6 octets)            +      |      |           |
   |                               |      |      |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      |      |           |
   |        Metric (continue)      |      | Value of         |
   +               +-+-+-+-+-+-+-+-+      | Container TLV    |
   |               |sub-TLVs-length|      |      |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+      |      |           |
   |          sub-TLV K+1          |   Piece 2   |           |
   :             :                 :      |      |           |
   :          sub-TLV N            :      |      |           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ----+------------------+
```

Figure 5: Example Encoding of Value Field > 255 Octets

After receiving the container TLV with type T = 22 and the same identification filed(ID X), the node can glue the piece 1 and piece 2 directly accordingly to the F flag that indicate the first first piece and other sequences piece.

Alternatively, when a node has multiple Big-TLVs of type T=22, for each Big-TLV of type T split into a sequence of pieces, the node originates a sequence of container TLVs with type T and unique identification value for each sequence.

After receiving the container TLV with type T = 22, and different sequence of container TLVs, the node can glue the piece 1 and piece 2 for each sequence through the same identification value.

The sliced container TLVs of one Big-TLV SHOULD be encapsulated within one LSP if all the lengths of the container TLV can fit into one LSP. If such condition can't be met, these sliced container TLVs can be put into different LSPs.

If the container TLVs of one Big-TLV locate in different LSPs, the receiver SHOULD wait until it receives all these updated LSPs, then begin the concatenating process and SPF calculation. Such process is similar as the general incremental updates of LSPs.

5. Big-TLV Capability

A new sub-TLV, called Big-TLV Capability sub-TLV, is defined in the Router Capability TLV [RFC7981]. A node advertising this sub-TLV indicates that the node supports the Big-TLV. The format of the sub-TLV is shown in Figure 6.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type (TBD2) |   Length (1)  |     Flags      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Big-TLV Capability sub-TLV

Type (TBD2) field: 1 octet. The type of the Big-TLV Capability sub-TLV, its value is assigned by IANA.

Length field: 1 octet. Its value is 1.

Flags field: 1 octet. The field of flags. No flag is defined now.

A node supporting the Big-TLV MUST advertise this sub-TLV in a Router Capability TLV.

6. Incremental Deployment

For a network using IS-IS, users can deploy the extension for Big-TLV in a part of the network step by step. The network has some nodes supporting the extension (or say new nodes for short) and the other nodes not supporting the extension (or say old nodes for short) before the extension is deployed in the entire network.

The pieces of the Big-TLV, advertised in the Container TLVs, will only be understood by the new nodes and will be ignored by the old nodes. The originator of the Big-TLV MUST consider the above properties when splitting the Big-TLV into multiple pieces.

If all the nodes need to have the same new information for using the new information, every node needs to check if all the nodes support the Big-TLV capability which is distributed by the nodes supporting it. If all the nodes support it, every node uses the new information.

If it is not required that all the nodes must have the same new information for using the new information, the nodes supporting the Big-TLV capability can use the new information, the nodes not supporting the Big-TLV capability ignore the new information.

7. Security Considerations

The mechanism described in this document does not raise any new security issues for the IS-IS protocols.

8. Acknowledgement

TBD

9. IANA Considerations

IANA is requested to make a new allocation in the "IS-IS TLV Codepoint Registry" under the registry name "IS-IS TLV Codepoints" as follows:

Type	Name	IIH	LSP	SNP	Purge	reference
TBD1	Container	Y	Y	N	N	This document

IANA is requested to make a new allocation under the registry name "IS-IS Sub-TLVs for IS-IS Router CAPABILITY TLV" as follows:

Value	Description	IIH	LSP	SNP	Purge	reference
TBD2	Big-TLV Capability	N	Y	N	N	This document

10. References

10.1. Normative References

[ISO10589] ISO, "Information technology -- Telecommunications and information exchange between systems -- Intermediate System to Intermediate System intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)", ISO/IEC 10589:2002, Second Edition, November 2002.

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

[RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <https://www.rfc-editor.org/info/rfc7981>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <https://www.rfc-editor.org/info/rfc7356>.

Authors’ Addresses

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing
102209
China
Email: wangaj3@chinatelecom.cn

Huaimo Chen
Individual
Boston, MA,
United States of America
Email: hchen.ietf@gmail.com

Jie Dong
Huawei Technologies
Beijing
China
Email: jie.dong@huawei.com

Changwang
New H3C Technologies
China
Email: linchangwang.04414@h3c.com

Gyan S. Mishra
Verizon
13101 Columbia Pike
Silver Spring, MD 20904
United States of America
Phone: 301 502-1347
Email: hayabusagsm@gmail.com