

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2020

H. Chen
A. Retana
R. Li
Futurewei
A. Kumar S N
RtBrick
N. So

V. Liu

M. Toy
Verizon
L. Liu
Fijitsu
March 9, 2020

Topology-Transparent Zone
draft-chen-isis-ttz-08.txt

Abstract

This document presents a topology-transparent zone in an area. A zone is a block/piece of an area, which comprises a group of routers and a number of circuits connecting them. It is abstracted as a virtual entity such as a single pseudo node or zone edges mess. Any router outside of the zone is not aware of the zone. The information about the circuits and routers inside the zone is not distributed to any router outside of the zone.

Status of This Memo

This Internet-Draft is submitted to IETF 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 September 10, 2020.

Copyright Notice

Copyright (c) 2020 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 (<https://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.

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	3
1.2.	Terminology	3
2.	Requirements	4
3.	Zone Abstraction	4
4.	Topology-Transparent Zone	5
4.1.	Zone as a Single Pseudo Node	5
4.1.1.	An Example of Zone as a Single Node	5
4.1.2.	Zone Leader Election	7
4.1.3.	LS Generation for Zone as a Single Node	8
4.1.4.	Adjacency Establishment	8
4.1.5.	Computation of Routes	10
4.1.6.	Extensions to Protocols	10
4.2.	Zone as Edges Full Mess	12
4.2.1.	Extensions to IS-IS	13
4.3.	Advertisement of LSs	15
4.3.1.	Advertisement of LSs within Zone	15
4.3.2.	Advertisement of LSs through Zone	16
5.	Seamless Migration	16
5.1.	Transfer zone to a Single Node	16
5.2.	Roll Back from Zone as a Single Node	17
6.	Operations	19
7.	Security Considerations	19
8.	IANA Considerations	19
9.	Acknowledgement	19
10.	References	19
10.1.	Normative References	19
10.2.	Informative References	20
	Authors' Addresses	21

1. Introduction

[ISO10589] and [RFC2328] describe two levels of areas, which are level 1 and level 2 areas in IS-IS, and backbone and non-backbone areas in OSPF. There are scalability issues in using areas as the number of routers in a network becomes larger and larger.

Through splitting the network into multiple areas, we may extend the network further. However, dividing a network from one area into multiple areas or from a number of existing areas to even more areas is a very challenging and time consuming task since it is involved in significant network architecture changes.

Furthermore, the services carried by the network may be interrupted while the network is being split from one area into multiple areas or from a few existing areas into even more areas.

These issues can be resolved by using topology-transparent zone (TTZ), which abstracts a zone (i.e., a block/piece of an area) as a single pseudo node or zone edges' mess with minimum efforts and minimum service interruption. Note that a zone can be an area (i.e., the entire piece of an area).

This document presents a topology-transparent zone and describes extensions to IS-IS and OSPF for supporting the topology-transparent zone.

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. Terminology

LSP: A Link State Protocol Data Unit (PDU) in IS-IS.

LSA: A Link State Advertisement in OSPF.

LS: A Link State, which is an LSA in OSPF or LSP in IS-IS.

TTZ: A Topology-Transparent Zone.

Zone: A block or piece of an area. In a special case, a zone is an area (i.e., the entire piece of an area).

Zone External Node: A node outside of a zone.

Zone Internal Node: A node of a zone without any connection to a node outside of the zone.

Zone Edge/Border: A node of a zone connecting to a node outside of the zone.

Zone Node: A zone internal node or a zone edge/border node (i.e., a node of a zone).

Zone Link: A link connecting zone nodes (i.e., a link of a zone).

Zone Neighbor: A node outside of a zone that is a neighbor of a zone edge/border.

2. Requirements

Topology-Transparent Zone (TTZ) may be deployed for resolving some critical issues such as scalability in existing networks and future networks. The requirements for TTZ are listed as follows:

- o TTZ MUST be backward compatible. When a TTZ is deployed on a set of routers in a network, the routers outside of the TTZ in the network do not need to know or support TTZ.
- o TTZ MUST support at least one more levels of network hierarchies, in addition to the hierarchies supported by existing routing protocols.
- o Abstracting a zone as a virtual abstracted entity, which is a single pseudo node or zone edges' mess, SHOULD be smooth with minimum service interruption.
- o De-abstracting (or say rolling back) a virtual abstracted entity to a zone SHOULD be smooth with minimum service interruption.
- o Users SHOULD be able to easily set up an end to end service crossing TTZs.
- o The configuration for a TTZ in a network SHOULD be minimum.
- o The changes on the existing protocols for supporting TTZ SHOULD be minimum.

3. Zone Abstraction

A zone can be abstracted as a single pseudo node or the zone edges' full mess.

When a zone is abstracted as a single pseudo node, this single node is connected to all the neighbors of the zone, and is in the same area as the neighbors.

When a zone is abstracted as its edges' full mess, there is a full mess connections among the edges and each edge is also connected to its neighbors outside of the zone.

4. Topology-Transparent Zone

A Topology-Transparent Zone (TTZ) comprises an Identifier (ID) and a piece/block of an area such as a Level 2 area in IS-IS and a backbone area in OSPF. It is abstracted as a single pseudo node or its edges' full mess. TTZ and zone will be used exchangeably below.

4.1. Zone as a Single Pseudo Node

After a zone is abstracted as a single pseudo node having a pseudo node ID, every node outside of the zone sees a number of links connected to this single node. Each of these links connects a zone neighbor. The link states inside the zone are not advertised to any node outside of the zone. The pseudo node ID may be derived from the zone ID.

4.1.1. An Example of Zone as a Single Node

The figure below shows an example of an area containing a TTZ: TTZ 600.

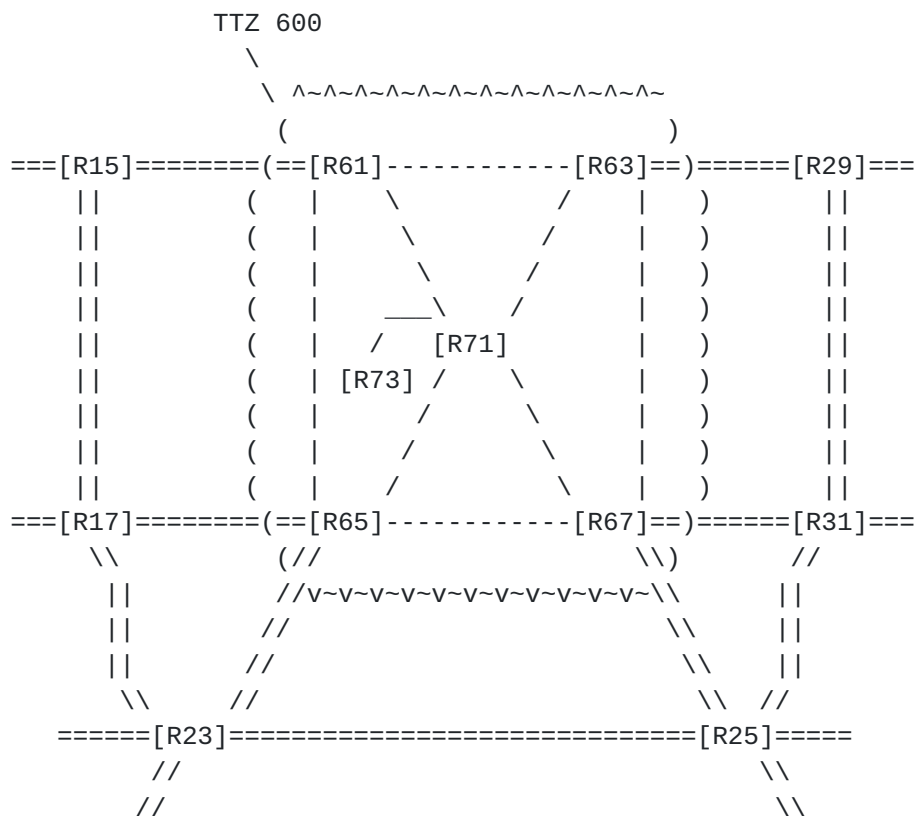


Figure 1: An Example of TTZ 600

The area comprises routers R15, R17, R23, R25, R29 and R31. It also contains TTZ 600, which comprises routers R61, R63, R65, R67, R71 and R73, and the circuits connecting them.

There are two types of routers in a TTZ: TTZ internal routers and TTZ edge/border routers. A TTZ internal router is a router inside the TTZ and its adjacent routers are inside the TTZ. A TTZ edge/border router is a router inside the TTZ and has at least one adjacent router that is outside of the TTZ.

The TTZ in the figure above comprises four TTZ edge/border routers R61, R63, R65 and R67. Each TTZ edge/border router is connected to at least one router outside of the TTZ. For instance, router R61 is a TTZ edge/border router since it is connected to router R15, which is outside of the TTZ.

In addition, the TTZ comprises two TTZ internal routers R71 and R73. A TTZ internal router is not connected to any router outside of the TTZ. For instance, router R71 is a TTZ internal router since it is not connected to any router outside of the TTZ. It is just connected to routers R61, R63, R65, R67 and R73 inside the TTZ.

A TTZ MUST hide the information inside the TTZ from the outside. It MUST NOT directly distribute any internal information about the TTZ to a router outside of the TTZ.

For instance, the TTZ in the figure above MUST NOT send the information about TTZ internal router R71 to any router outside of the TTZ in the routing domain; it MUST NOT send the information about the circuit between TTZ router R61 and R65 to any router outside of the TTZ.

From a router outside of the TTZ, a TTZ is seen as a single node (refer to the Figure below). For instance, router R15, which is outside of TTZ 600, sees TTZ 600 as a single node R_z , which has normal connections to R15, R29, R17 and R23, R25 and R31.

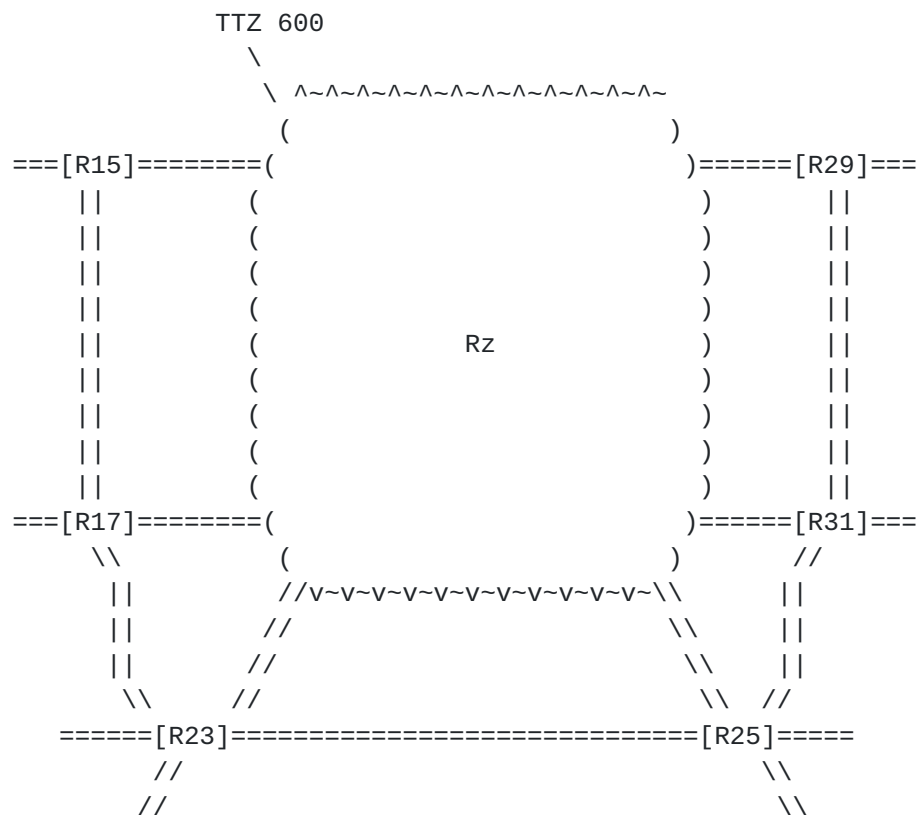


Figure 2: TTZ 600 as Single Node Rz

4.1.2. Zone Leader Election

A node in a zone is elected as a leader for the zone, which is the node with the highest priority (and the highest node ID when there are more than one nodes having the same highest priority) in the zone. The leader election mechanism described in

[I-D.ietf-lsr-dynamic-flooding] may be used to elect the leader for the zone.

4.1.3. LS Generation for Zone as a Single Node

The leader for the zone originates a LS (i.e., a LSP in IS-IS and a LSA in OSPF) for the zone as a single Pseudo node and sends it to its neighbors. This is the only LS to be flooded to the nodes outside of the zone.

The LS comprises all the links connecting the zone neighbors. The LS ID is the ID of the Pseudo node for the zone. The Source ID or Advertising Node/Router ID is the ID of the Pseudo node.

In addition, the LS may contain the stub links for the routes such as the loopback addresses inside the zone to be accessed by zone external nodes (i.e., nodes outside of the zone).

4.1.4. Adjacency Establishment

A zone edge node, acting as a single Pseudo node for the zone, forms an adjacency with a node outside of the zone in a way described below.

Case 1 for a new adjacency (i.e., no adjacency exists between the edge and the node outside of the zone also called zone neighbor):

The edge node originates and sends the zone neighbor every protocol packet such as Hello, which contains the Pseudo node ID as Source ID.

When the edge node synchronizes its link state database (LSDB) with the zone neighbor, it sends the zone neighbor the information about all the link states except for the link states belonging to the zone that are hidden from any node outside of the zone.

At the end of the LSDB synchronization, the LS for the zone as the single pseudo node is originated by the zone leader and distributed to the zone neighbor. This LS contains the links connecting all the zone neighbors, including this newly formed zone neighbor.

Case 2 for an existing adjacency (i.e., an adjacency already exists between the zone edge and the zone neighbor):

The zone edge sends Hellos to the zone neighbor with additional information, including a flag T-bit set to one and a TLV with the Pseudo node ID. This information requests the zone neighbor to transfer the existing adjacency to the new adjacency smoothly through working together with the zone edge in following steps.

Zone Edge	Zone Neighbor
(Transfer Zone to Pseudo Node)	
Hello(T=1, Pseudo ID)	
	-----> OK for Transfer Adjacency
	Hello(T=1, Pseudo ID)
Remote Ready for Transfer	<-----
Hello(Source=Pseudo ID)	
Start Transfer	-----> Transfer to New Adjacency
	Hello
Transfer to New Adjacency	<-----
	. . .

Step 1: When "Transfer Zone to Pseudo Node" is triggered, the zone edge sends the zone neighbor a Hello containing additional information T=1 and Pseudo node ID.

Step 2: After receiving the Hello with T=1 and Pseudo node ID from the zone edge, the zone neighbor sends the zone edge a Hello with T=1 and Pseudo node ID, which means ok for transfer to the new adjacency.

Step 3: The edge sends the zone neighbor a Hello containing the Pseudo node ID as Source ID after receiving the Hello with T=1 and Pseudo node ID from the zone neighbor, which starts to transfer to the new adjacency.

Step 4: The zone neighbor changes the existing adjacency to the new adjacency after receiving the Hello containing the Pseudo node ID as Source ID from the zone edge; and sends the zone edge a Hello without the additional information, which means that it transferred to the new adjacency.

Step 5: The zone edge changes the existing adjacency to the new adjacency after receiving the Hello without the additional information from the zone neighbor; and continues to send the zone neighbor a Hello containing the Pseudo node ID as Source ID. At this point, the old adjacency is transferred to the new one.

For the zone neighbor, changing the existing adjacency to the new one includes:

- o Changing the existing adjacency ID from the edge node ID to the Pseudo node ID through either removing the existing adjacency and adding a new adjacency with the Pseudo node ID or just changing

the existing adjacency ID from the edge node ID to the Pseudo node ID,

- o Removing the link to the zone edge node from its LS and adding a new link to the Pseudo node (or just changing the link to the edge node to the link to the Pseudo node in its LS), and
- o Continuing sending the zone edge Hellos without additional information.

For the zone edge, changing the existing adjacency to the new one includes:

- o Keeping the link to the zone neighbor in its LS, and
- o Continuing sending the zone neighbor Hellos containing the Pseudo node ID as Source ID.

4.1.5. Computation of Routes

After a zone edge migrates to zone as a pseudo node, it computes the routes (i.e., shortest paths to the destinations) in the zone in the same way as that described in [RFC 2328](#). That is, it computes the routes in the normal way using the zone topology (i.e., the topology of the zone without the pseudo node).

For the routes outside of the zone, it computes them using the zone topology, the topology outside of the zone without the pseudo node and the connections between each zone edge and its zone neighbor.

After a zone internal node migrates to zone as a pseudo node, it computes the routes using the zone topology, the topology outside of the zone without the pseudo node and the connections between each zone edge and its zone neighbor.

4.1.6. Extensions to Protocols

The following TLVs are defined in IS-IS and OSPF.

- o Adjacent Node ID TLV: containing an adjacent node ID, to which an adjacency is transferred or rolled back. In case of transfer, the TLV contains the Pseudo node ID; in case of roll back, the TLV contains the edge node ID.
- o Zone ID TLV: containing an zone ID.
- o Zone Options TLV: containing one of some operation options.

In addition, two new flag bits are defined in a TLV. In OSPF, this TLV is the Extended Options and Flags (EOF) TLV of LLS Type 1, which is defined in OSPF Link-Local Signaling [RFC5613]. In IS-IS, this TLV may be the IS-IS Flooding Request TLV, which is defined in Dynamic Flooding on Dense Graphs [I-D.ietf-lsr-dynamic-flooding].

- o T-bit: Short for Transfer Adjacency bit. The T-bit set to one indicates a request for transferring to a new 'virtual' adjacency from the existing adjacency and the new adjacency is identified by the Pseudo node ID (or say abstract node ID), which is included in a TLV, called Adjacent Node ID TLV.
- o N-bit: Short for Roll Back to Normal Adjacency bit. The N-bit set to one indicates a request for rolling back to a Normal adjacency from the existing 'virtual' adjacency and the normal adjacency is identified by the edge node ID, which is included in an Adjacent Node ID TLV.

4.1.6.1. Extensions to IS-IS

The format of Adjacent Node ID TLV is illustrated below.

```

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 (TBD) | Length (6) | Pseudo/Edge Node ID |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Pseudo/Edge Node ID (Continue) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The format of Zone ID TLV is illustrated below.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|ZoneID TLV Type| TLV-Length (8)| Zone ID |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Zone ID (Continue) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Reserved (MUST be zero) |E|Z|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

E = 1: Indicating a node is a zone edge node

Z = 1: Indicating a node has migrated to Zone as an abstracted entity

When a zone node originates a zone LS containing a zone ID TLV, it MUST set flag E to 1 in the zone ID TLV if it is a zone edge node and to 0 if it is a zone-internal node. It MUST set flag Z to 1 after it

has migrated to zone as an abstracted entity and to 0 before it migrates zone to the abstracted entity or after it rolls back from zone as an abstracted entity.

The format of Zone Options TLV is illustrated below.

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|ZoneOP TLV Type| TLV-Length (2)|  OP | Reserved (MUST be zero) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

OP Value	Meaning (Operation)
0x001 (T):	Advertising Zone Topology Information for Migration
0x010 (M):	Migrating Zone to an Abstracted Entity
0x011 (N):	Advertising Normal Topology Information for Rollback
0x100 (R):	Rolling Back from the Abstracted Entity

An OP field of 3 bits is defined. It may have a value of 0x001 for T, 0x010 for M, 0x011 for N, or 0x100 for R, which indicates one of the four operations above. When any of the other values is received, it is ignored. The details on these OPs are described in OSPF Topology-Transparent Zone [[RFC8099](#)].

4.1.6.2. Extensions to OSPF

The format of Adjacent Node ID TLV in OSPF is illustrated below.

```

      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 (TBD)              |      Length (4)              |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                               |
|      Pseudo/Edge Node ID (i.e., Pseudo/Edge Router ID)      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The Zone ID TLV and Zone Options TLV in OSPF are defined in OSPF Topology-Transparent Zone [[RFC8099](#)].

4.2. Zone as Edges Full Mess

OSPF Topology-Transparent Zone [[RFC8099](#)] describes the zone as edges' full mess and the extensions to OSPF for supporting zone as edges' full mess. Based on these extensions, IS-IS is extended by a few new TLVs or Sub-TLVs.

4.2.1. Extensions to IS-IS

4.2.1.1. Zone TLV

A new TLV, which is called IS-IS Zone TLV, may be added into an LSP or a Hello PDU for a zone node. It has the following format.

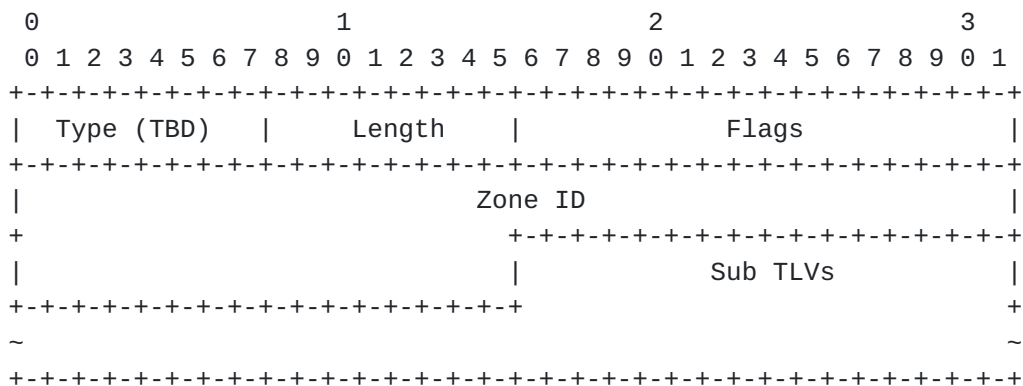
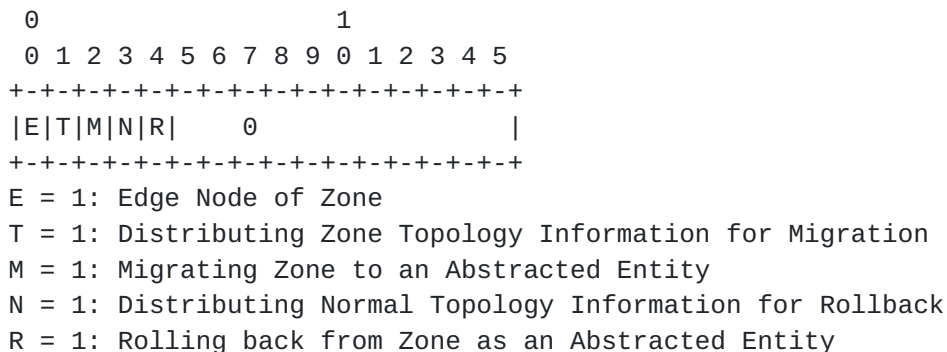


Figure 3: IS-IS Zone TLV

A IS-IS Zone TLV has 1 byte of Type, 1 byte of Length of the value field of the TLV, 2 bytes of Flags and 6 bytes of Zone ID. A IS-IS Zone TLV in an LSP may contains a number of Sub TLVs and have Flags defined as follows.



When a node in a zone receives a CLI command triggering zone information distribution for migration, it updates its LSP by adding a IS-IS Zone TLV with T set to 1. When a node in a zone receives a CLI command activating migration zone to an abstracted entity, it sets M to 1 in the Zone TLV in its LSP.

Two new sub-TLVs are defined, which may be added into a IS-IS Zone TLV in an LSP. One is Zone IS Neighbor sub-TLV, or Zone ISN sub-TLV for short. The other is Zone ES Neighbor sub-TLV, or Zone ESN sub-TLV for short. A Zone ISN sub-TLV contains the information about a

number of Zone IS neighbors connected to a zone edge router. It has the format below.

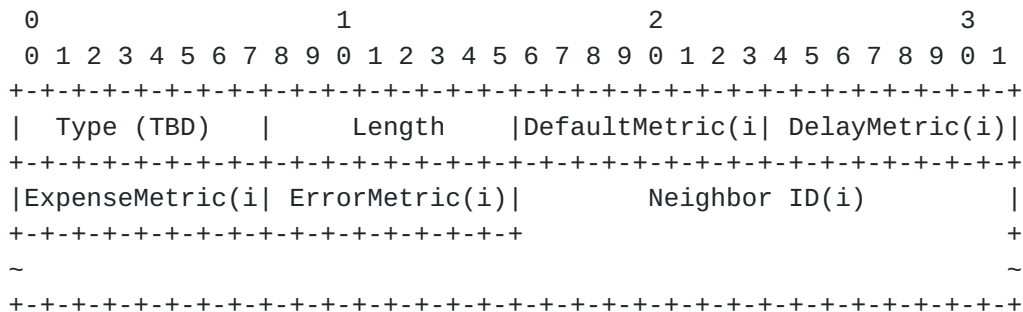


Figure 4: Zone ISN Sub TLV

A Zone ISN Sub TLV has 1 byte of Type, 1 byte of Length of $n \cdot (\text{IDLength} + 4)$, which is followed by n tuples of Default Metric, Delay Metric, Expense Metric, Error Metric and Neighbor ID.

A Zone ESN sub-TLV contains the information about a number of Zone ES neighbors connected to a zone edge node. It has the format below.

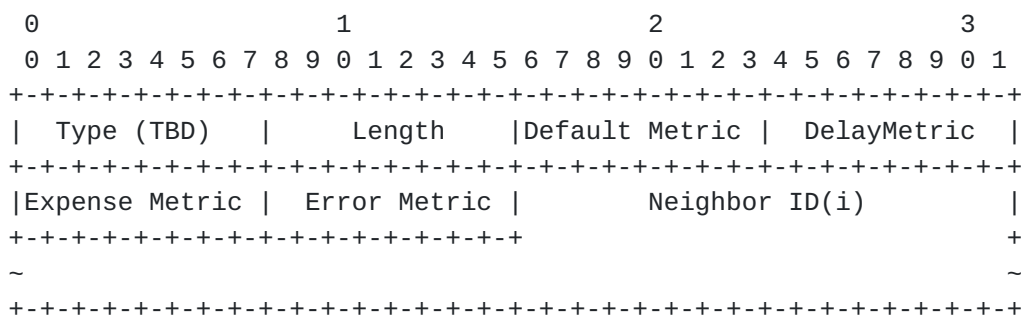


Figure 5: Zone ESN Sub TLV

4.2.1.2. Updating LSPs for Zone

A zone internal node adds a IS-IS Zone TLV into its LSP after it receives an LSP containing a IS-IS Zone TLV with T = 1 or a CLI command triggering zone information distribution for migration. The TLV has a zone ID set to the ID of the zone and E bit in Flags set to 0 indicating zone internal node. The node floods its LSP to its neighbors in the zone.

When a node inside the zone receives an LSP containing a IS-IS Zone TLV from a neighboring node in the zone, it stores the LSP and floods the LSP to the other neighboring nodes in the zone.

For every zone edge node, it updates its LSP in three steps and floods the LSP to all its neighbors.

At first, the zone edge node adds a IS-IS Zone TLV into its LSP after it receives an LSP containing a IS-IS Zone TLV with T = 1 or a CLI command triggering zone information distribution for migration. The TLV has a zone ID set to the ID of the zone, E bit in Flags set to 1 indicating zone edge node and a Zone ISN Sub TLV. The Sub TLV contains the information about the zone IS neighbors connected to the zone edge node. In addition, the TLV may has a Zone ESN Sub TLV comprising the information about the zone end systems connected to the zone edge node.

Secondly, it adds each of the other zone edge nodes as an IS neighbor into the Intermediate System Neighbors TLV in the LSP after it receives an LSP containing a IS-IS Zone TLV with M = 1 or a CLI command activating migration zone to an abstracted entity. The metric to the neighbor is the metric of the shortest path to the edge node within the zone.

In addition, it adds a Prefix Neighbors TLV into its LSP. The TLV contains a number of address prefixes in the zone to be reachable from outside of the zone.

And then it removes the IS neighbors corresponding to the IS neighbors in the Zone TLV (i.e., in the Zone ISN sub TLV) from Intermediate System Neighbors TLV in the LSP, and the ES neighbors corresponding to the ES neighbors in the Zone TLV (i.e., in the Zone ESN sub TLV) from End System Neighbors TLV in the LSP. This SHOULD be done after it receives the LSPs for virtualizing zone from the other zone edges for a given time.

4.3. Advertisement of LSs

LSs can be divided into a couple of classes according to their Advertisements. The first class of LSs is advertised within a zone. The second is advertised through a zone.

4.3.1. Advertisement of LSs within Zone

Any LS about a link state in a zone is advertised only within the zone. It is not advertised to any router outside of the zone. For example, a router LS generated for a zone internal router is advertised only within the zone.

Any network LS generated for a broadcast network in a zone is advertised only within the zone. It is not advertised outside of the zone.

After migrating to zone as a single pseudo node or edges' full mess, every zone edge MUST NOT advertise any LS belonging to the zone or any information in a LS belonging to the zone to any node outside of the zone. The zone edge determines whether an LS is about a zone internal link state by checking if the advertising router of the LS is a zone internal router.

For any zone LS originated by a node within the zone, every zone edge node MUST NOT advertise it to any node outside of the zone.

4.3.2. Advertisement of LSs through Zone

Any LS about a link state outside of a zone received by a zone edge is advertised using the zone as transit. For example, when a zone edge node receives an LS from a node outside of the zone, it floods the LS to its neighbors both inside and outside of the zone. This LS may be any LS such as a router LSA that is advertised within an OSPF area.

The nodes in the zone continue to flood the LS. When another zone edge receives the LS, it floods the LS to its neighbors both inside and outside of the zone.

5. Seamless Migration

This section presents the seamless migration between a zone and its single pseudo node. The seamless migration between a zone and its edges' full mess for IS-IS is similar to that described in OSPF Topology-Transparent Zone [[RFC8099](#)] for OSPF.

5.1. Transfer zone to a Single Node

After transfer a Zone to a Node is triggered, the zone is abstracted as a single Pseudo node in two steps:

Step 1: Every zone edge node works together with each of its zone neighbor nodes to smoothly transfer the existing adjacency between the zone edge and the zone neighbor to a new adjacency between the Pseudo node and the neighbor node in the way described in [Section 4.1.4](#) for Adjacency Establishment procedure for case 2.

Step 2: The zone leader originates a LS for the Pseudo node after receiving the updated LSs originated by all the zone neighbor nodes, where the updated LSs contain all the zone neighbors. Every zone edge does not send any LS inside the zone to any zone neighbors.

5.2. Roll Back from Zone as a Single Node

After roll back from Zone as a Node is triggered, rolling back is done in two steps:

Step 1: Using the procedure described in the following, every zone edge rolls back the existing virtual adjacency between the edge node acting as the Pseudo node and the zone neighbor node to a normal adjacency between the edge node and the neighbor.

Step 2: The zone leader may flush the LS for the Pseudo node. Every zone edge sends Hello and other packages such as CSNP in IS-IS to its zone neighbors, where the packages contain the edge node ID as Source ID.

The procedure below smoothly rolls back the existing virtual adjacency between the edge node acting as the Pseudo node and the zone neighbor node to a normal adjacency between the edge node and the neighbor node.

The edge node sends the neighbor node Hellos with additional information, including a flag N-bit set to one and a TLV with the edge node ID such as the Adjacent Node ID TLV with the edge node ID. This information requests the neighbor node to roll back the existing virtual adjacency to the normal adjacency smoothly through working together with the edge node.

The following steps will roll back the existing virtual adjacency to the normal one:

zone Edge		zone Neighbor
(Roll Back to		
Normal Adjacency)	Hello (N=1, Edge ID)	
	----->	OK to Roll Back to
		Normal Adjacency
	Hello (N=1, Edge ID)	
Remote Ready for	<-----	
Rolling Back		
	Hello(Source=Edge ID)	
Start Roll Back	----->	Roll Back to
		Normal Adjacency
	Hello	
Roll Back to	<-----	
Normal Adjacency	. . .	

Step 1: When "Roll Back from Zone as a Node" is triggered, the edge node sends the neighbor node a Hello with the additional

information N=1 and Edge ID as normal adjacency ID in order to roll back to the normal adjacency from the virtual adjacency.

Step 2: After receiving the Hello with the additional information from the edge node, the neighbor node sends the edge node a Hello with the additional information (i.e., N=1 and Edge ID as normal adjacency ID), which means ok for rolling back to the normal adjacency.

Step 3: The edge sends the neighbor a Hello containing the edge node ID as Source ID after receiving the Hello with the additional information from the neighbor, which starts to roll back to the normal adjacency.

Step 4: The neighbor node changes the existing adjacency to the normal adjacency after receiving the Hello containing the edge node ID as Source ID from the edge node; and sends the edge node a Hello without the additional information, which means that it rolled back to the normal adjacency.

Step 5: The edge node changes the existing adjacency to the normal adjacency after receiving the Hello without the additional information from the neighbor node; and continues to send the neighbor Hello containing the edge node ID as Source ID. At this point, the virtual adjacency is rolled back to the normal adjacency.

For the neighbor node, changing the existing virtual adjacency to the normal one includes:

- o Changing the existing adjacency ID from the Pseudo node ID to the edge node ID through either removing the existing adjacency and adding a new adjacency with the edge node ID or just changing the existing adjacency ID from the Pseudo node ID to the edge node ID,
- o Removing the link to the Pseudo node from its LS and adding a new link to the edge node (or just changing the link to the Pseudo node to the link to the edge node in its LS), and
- o Continuing sending the edge node Hellos without additional information.

For the edge node, changing the existing virtual adjacency to the normal one includes:

- o Sending its LS to the neighbor, and

- o Continuing sending the neighbor node Hellos containing the edge node ID as Source ID without additional information.

6. Operations

The Operations on TTZ described in OSPF Topology-Transparent Zone [[RFC8099](#)] are for Zone as Edges Full Mess in OSPF. They can be used for Zone as Edges Full Mess in IS-IS. They can also be used for Zone as a Single Pseudo Node in OSPF and IS-IS.

7. Security Considerations

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

8. IANA Considerations

9. Acknowledgement

The authors would like to thank Acee Lindem, Abhay Roy, Christian Hopps, Dean Cheng, Russ White, Tony Przygienda, Wenhui Lu, Lin Han, Kiran Makhijani, Padmadevi Pillay Esnault, and Yang Yu for their valuable comments on TTZ.

10. References

10.1. Normative References

[I-D.ietf-lsr-dynamic-flooding]

Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., and S. Dontula, "Dynamic Flooding on Dense Graphs", [draft-ietf-lsr-dynamic-flooding-04](#) (work in progress), November 2019.

[I-D.ietf-spring-segment-routing]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [draft-ietf-spring-segment-routing-15](#) (work in progress), January 2018.

[ISO10589]

International Organization for Standardization, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Nov. 2002.

- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", [RFC 1195](#), DOI 10.17487/RFC1195, December 1990, <<https://www.rfc-editor.org/info/rfc1195>>.
- [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>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, [RFC 2328](#), DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC5029] Vasseur, JP. and S. Previdi, "Definition of an IS-IS Link Attribute Sub-TLV", [RFC 5029](#), DOI 10.17487/RFC5029, September 2007, <<https://www.rfc-editor.org/info/rfc5029>>.
- [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>>.
- [RFC5613] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", [RFC 5613](#), DOI 10.17487/RFC5613, August 2009, <<https://www.rfc-editor.org/info/rfc5613>>.
- [RFC7142] Shand, M. and L. Ginsberg, "Reclassification of [RFC 1142](#) to Historic", [RFC 7142](#), DOI 10.17487/RFC7142, February 2014, <<https://www.rfc-editor.org/info/rfc7142>>.
- [RFC8099] Chen, H., Li, R., Retana, A., Yang, Y., and Z. Liu, "OSPF Topology-Transparent Zone", [RFC 8099](#), DOI 10.17487/RFC8099, February 2017, <<https://www.rfc-editor.org/info/rfc8099>>.

10.2. Informative References

- [Clos] Clos, C., "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal Vol. 32(2), DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953, <<http://dx.doi.org/10.1002/j.1538-7305.1953.tb01433.x>>.
- [RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", [RFC 5307](#), DOI 10.17487/RFC5307, October 2008, <<https://www.rfc-editor.org/info/rfc5307>>.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA
USA

Email: huaimo.chen@futurewei.com

Alvaro Retana
Futurewei
Raleigh, NC
USA

Email: alvaro.retana@futurewei.com

Richard Li
Futurewei
2330 Central expressway
Santa Clara, CA
USA

Email: richard.li@futurewei.com

Anil Kumar S N
RtBrick
Bangalore
India

Email: anil.ietf@gmail.com

Ning So
Plano, TX 75082
USA

Email: ningso01@gmail.com

Vic Liu
USA

Email: liu.cmri@gmail.com

Mehmet Toy
Verizon
USA

Email: mehmet.toy@verizon.com

Lei Liu
Fijitsu
USA

Email: liulei.kddi@gmail.com