Workgroup: Network Working Group
Internet-Draft: draft-chen-lsr-isis-big-tlv-00
Published: 12 March 2023
Intended Status: Standards Track
Expires: 13 September 2023
Authors: H. Chen B. Decraene G. Mishra A. Wang
Futurewei Orange Verizon China Telecom
Z. Li Y. Fan X. Liu
China Mobile Casa Systems IBM Corporation
L. Liu D. Eastlake
Fujitsu Futurewei

IS-IS Extension for Big TLV

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 backward compatible 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

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 13 September 2023.

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 in a backward compatible way.

IS-IS has been optionally extended by [RFC7356] 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 which will rarely be larger than 1,500 octets or, even with jumbo frames, around 9,000 octets.

This document proposes a simple IS-IS extension for encoding and distributing the big TLVs whose value parts are bigger than can be accommodated by the TLV length field and bigger than can be accommodated in a single packet. This extension uses a "Container TLV".

2. 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.

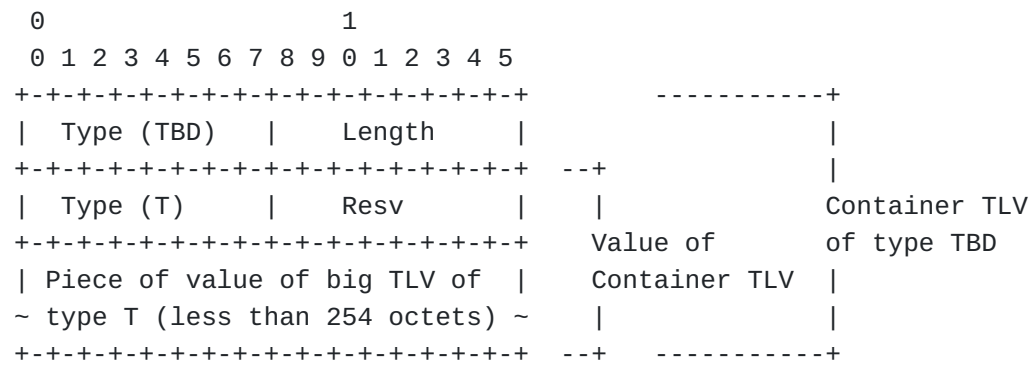

Figure 1: Format of Container TLV

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

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

Value field: contains a Type (T) field, a Resv field, and a Piece of value of big TLV of type T (Piece field for short).

Type (T) field: A one octet field giving the Type of the Big TLV that is being transported in this Container TLV.

Piece field: A piece of the value of the big TLV of type T that is being transported in this Container TLV.

Resv field: One octet that MUST be sent as zero and ignored on receipt.

When a node has a big TLV of type T to be originated, it splits the value of the big TLV into a number of pieces, from Piece 0, Piece 1 to Piece n. The first piece (i.e., Piece 0) is less than 256 octets. Each of the rest pieces from Piece 1 to Piece n is less than 254 octets. This is illustrated in Figure 2.

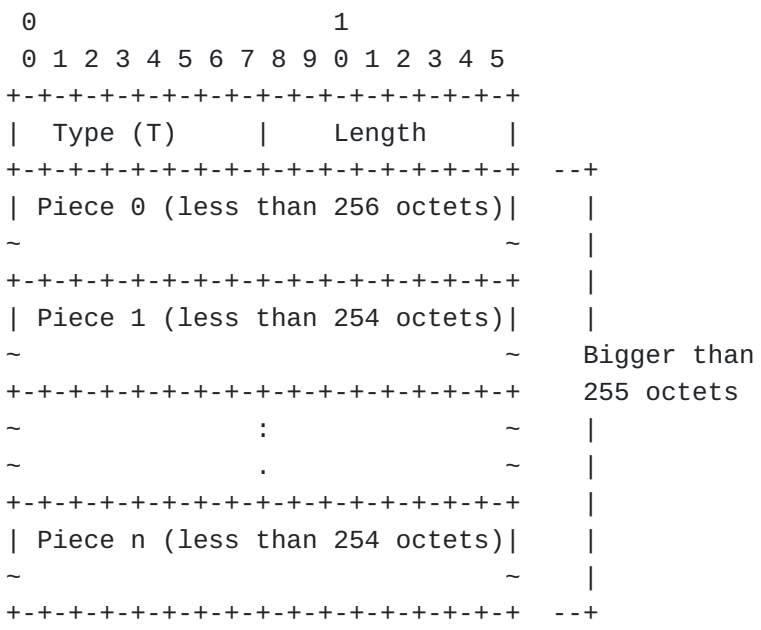

Figure 2: Big TLV of type T with value field bigger than 255 octets

Each piece carries a subset of entries in the big TLV. An entry is an existing sub-TLV or structure.

The node originates n+1 TLVs for the big TLV of type T. These TLVs are the TLV of type T with a normal payload (i.e., the payload less than 256 octets), and n new TLVs of type TBD 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.

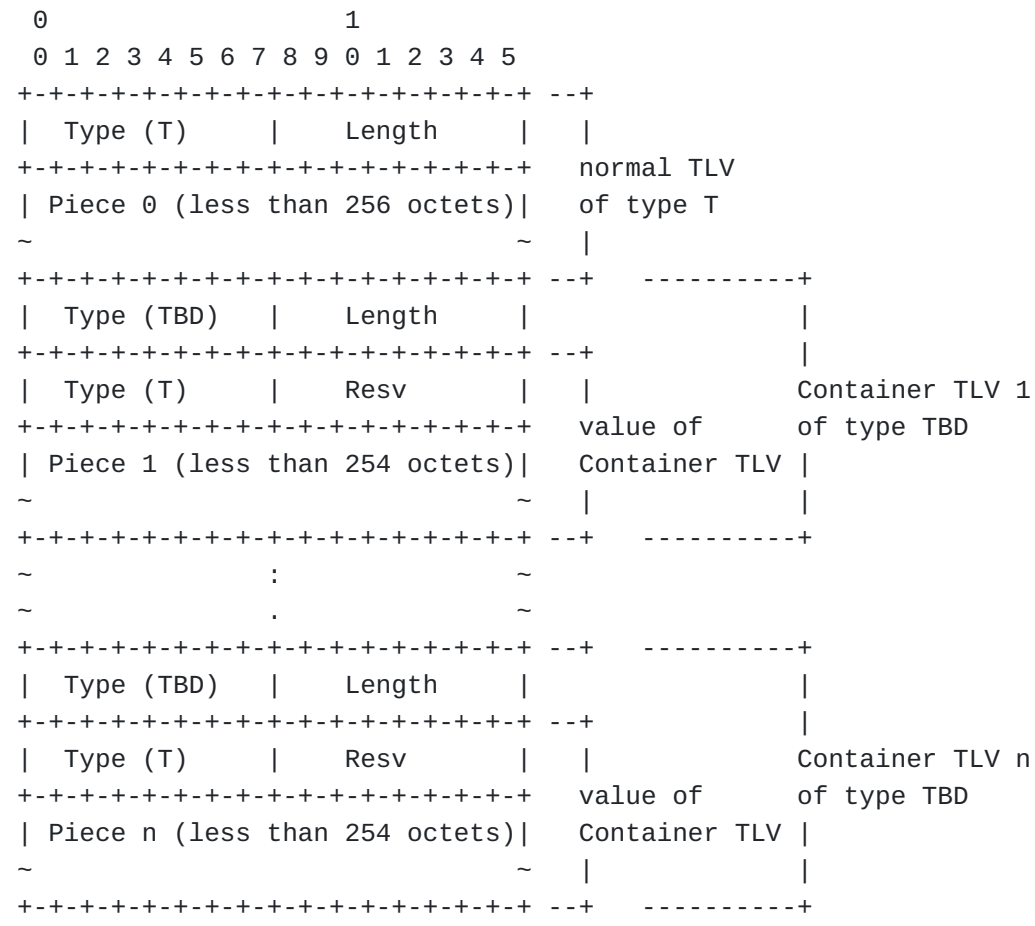

Figure 3: Encoding value bigger than 255 octets

The normal TLV of type T contains Piece 0 (i.e., the first piece) of the value of the big TLV. For each of the rest n pieces of the value of the big TLV, a Container TLV of type TBD 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. The value field in the normal TLV of type T contains Piece 0 of the big TLV value. Each of the n Container TLVs having Type (T) contains a piece of the big TLV value in its Piece field.

3. Split and Glue

This section discusses a couple of ways in which a big TLV is split into pieces at originating/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.

The node originates a native TLV of type T containing the first piece in the sequence. In addition, it originates container TLVs with type T containing the other pieces directly if there is only one big TLV of type T.

When there are multiple (big) TLVs of type T, the node originates multiple native TLVs of type T. For each (big) TLV of type T split into a sequence of pieces, a native TLV of type T containing the first piece in the sequence is originated. The container TLVs with type T containing directly the other pieces are also originated. These container TLVs MUST follow the native TLV in order and between the native TLV and these container TLVs there MUST NOT be any other TLV of type T or any other container TLVs with type T.

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 0 and piece 1 at boundary between Sub-TLV K and K+1.

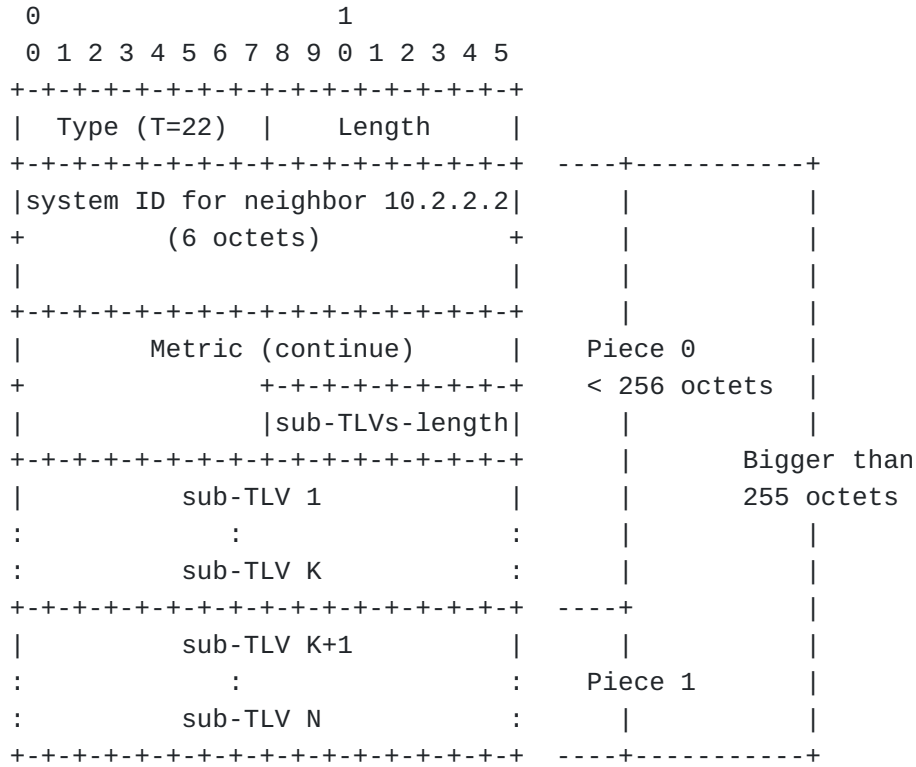

Figure 4: Example Big TLV of type T=22 with Value Field > 255 Octets

For this big TLV of type T = 22, the node originates a native TLV of type T = 22 containing the first piece (i.e., piece 0) and a container TLV with type T = 22 containing the other piece (i.e.,

piece 1) directly. This native TLV and container TLV is illustrated in Figure 5.

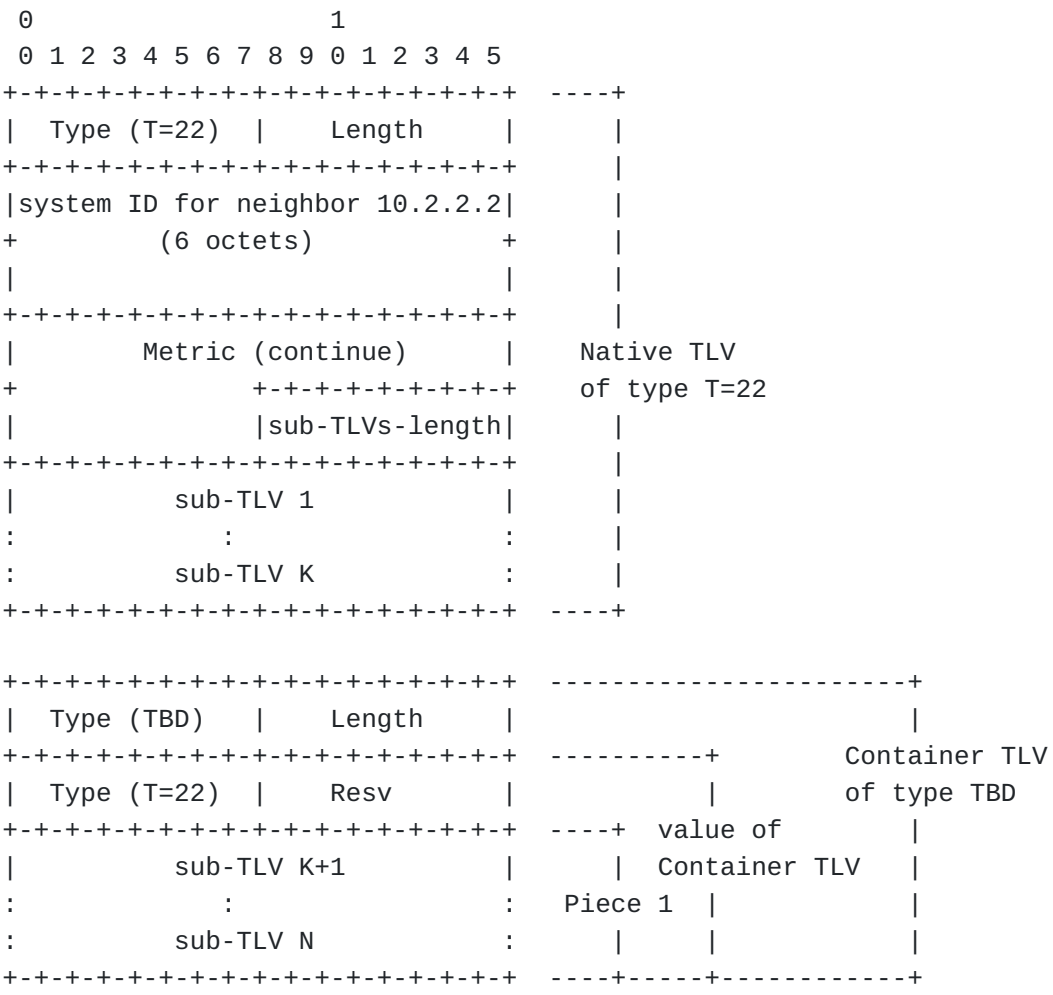

Figure 5: Example Encoding of Value Field > 255 Octets

After receiving the native TLV of type T = 22, followed the container TLV, another node glues the piece 0 and piece 1 directly.

Alternatively, when a node has multiple (big) TLVs of type T, for each (big) TLV of type T split into a sequence of pieces, the node originates a native TLV of type T containing the first piece in the sequence and container TLVs with type T containing the other pieces; where each container TLV also contains a key or header of value in the native TLV (or big TLV). There is no order requirement between the native TLV and the container TLVs.

For example, Figure 6 shows the container TLV containing neighbor ID (i.e., system ID for neighbor) of the value field in the big TLV as the key and piece 1.

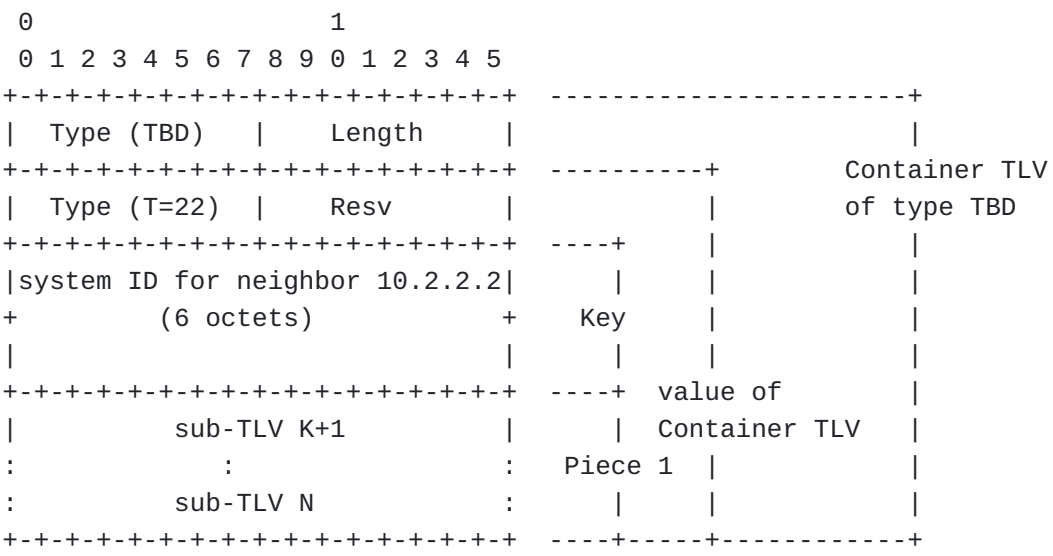

Figure 6: Example Container TLV with Key

After receiving the native TLV of type T = 22, and the container TLV with type T = 22, another node glues the piece 0 and piece 1 through the same neighbor ID in the TLVs as the key.

Figure 7 shows the container TLV containing the header of the value field in the big TLV and piece 1.

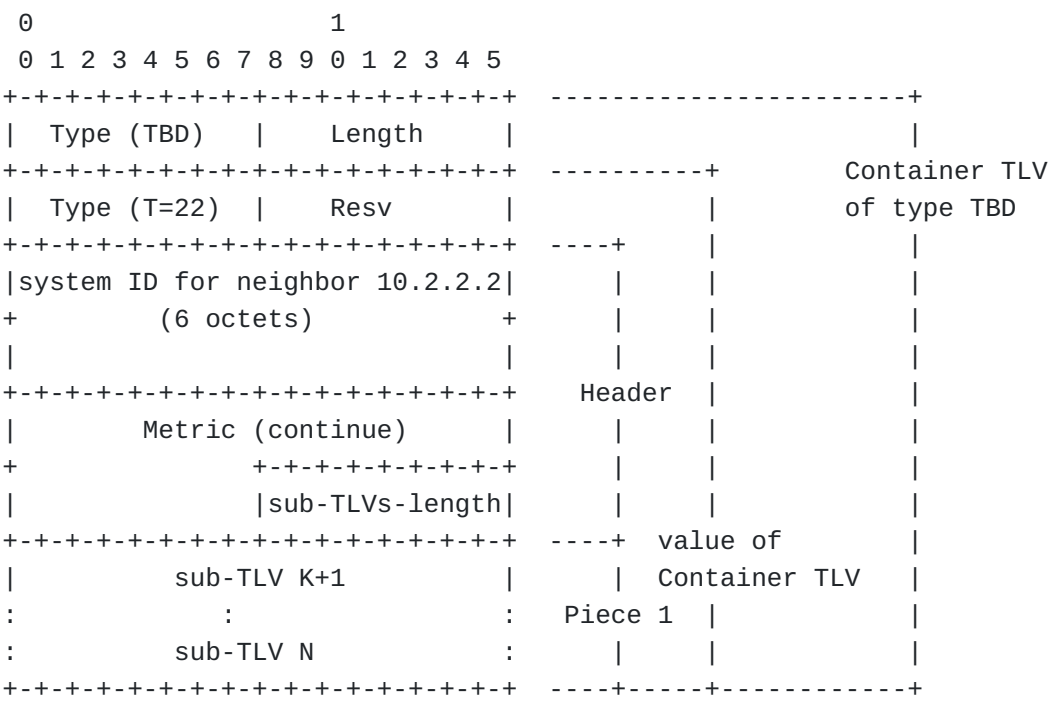

Figure 7: Example Container TLV with Structure Header

After receiving the native TLV of type T = 22, and the container TLV with type T = 22, another node glues the piece 0 and piece 1 through the same header (except for sub-TLVs-length) in the TLVs as the key.

4. 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 first piece of a big TLV, advertised in the native TLV, is backward compatible and will be understood by both old and new nodes. The subsequent 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.

5. Security Considerations

TBD.

6. 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
TBD	Container	Y	Y	N	N	This document

7. References

7.1. 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>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[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>.

7.2. 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>.

Acknowledgments

TBD.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA,
United States of America

Email: Huaimo.chen@futurewei.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Gyan S. Mishra
Verizon
13101 Columbia Pike
Silver Spring, MD 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing
102209
China

Email: wangaj3@chinatelecom.cn

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing
100032
P.R. China

Email: li_zhenqiang@hotmail.com

Yanhe Fan
Casa Systems
United States of America

Email: yfan@casa-systems.com

Xufeng Liu
IBM Corporation
United States of America

Email: xufeng.liu.ietf@gmail.com

Lei Liu
Fujitsu
United States of America

Email: liulei.kddi@gmail.com

Donald E. Eastlake 3rd
Futurewei
2386 Panoramic Circle
Apopka, FL, 32703
United States of America

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com