Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: March 18, 2021

H. Chen
R. Li
Futurewei
Y. Yang
IBM
A. Kumar S N
RtBrick
Y. Fan
Casa Systems
N. So

V. Liu

M. Toy
Verizon
L. Liu
Fujitsu
K. Makhijani
Futurewei
September 14, 2020

IS-IS Topology-Transparent Zone
draft-ietf-lsr-isis-ttz-00.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 virtual node or zone edges mesh. 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

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 March 18, 2021.

Copyright Notice

Table of Contents

1. Introduction

[ISO10589] describes two levels of areas, which are level 1 and level 2 areas in IS-IS. 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.

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 virtual node or zone edges' mesh 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 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.

LS: A Link Sate, which is short for 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 entity, which is a single virtual node or zone edges' mesh, SHOULD be smooth with minimum service interruption.

- o De-abstracting (or say rolling back) a virtual 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 virtual node or the zone edges' full mesh.

When a zone is abstracted as a single virtual 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 mesh, there is a full mesh 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. It is abstracted as a single virtual node or its edges' full mesh. TTZ and zone will be used exchangably below.

4.1. Zone as a Single Node

After a zone is abstracted as a single virtual node having a virtual 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 virtual 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.

```
                 TTZ 600
                   \
                    \ ^~^~^~^~^~^~^~^~^~^~^~^~
                     (                        )
    ===[R15]========(==[R61]------------[R63]==)======[R29]===
        ||         (   |    \          /    |   )       ||
        ||         (   |     \        /     |   )       ||
        ||         (   |      \      /      |   )       ||
        ||         (   |    ___\    /       |   )       ||
        ||         (   |   /   [R71]        |   )       ||
        ||         (   | [R73] /    \       |   )       ||
        ||         (   |      /      \      |   )       ||
        ||         (   |     /        \     |   )       ||
        ||         (   |    /          \    |   )       ||
    ===[R17]========(==[R65]------------[R67]==)======[R31]===
         \\          (//                    \\)       //
          ||         //v~v~v~v~v~v~v~v~v~v~v~\\      ||
          ||        //                        \\     ||
          ||       //                          \\    ||
           \\     //                            \\  //
       ======[R23]==============================[R25]=====
             //                                    \\
            //                                      \\
```

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 Rz, which has normal connections to R15, R29, R17 and R23, R25 and R31.

```
                 TTZ 600
                   \
                    \ ^~^~^~^~^~^~^~^~^~^~^~^~
                     (                        )
    ===[R15]========(                          )======[R29]===
        ||         (                            )       ||
        ||         (                            )       ||
        ||         (                            )       ||
        ||         (                            )       ||
        ||         (             Rz             )       ||
        ||         (                            )       ||
        ||         (                            )       ||
        ||         (                            )       ||
        ||         (                            )       ||
    ===[R17]========(                          )======[R31]===
         \\          (                        )       //
          ||         //v~v~v~v~v~v~v~v~v~v~v~\\      ||
          ||        //                        \\     ||
          ||       //                          \\    ||
           \\     //                            \\  //
       ======[R23]==============================[R25]=====
             //                                     \\
            //                                       \\
```

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 an LS (i.e., an LSP in IS-IS) for the zone as a single virtual node and sends it to its neighbors.

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

A zone edge node, acting as a single virtual 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 virtual 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 virtual 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):

At first, the edge acting as virtual node creates a new adjacency between the virtual node for the zone and the zone external node in a normal way. It sends Hellos and other packets containing the virtual node ID as Source ID to the zone external node. The zone external node establishes the adjacency with the virtual in the normal way.

And then, the edge terminates the existing adjacency between the edge and the external node after the zone has been transferred to the virtual node. It stops sending Hellos for the adjacency to the zone external node. Without receiving Hellos from the edge node for a given time such as hold-timer interval, the zone external node removes the adjacency to the edge node. Even though this adjacency terminates, the edge node keeps the link to the external node in its LS.

In another option, 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 virtual 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 Virtual Node)    Hello(T=1, Virtual ID)
                     -----------------------> OK for Transfer
                                              Adjacency
                      Hello(T=1, Virtual ID)
   Remote Ready for <-----------------------
   Transfer
                    Hello(Source=Virtual ID)
   Start Transfer   ------------------------> Transfer to New
                                              Adjacency
                     Hello
   Transfer to New  <------------------------
   Adjacency                 . . .
```

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

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

Step 3: The edge sends the zone neighbor a Hello containing the virtual node ID as Source ID after receiving the Hello with T=1 and virtual 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 virtual 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 virtual 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 virtual node ID through either removing the existing adjacency and adding a new adjacency with the virtual node ID or just changing the existing adjacency ID from the edge node ID to the virtual node ID,

o Removing the link to the zone edge node from its LS and adding a new link to the virtual node (or just changing the link to the edge node to the link to the virtual 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 virtual node ID as Source ID.

4.1.5. Computation of Routes

After a zone edge migrates to zone as a virtual node, it computes the routes (i.e., shortest paths to the destinations) in the zone using the zone topology (i.e., the topology of the zone without the virtual node).

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

After a zone internal node migrates to zone as a virtual node, it computes the routes using the zone topology, the topology outside of the zone without the virtual 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.

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 virtual node ID; in case of roll back, the TLV contains the edge node ID.

o Zone TLV: containing a zone ID, a flags field and optional sub-TLVs.

4.1.6.1. Adjacent Node ID TLV

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 (TBD1)  |  Length (8)   |     Virtual/Edge Node ID      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Virtual/Edge Node ID (Continue)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Flags          |N|T|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type (1 byte): To be assigned by IANA.

Length (1 byte): Its value is 8.

Virtual/Edge Node ID (6 bytes): An adjacent node ID, to which an adjacency is transferred or rolled back.

Flags field (16 bits): two new flag bits are defined as follows:

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 virtual node ID (or say abstract node ID).

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.

4.1.6.2. Zone TLV

The format of IS-IS Zone TLV is illustrated below. It may be added into an LSP or a Hello PDU for a zone node. When a node in a zone receives a CLI command triggering zone information distribution for migration, it updates its LSP by adding an 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.

```
 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    |            Zone ID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Zone ID (Continue)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Flags             |E|Z|S|  OP |            Sub TLVs           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: IS-IS Zone TLV

Type (1 byte): To be assigned by IANA.

Length (1 byte): Its value is variable.

Zone ID (6 bytes): It is the identifier (ID) of a zone.

Flags field (16 bits): Three flag bits E, Z and S, and OP of 3 bits are defined.

E = 1: Indicating a node is a zone edge node
Z = 1: Indicating a node has migrated to Zone as a virtual entity
S = 1: Indicating the virtual entity is a Single virtual node

When a zone node originates an LS containing a zone TLV, it MUST set flag E to 1 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 a virtual entity and to 0 before it migrates zone to the virtual entity or after it rolls back from zone as a virtual entity. When the entity abstracted from a zone is a Single virtual node, flag S MUST be set to 1.

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

The value of OP indicates one of the four operations above. When any of the other values is received, it is ignored.

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

Two new sub-TLVs are defined, which may be added into an 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 IS neighbors in the zone connected to a zone edge router. It has the format 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    |DefaultMetric(i| DelayMetric(i)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|ExpenseMetric(i| ErrorMetric(i)|        Neighbor ID(i)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Zone ISN Sub TLV

A Zone ISN Sub TLV has 1 byte of Type, 1 byte of Length of n*(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 ES neighbors in the zone connected to a zone edge node. It has the format 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    |Default Metric |  DelayMetric  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Expense Metric |  Error Metric |        Neighbor ID(i)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Zone ESN Sub TLV

4.2. Zone as Edges Full Mesh

OSPF Topology-Transparent Zone [RFC8099] describes the zone as edges' full mesh and the extensions to OSPF for supporting zone as edges' full mesh. 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. Updating LSPs for Zone

A zone internal node adds an IS-IS Zone TLV into its LSP after it receives an LSP containing an 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 an 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 an IS-IS Zone TLV into its LSP after it receives an LSP containing an 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 an 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 virtual node or edges' full mesh, 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 virtual node. The seamless migration between a zone and its edges' full mesh 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 Single Virtual Node is triggered, the zone is abstracted as a single virtual node in two steps:

Step 1: Every zone edge node works together with each of its zone neighbor nodes to create a new adjacency between the virtual node and the neighbor node in the way described in Section 4.1.4 for Adjacency Establishment and Termination procedure for case 2. After creating the adjacency, each of the zone neighbor nodes update its LS by adding the adjacency/link into its LS.

Step 2: The zone leader originates an LS for the virtual node after receiving the updated LSes originated by all the zone neighbor nodes, where the updated LSes contain all the zone neighbors.

Step 3: After receiving the LS for the virtual node, every zone edge does not send any LS inside the zone to any zone neighbors. It advertises its LS without any links inside the zone to the nodes outside of the zone and terminates its adjacency to each of its zone neighbors in the way described in Section 4.1.4 for Adjacency Establishment and Termination procedure for case 2.

5.2. Roll Back from Zone as a Single Node

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

Step 1: Every zone edge creates an adjacency to each of its zone neighbors in a normal way.

Step 2: After all the adjacencies between the zone edges and the zone neighbors are created, the zone leader updates the LS for the virtual node by changing every link metric to the maximum metric in the LS.

Step 3: Every zone edge sends its LS with the links inside the zone and all the LSes inside the zone to its zone neighbors. Every zone edge acting as the virtual node terminates the adjacency between the virtual node and each of its zone neighbors through stopping Hellos to the neighbors.

In another option, rolling back is done as follows:

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 virtual 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 virtual node. Every zone edge sends Hello and other packets to its zone neighbors, where the packets 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 virtual 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 Single 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 virtual 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 virtual node ID to the edge node ID,

- o Removing the link to the virtual node from its LS and adding a new link to the edge node (or just changing the link to the virtual 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 Mesh in OSPF. They can be used for Zone as Edges Full Mesh in IS-IS. They can also be used for Zone as a Single Virtual Node in IS-IS.

7. Security Considerations

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

8. IANA Considerations

Under the registry name "IS-IS TLV Codepoints", IANA is requested to assign new registry types for Adjacent Node ID, Zone ID and Zone Options as follows:

TLV Type	TLV Name	reference
26(suggested)	Adjacent Node ID	This document
27(suggested)	Zone	This document

9. Contributors

Alvaro Retana
Futurewei
Raleigh, NC
USA

Email: alvaro.retana@futurewei.com

10. Acknowledgement

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

11. References

11.1. Normative References

[I-D.ietf-lsr-dynamic-flooding]
Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., Dontula, S., and G. Mishra, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-07 (work in progress), June 2020.

[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>.

[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>.

[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>.

11.2. 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

Richard Li
Futurewei
2330 Central expressway
Santa Clara, CA
USA

Email: richard.li@futurewei.com

Yi Yang
IBM
Cary, NC
United States of America

Email: yyietf@gmail.com

Anil Kumar S N
RtBrick
Bangalore
India

Email: anil.ietf@gmail.com

Yanhe Fan
Casa Systems
USA

Email: yfan@casa-systems.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
Fujitsu
USA

Email: liulei.kddi@gmail.com

Kiran Makhijani
Futurewei
USA

Email: kiranm@futurewei.com