Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 24, 2019

H. Chen
D. Cheng
Huawei Technologies
M. Toy
Verizon
Y. Yang
IBM

September 20, 2018

LS Flooding Reduction
draft-cc-ospf-flooding-reduction-04

Abstract

This document proposes an approach to flood link states on a topology that is a subgraph of the complete topology per underline physical network, so that the amount of flooding traffic in the network is greatly reduced, and it would reduce convergence time with a more stable and optimized routing environment. The approach can be applied to any network topology in a single area.

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].

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 March 24, 2019.

Table of Contents

1. Introduction

For some networks such as dense Data Center (DC) networks, the existing Link State (LS) flooding mechanism is not efficient and may have some issues. The extra LS flooding consumes network bandwidth. Processing the extra LS flooding, including receiving, buffering and decoding the extra LSs, wastes memory space and processor time. This may cause scalability issues and affect the network convergence negatively.

This document proposes an approach to minimize the amount of flooding traffic in the network. Thus the workload for processing the extra LS flooding is decreased significantly. This would improve the scalability, speed up the network convergence, stable and optimize the routing environment.

This approach is also flexible. It has multiple modes for computation of flooding topology. Users can select a mode they prefer, and smoothly switch from one mode to another. The approach is applicable to any network topology in a single area. It is backward compatible.

2. Terminology

Flooding Topology:
A sub-graph or sub-network of a given (physical) network topology that has the same reachability to every node as the given network topology, through which link states are flooded.

critical link or interface on a flooding topology:
A only link or interface among some nodes on the flooding topology. When this link or interface goes down, the flooding topology will be split.

critical node on a flooding topology:

A only node connecting some nodes on the flooding topology. When this node goes down, the flooding topology will be split.

backup path:
A path or a sequence of links, when a critical link or node goes down, providing a connection to connect two parts of a split flooding topology. When a critical node goes down, the flooding topology may be split into more than two parts. In this case, two or more backup paths are needed to connect all the split parts into one.

Remaining Flooding Topology:
A topology from a flooding topology by removing the failed links and nodes from the flooding topology.

LSA:
A Link State Advertisement in OSPF.

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

LS:
A Link Sate, which is an LSA or LSP.

3. Conventions Used in This Document

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].

4. Problem Statement

OSPF and IS-IS deploy a so-called reliable flooding mechanism, where a node must transmit a received or self-originated LS to all its interfaces (except the interface where an LS is received). While this mechanism assures each LS being distributed to every node in an area or domain, the side-effect is that the mechanism often causes redundant LS, which in turn forces nodes to process identical LS more than once. This results in the waste of link bandwidth and nodes' computing resources, and the delay of topology convergence.

This becomes more serious in networks with large number of nodes and links, and in particular, higher degree of interconnection (e.g., meshed topology, spine-leaf topology, etc.). In some environments such as in data centers, the drawback of the existing flooding mechanism has already caused operational issues, including repeated and waves of flooding storms, chock of computing resources, slow

convergence, oscillating topology changes, instability of routing environment.

One example is as shown in Figure 1, where Node 1, Node 2 and Node 3 are interconnected in a mesh. When Node 1 receives a new or updated LS on its interface I11, it by default would forward the LS to its interface Il2 and I13 towards Node 2 and Node 3, respectively, after processing. Node 2 and Node 3 upon reception of the LS and after processing, would potentially flood the same LS over their respective interface I23 and I32 toward each other, which is obviously not necessary and at the cost of link bandwidth as well as both nodes' computing resource.

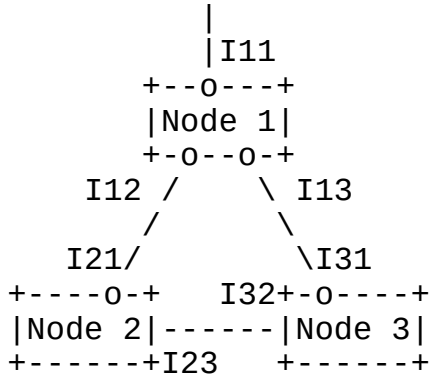


Figure 1

5. Flooding Topology

For a given network topology, a flooding topology is a sub-graph or sub-network of the given network topology that has the same reachability to every node as the given network topology. Thus all the nodes in the given network topology MUST be in the flooding topology. All the nodes MUST be inter-connected directly or indirectly. As a result, LS flooding will in most cases occur only on the flooding topology, that includes all nodes but a subset of links. Note even though the flooding topology is a sub-graph of the original topology, any single LS MUST still be disseminated in the entire network.

5.1. Construct Flooding Topology

Many different flooding topologies can be constructed for a given network topology. A chain connecting all the nodes in the given network topology is a flooding topology. A circle connecting all the nodes is another flooding topology. A tree connecting all the nodes is a flooding topology. In addition, the tree plus the connections

between some leaves of the tree and branch nodes of the tree is a flooding topology.

The following parameters need to be considered for constructing a flooding topology:

- Number of links: The number of links on the flooding topology is a key factor for reducing the amount of LS flooding. In general, the smaller the number of links, the less the amount of LS flooding.

- Diameter: The shortest distance between the two most distant nodes on the flooding topology is a key factor for reducing the network convergence time. The smaller the diameter, the less the convergence time.

- Redundancy: The redundancy of the flooding topology means a tolerance to the failures of some links and nodes on the flooding topology. If the flooding topology is split by some failures, it is not tolerant to these failures. In general, the larger the number of links on the flooding topology is, the more tolerant the flooding topology to failures.

There are many different ways to construct a flooding topology for a given network topology. A few of them are listed below:

- Central Mode: One node in the network builds a flooding topology and floods the flooding topology to all the other nodes in the network (This seems not good. Flooding the flooding topology may increase the flooding. The amount of traffic for flooding the flooding topology should be minimized.);

- Distributed Mode: Each node in the network automatically calculates a flooding topology by using the same algorithm (No flooding for flooding topology);

- Static Mode: Links on the flooding topology are configured statically.

Note that the flooding topology constructed by a node is dynamic in nature, that means when the base topology (the entire topology graph) changes, the flooding topology (the sub-graph) MUST be re-computed/ re-constructed to ensure that any node that is reachable on the base topology MUST also be reachable on the flooding topology.

For reference purpose, some algorithms that allow nodes to automatically compute flooding topology are elaborated in Appendix A.

However, this document does not attempt to standardize how a flooding topology is established.

5.2. Backup for Flooding Topology Split

It is hard to construct a flooding topology that reduces the amount of LS flooding greatly and is tolerant to multiple failures. To get around this, we can compute and use backup paths for a critical link and node on the flooding topology. Using backup paths may also speed up convergence when the link and node fail.

When a critical link on the flooding topology fails, the flooding topology without the critical link (i.e., the remaining flooding topology) is split into two parts. A backup path for the critical link connects the two parts into one. Through the backup path and the remaining flooding topology, an LS can be flooded to every node in the network. The combination of the backup path and the flooding topology is tolerant to the failure of the critical link.

When a critical node on the flooding topology goes down, the flooding topology without the critical node and the links attached to the node (i.e., the remaining flooding topology) is split into two or more parts. One or more backup paths for the critical node connects the split parts into one. Through the backup paths and the remaining flooding topology, an LS can be flooded to every live node in the network. The combination of the backup paths and the flooding topology is tolerant to the failure of the critical node.

In addition to the backup paths for a critical link and node, backup paths for every non critical link and node on the flooding topology can be computed. When the failures of multiple links and nodes on the flooding topology happen, through the remaining flooding topology and the backup paths for these links and nodes, an LS can be flooded to every live node in the network. The combination of the backup paths and the flooding topology is tolerant to the failures of these links and nodes. If there are other failures that break the backup paths, an LS can be flooded to every live node by the traditional flooding procedure.

In a centralized mode, the leader computes the backup paths and floods them to all the other nodes. In a distributed mode, every node computes the backup paths.

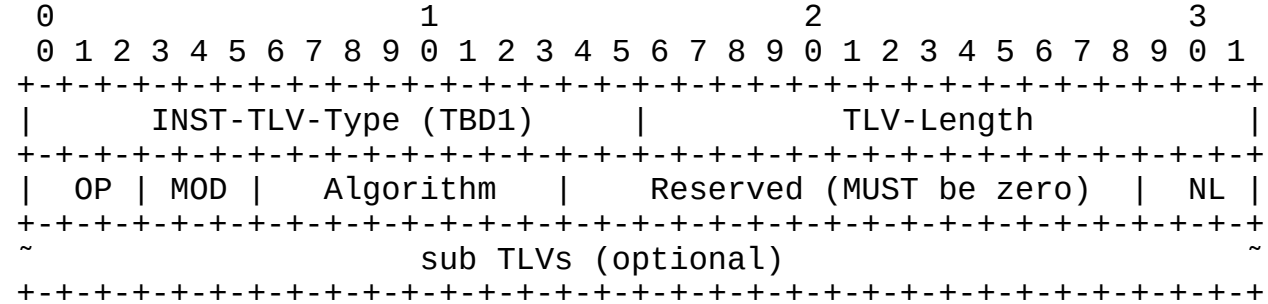
6. Extensions to OSPF

The extensions to OSPF comprises two parts: one part is for operations on flooding reduction, the other is specially for centralized mode flooding reduction.

6.1. Extensions for Operations

A new TLV is defined in OSPF RI LSA [RFC7770]. It contains instructions about flooding reduction, which is called Flooding Reduction Instruction TLV or Instruction TLV for short. This TLV is originated from only one node at any time.

The format of a Flooding Reduction Instruction TLV is as follows.



Flooding Reduction Instruction TLV

A OP field of three bits is defined in the TLV. It may have a value of the followings.

- 0x001 (R): Perform flooding Reduction, which instructs the nodes in a network to perform flooding reduction.

- 0x010 (N): Roll back to Normal flooding, which instructs the nodes in a network to roll back to perform normal flooding.

When any of the other values is received, it is ignored.

A MOD field of three bits is defined in the TLV and may have a value of the followings.

- 0x001 (C): Central Mode, which instructs 1) the nodes in a network to select leaders (primary/designated leader, secondary/backup leader, and so on); 2) the leaders in a network to compute a flooding topology and the primary leader to flood the flooding topology to all the other nodes in the network; 3) every node in the network to receive and use the flooding topology originated by the primary leader.

- 0x010 (D): Distributed Mode, which instructs every node in a network to compute and use its own flooding topology.

- 0x011 (S): Static Mode, which instructs every node in a network to use the flooding topology statically configured on the node.

When any of the other values is received, it is ignored.

An Algorithm field of eight bits is defined in the TLV to instruct the leader node in central mode or every node in distributed mode to use the algorithm indicated in this field for computing a flooding topology.

A NL field of three bits is defined in the TLV, which indicates the number of leaders to be selected when Central Mode is used. NL set to 2 means two leaders (a designated/primary leader and a backup/ secondary leader) to be selected for an area, and NL set to 3 means three leaders to be selected. When Central Mode is not used, The NL field is not valid.

Some optional sub TLVs may be defined in the future, but none is defined now.

6.2. Extensions for Centralized Mode

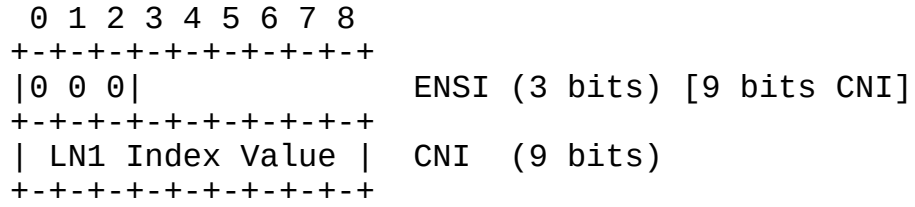
6.2.1. Message for Flooding Topology

A flooding topology can be represented by the links in the flooding topology. For the links between a local node and a number of its adjacent (or remote) nodes, we can encode the local node in a way, and encode its adjacent nodes in the same way or another way. After all the links in the flooding topology are encoded, the encoded links can be flooded to every node in the network. After receiving the encoded links, every node decodes the links and creates and/or updates the flooding topology.

For every node in an area, we may use an index to represent it. Every node in an area may order the nodes in a rule, which generates the same sequence of the nodes on every node in the area. The sequence of nodes have the index 0, 1, 2, and so on respectively. For example, every node orders the nodes by their router IDs in ascending order.

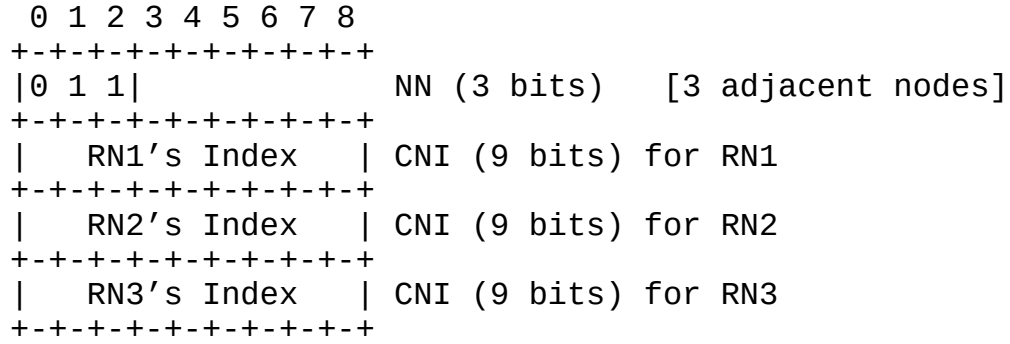
6.2.1.1. Links Encoding

A local node can be encoded in two parts: encoded node index size indication (ENSI) and compact node index (CNI). ENSI value plus a number (e.g., 9) gives the size of compact node index. For example, ENSI = 0 indicates that the size of CNIs is 9 bits. In the figure below, Local node LN1 is encoded as ENSI=0 using 3 bits and CNI=LN1’s Index using 9 bits. LN1 is encoded in 12 bits in total.



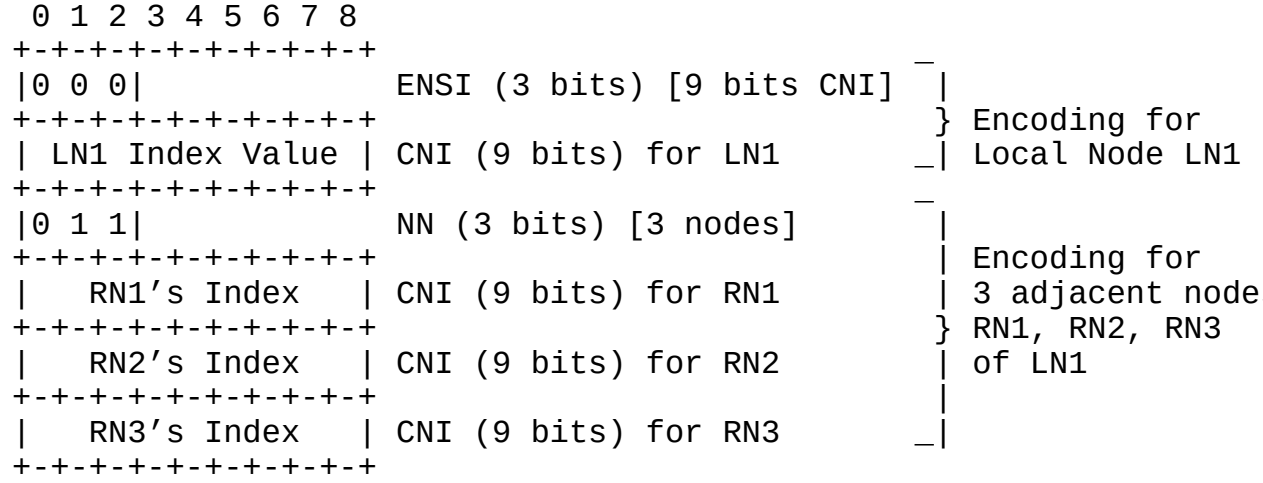
An Example of Local Node Encoding

The adjacent nodes can be encoded in two parts: Number of Nodes (NN) and compact node indexes (CNIs). The size of CNIs is the same as the local node. For example, three adjacent nodes RN1, RN2 and RN3 are encoded below in 30 bits (i.e., 3.75 bytes).



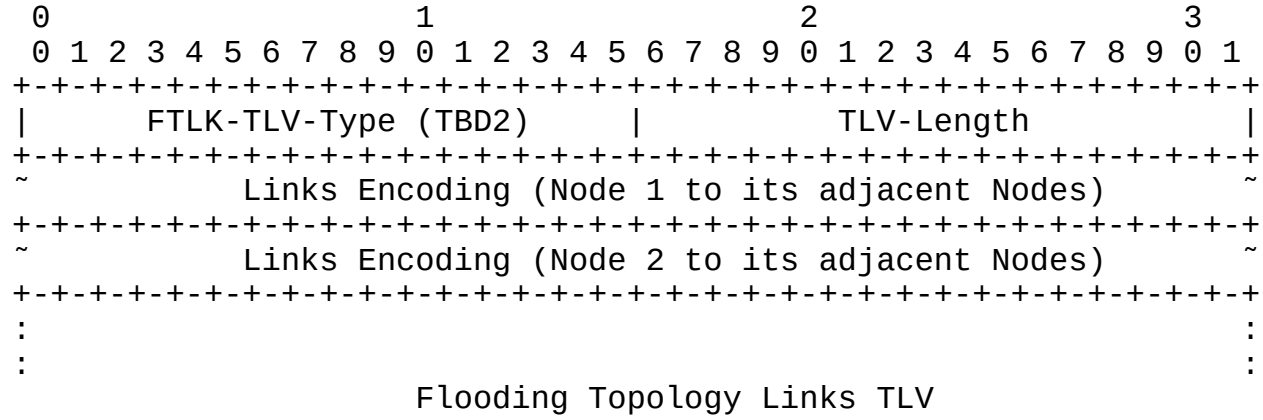
An Example of Adjacent Nodes Encoding

The links between a local node and a number of its adjacent (or remote) nodes can be encoded as the local node followed by the adjacent nodes. For example, three links between local node LN1 and its three adjacent nodes RN1, RN2 and RN3 are encoded below in 42 bits (i.e., 5.25 bytes).



An Example of Links Encoding

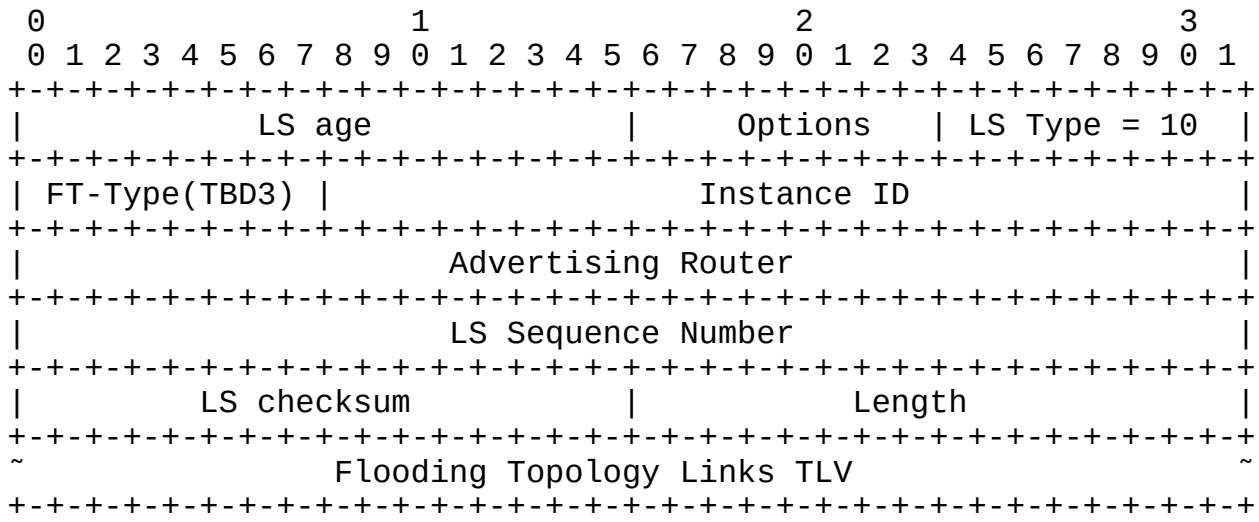
For a flooding topology computed by a leader of an area, it may be represented by all the links on the flooding topology. A Type-Length-Value (TLV) of the following format for the links encodings can be included in an LSA to represent the flooding topology (FT) and flood the FT to every node in the area.



Flooding Topology Links TLV

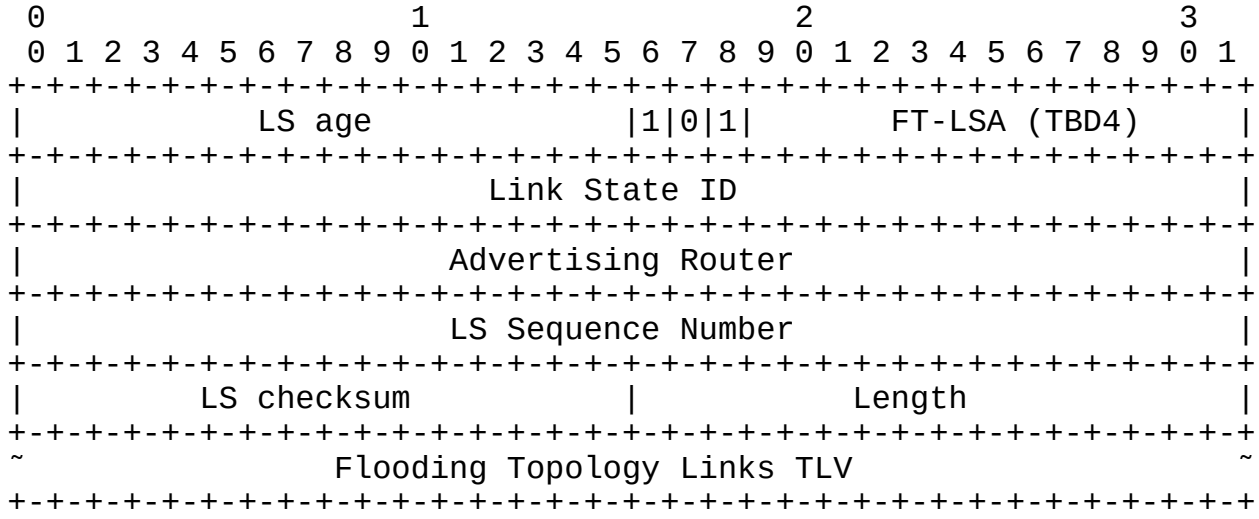
Note that a link between a local node LN and its adjacent node RN can be encoded once and as a bi-directional link. That is that if it is encoded in a Links Encoding from LN to RN, then the link from RN to LN is implied or assumed.

For OSPFv2, an Opaque LSA of a new opaque type (TBD3) containing a Flooding Topology Links TLV is used to flood the flooding topology from the leader of an area to all the other nodes in the area.



OSPFv2: Flooding Topology Opaque LSA

For OSPFv3, an area scope LSA of a new LSA function code (TBD4) containing a Flooding Topology Links TLV is used to flood the flooding topology from the leader of an area to all the other nodes in the area.



OSPFv3: Flooding Topology LSA

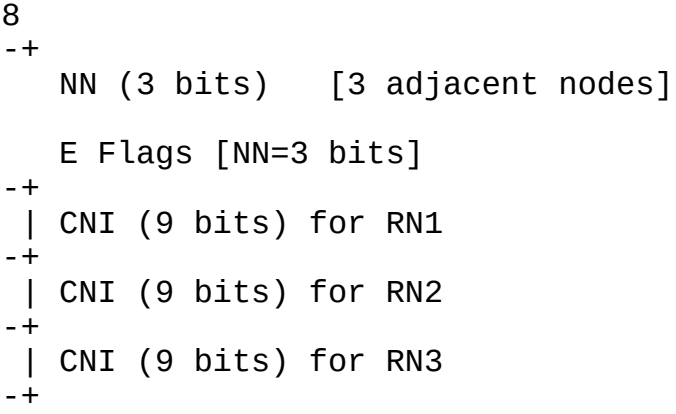
The U-bit is set to 1, and the scope is set to 01 for area-scoping.

6.2.1.2. Block Encoding

Block encoding uses a single structure to encode a block (or part) of topology, which can be a block of links in a flooding topology. It can also be all the links in the flooding topology. It starts with a local node LN and its adjacent (or remote) nodes RNi (i = 1, 2, ..., n), and can be considered as an extension to the links encoding.

The encoding of links between a local node and its adjacent nodes described in Section 6.2.1.1 is extended to include the links attached to the adjacent nodes.

The encoding for the adjacent nodes is extended to include Extending Flags (E Flags for short) between the NN (Number of Nodes) field and the CNIs (Compact Node Indexes) for the adjacent nodes. The length of the E Flags field is NN bits. The following is an example encoding of the adjacent nodes with E Flags of 3 bits, which is the value of the NN (the number of adjacent nodes).



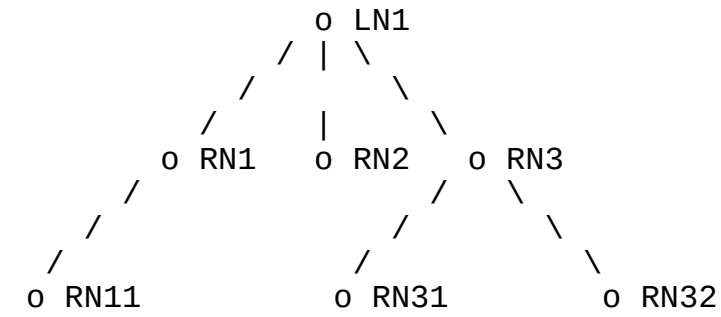
An Example of Adjacent Nodes with E Flags Encoding

There is a bit flag (called E flag) in the E Flags field for each adjacent node. The first bit (i.e., the most significant bit) in the E Flags field is for the first adjacent node (e.g., RN1), the second bit is for the second adjacent node (e.g., RN2), and so on. The E flag for an adjacent node RNi set to one indicates that the links attached to the adjacent node RNi are included below. The E flag for an adjacent node RNi set to zero means that no links attached to the adjacent node RNi are included below.

The links attached to the adjacent node RNi are represented by the RNi as a local node and the adjacent nodes of RNi. The encoding for the adjacent nodes of RNi is the same as that for the adjacent nodes

of a local node. It consists of an NN field of 3 bits, E Flags field of NN bits, and CNIs for the adjacent nodes of RNi.

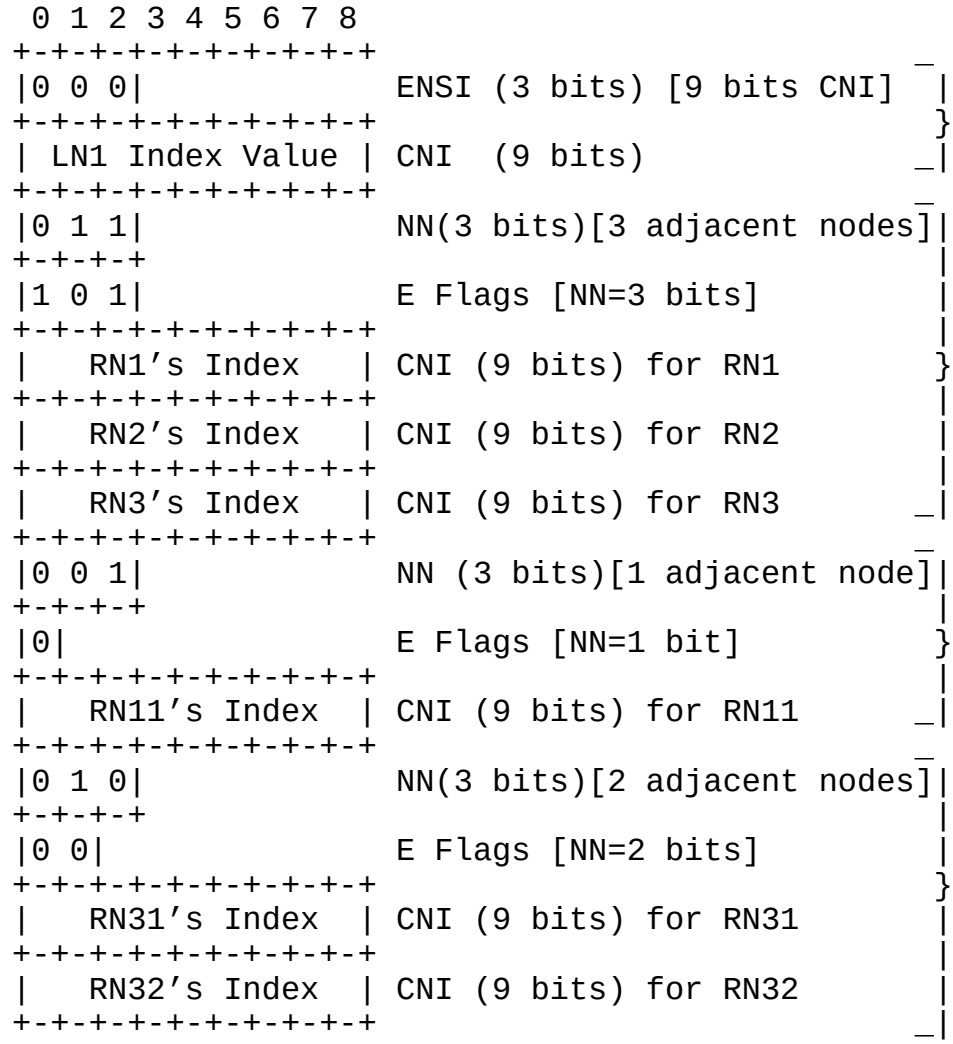
The following is an example of a block encoding for a block (or part) of flooding topology below.



An Example Block of Flooding Topology

It represents 6 links: 3 links between local node LN1 and its 3 adjacent nodes RN1, RN2 and RN3; 1 link between RN1 as a local node and its 1 adjacent node RN11; and 2 links between RN3 as a local node and its 2 adjacent nodes RN31 and RN32.

It starts with the encoding of the links between local node LN1 and 3 adjacent nodes RN1, RN2 and RN3 of the local node LN1. The encoding for the local node LN1 is the same as that for a local node described in Section 6.2.1.1. The encoding for 3 adjacent nodes RN1, RN2 and RN3 of local node LN1 comprises an NN field of 3 bits with value of 3, E Flags field of NN = 3 bits, and the indexes of adjacent nodes RN1, RN2 and RN3.



An Example of Block Encoding

The first E flag in the encoding for adjacent nodes RN1, RN2 and RN3 is set to one, which indicates that the links between the first adjacent node RN1 as a local node and its adjacent nodes are included below. In this example, 1 link between RN1 and its adjacent node RN11 is represented by the encoding for the adjacent node RN11 of RN1 as a local node. The encoding for 1 adjacent node RN11 consists of an NN field of 3 bits with value of 1, E Flags field of NN = 1 bits, and the index of adjacent node RN11. The size of the index of RN11 is the same as that of local node LN1 indicated by the ENSI in the encoding for local node LN1.

The second E flag in the encoding for adjacent nodes RN1, RN2 and RN3 is set to zero, which indicates that no links between the second

adjacent node RN2 as a local node and its adjacent nodes are included below.

The third E flag in the encoding for adjacent nodes RN1, RN2 and RN3 is set to one, which indicates that the links between the third adjacent node RN3 as a local node and its adjacent nodes are included below. In this example, 2 links between RN3 and its 2 adjacent nodes RN31 and RN32 are represented by the encoding for the adjacent nodes RN31 and RN32 of RN3 as a local node. The encoding for 2 adjacent nodes RN31 and RN32 consists of an NN field of 3 bits with value of 2, E Flags field of NN = 2 bits, and the indexes of adjacent nodes RN31 and RN32. The size of the index of RN31 and RN32 is the same as that of local node LN1 indicated by the ENSI in the encoding for local node LN1.

The block encoding may be used in the place of the links encoding in Section 6.2.1.1 for more efficiency. That is that it may be used in a Flooding Topology Links TLV. Alternatively, a new TLV, which is similar to the Flooding Topology Links TLV, may be defined to contain a number of block encodings.

6.2.2. Encodings for Backup Paths

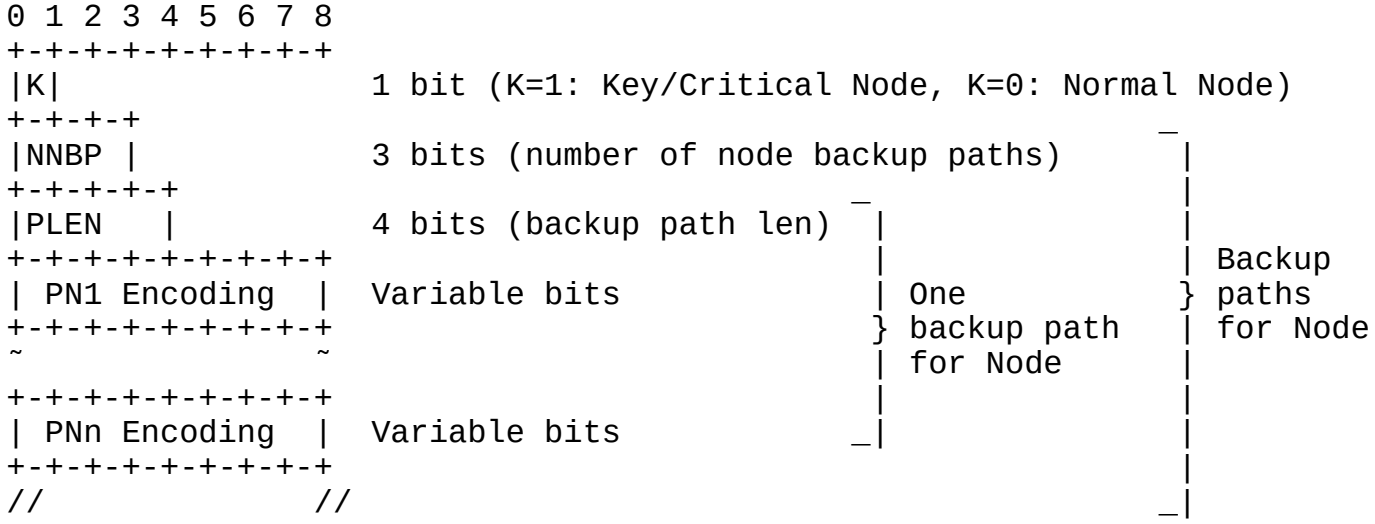
When the leader of an area computes a flooding topology, it may compute a backup path or multiple backup paths for a critical link on the flooding topology. When the critical link fails, a link state can be distributed to every node in the area through one backup path and other links on the flooding topology. In addition, it may compute a backup path or multiple backup paths for a node. When the node fails, a link state can be distributed to the other nodes in the area through the backup paths and the links on the flooding topology.

This section describes two encodings for backup paths: separated encoding and integrated one. In the former, backup paths are encoded in a new message, where the message for the flooding topology described in the previous section is required; In the latter, backup paths are integrated into the flooding topology links encoding, where one message contains the flooding topology and the backup paths.

6.2.2.1. Message for Backup Paths

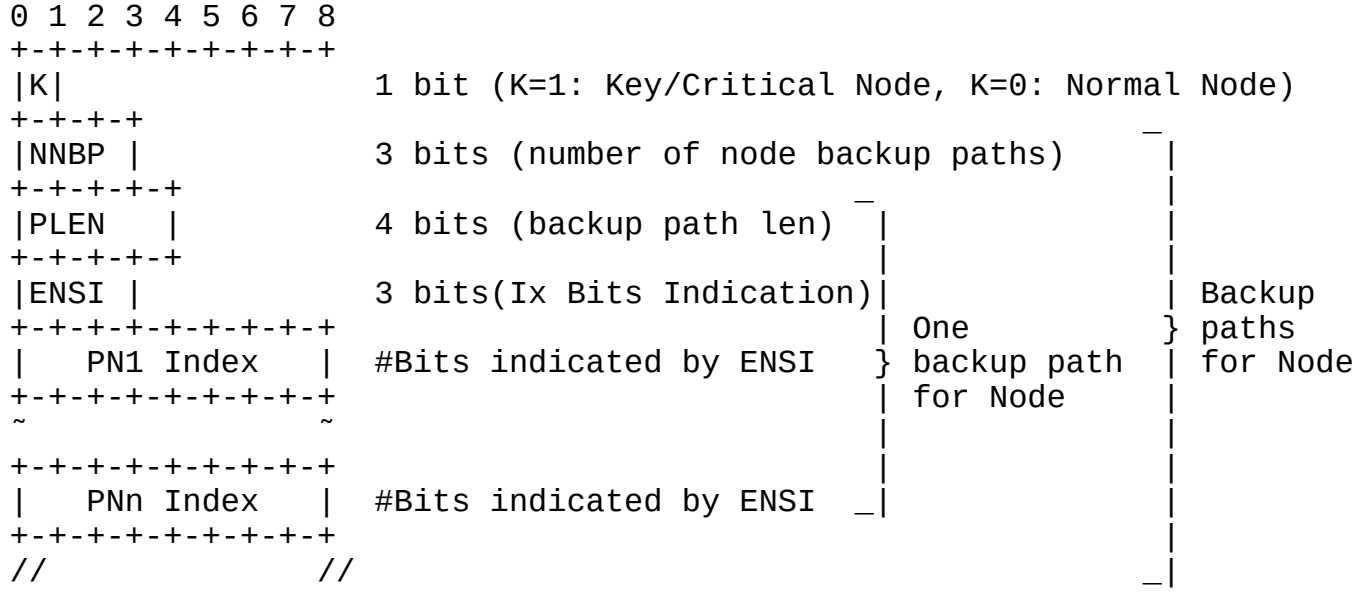
Backup paths for a node (such as Node1) may be represented by the node index encoding and node backup paths encoding. The former is similar to local node index encoding. The latter has the following format. It comprises a K flag (Key/Critical node flag) of 1 bit, a 3 bits NNBP field (number of node backup paths), and each of the backup paths encoding, which consists of the path length PLEN of 4 bits indicating the length of the path (i.e., the number of nodes), and

the encoding of the sequence of nodes along the path such as encodings for nodes PN1, ..., PNn. The encoding of every node may use the encoding of a local node, which comprises encoded node index size indication (ENSI) and compact node index (CNI).



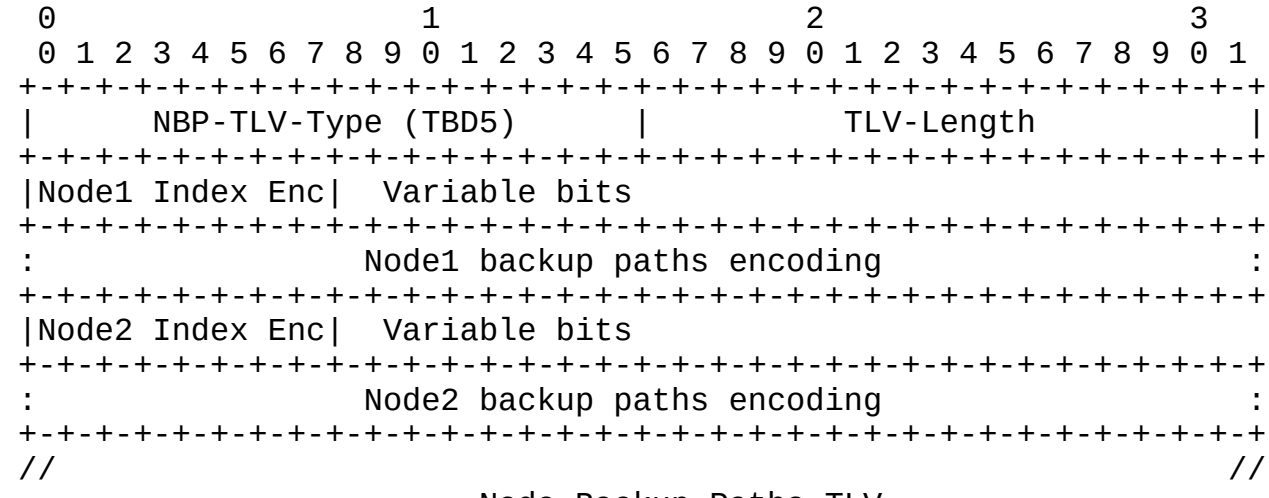
An Example of Node Backup Paths Encoding

Another encoding of the sequence of nodes along the path uses one encoded node index size indication (ENSI) for all the nodes in the path. Thus we have the following Node Backup Paths Encoding.



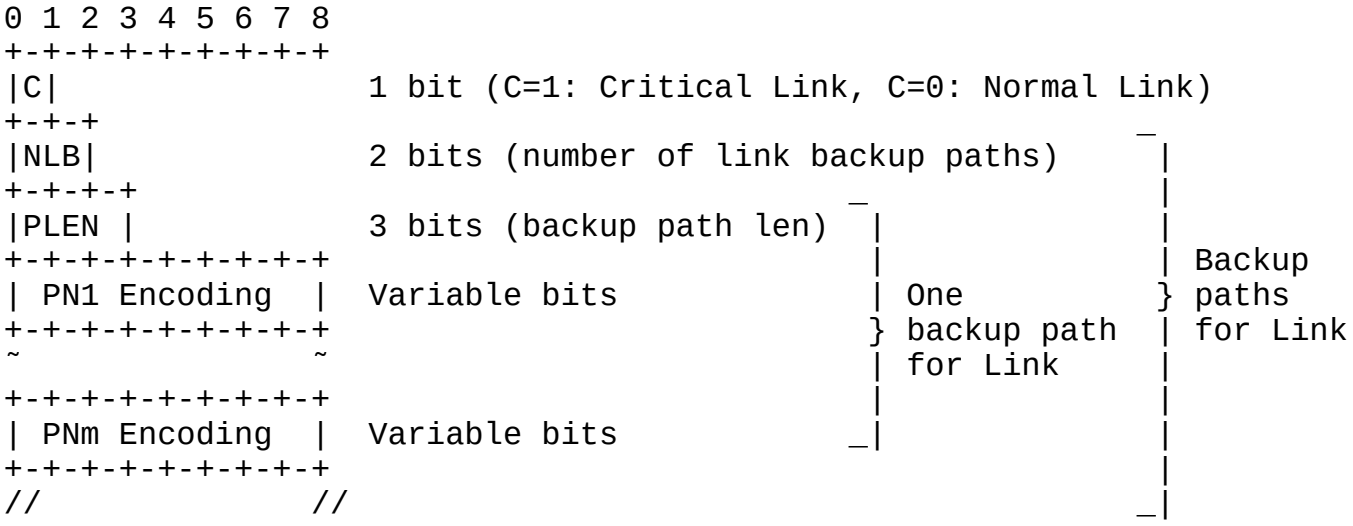
Another Example of Node Backup Paths Encoding

A new TLV called Node Backup Paths TLV is defined below. It may include multiple nodes and their backup paths. Each node is represented by its index encoding, which is followed by its node backup paths encoding.



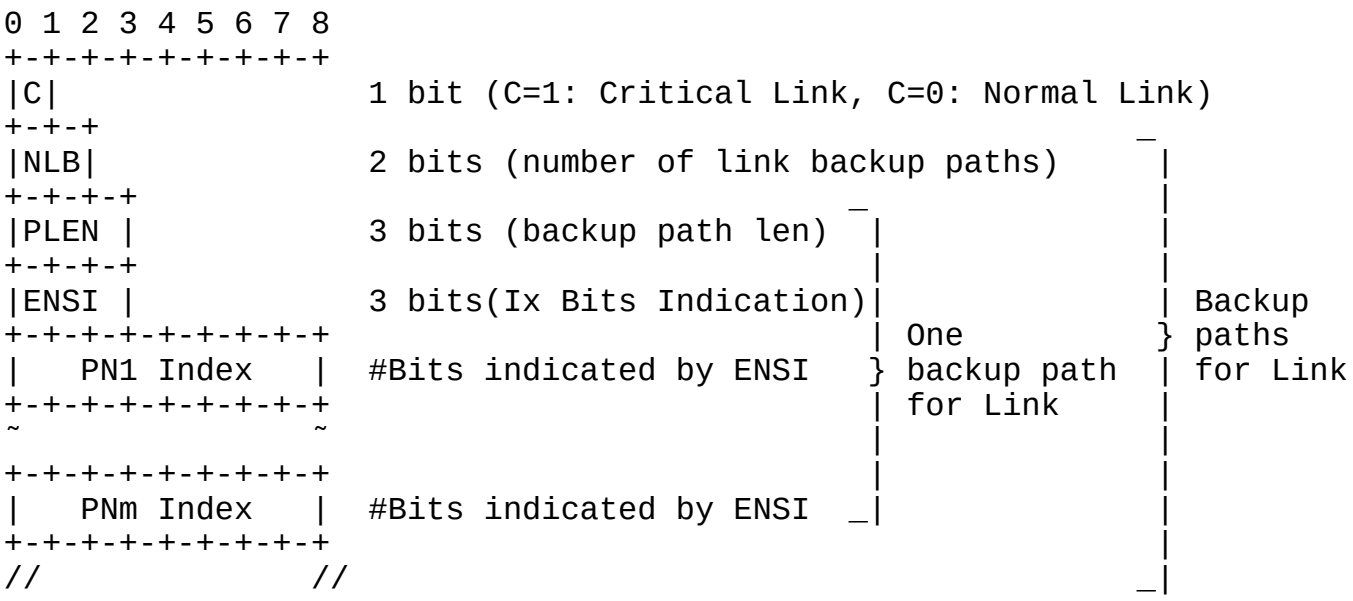
Node Backup Paths TLV

The encoding for backup paths for a link (such as Link1) on the flooding topology consists of the link encoding such as Link1 Index Encoding and the link backup paths encoding. The former is similar to local node encoding. It contains encoded link index size indication (ELSI) and compact link index (CLI). The latter has the following format. It comprises a C flag (Critical link flag) of 1 bit, a 2 bits NLB field (number of link backup paths), and each of the backup paths encoding, which consists of the path length PLEN of 3 bits indicating the length of the path (i.e., the number of nodes), and the encoding of the sequence of nodes along the path such as encodings for nodes PN1, ..., PNm. Note that two ends of a link (i.e., the local node and the adjacent/remote node of the link) are not needed in the path. The encoding of every node may use the encoding of a local node, which comprises encoded node index size indication (ENSI) and compact node index (CNI).



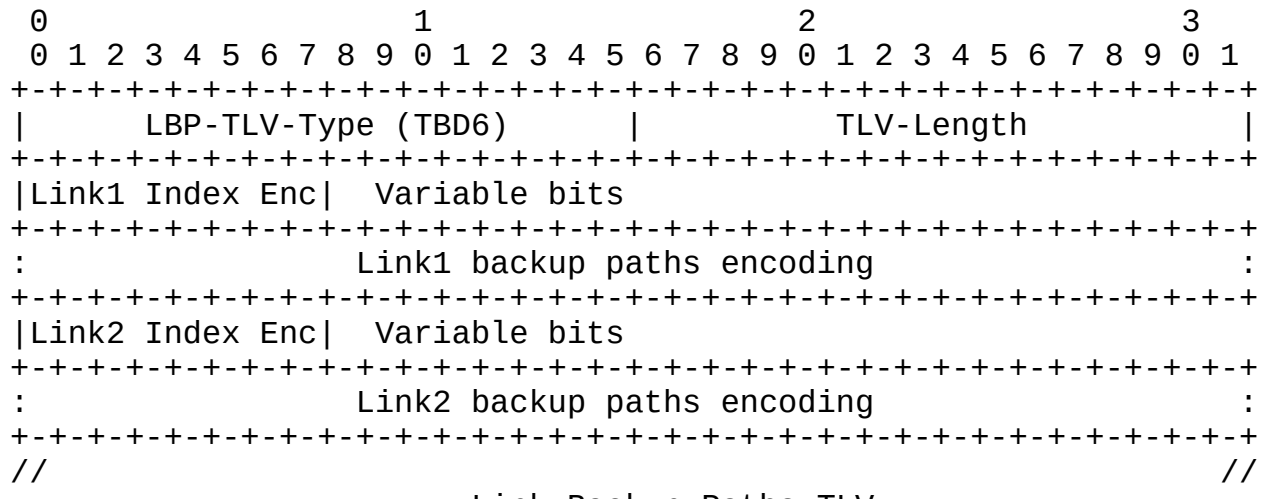
An Example of Link Backup Paths Encoding

Another encoding of the sequence of nodes along the path uses one encoded node index size indication (ENSI) for all the nodes in the path. Thus we have the following Link Backup Paths Encoding.



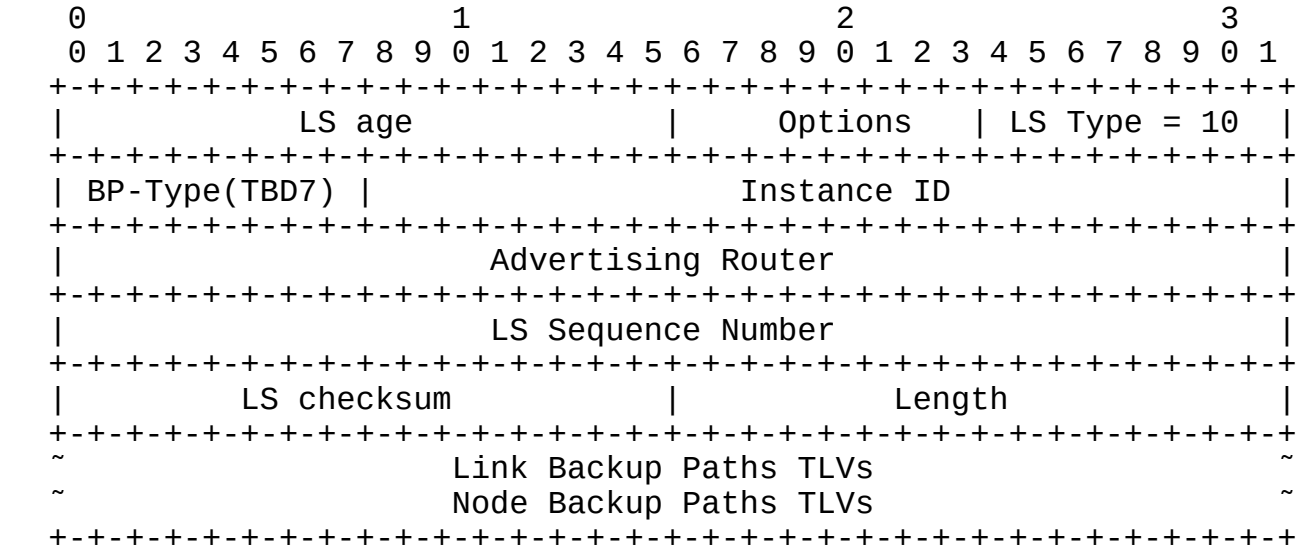
Another Example of Link Backup Paths Encoding

A new TLV called Link Backup Paths TLV is defined below. It may include multiple links and their backup paths. Each link is represented by its index encoding, which is followed by its link backup paths encoding.



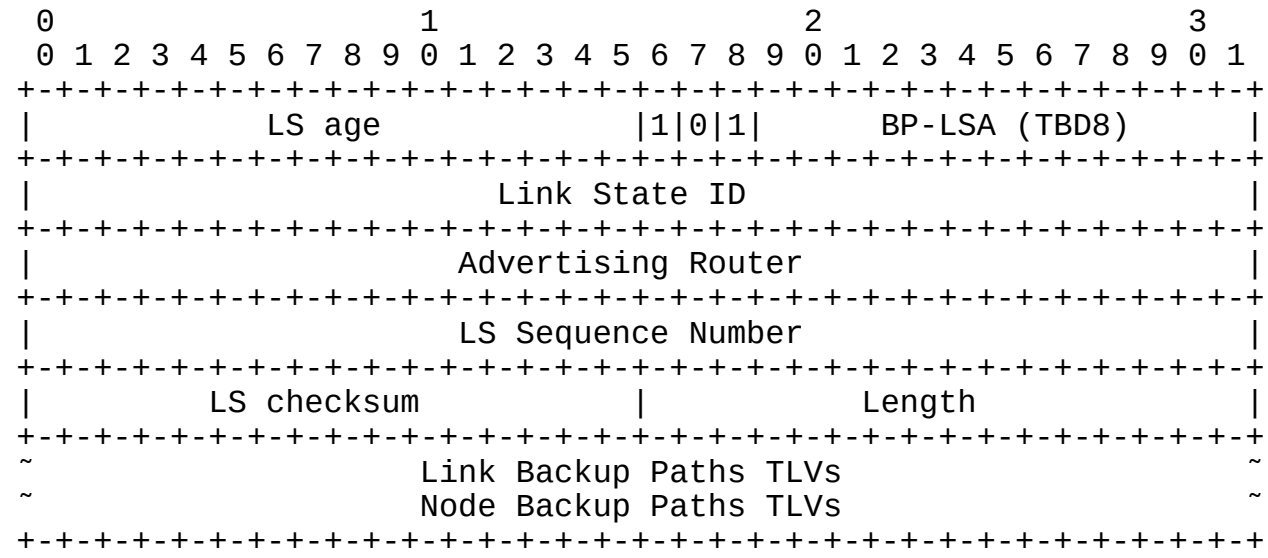
Link Backup Paths TLV

For OSPFv2, an Opaque LSA of a new opaque type (TBD7), containing node backup paths TLVs and link backup paths TLVs, is used to flood the backup paths from the leader of an area to all the other nodes in the area.



OSPFv2: Backup Paths Opaque LSA

For OSPFv3, an area scope LSA of a new LSA function code (TBD8), containing node backup paths TLVs and link backup paths TLVs, is used to flood the backup paths from the leader of an area to all the other nodes in the area.

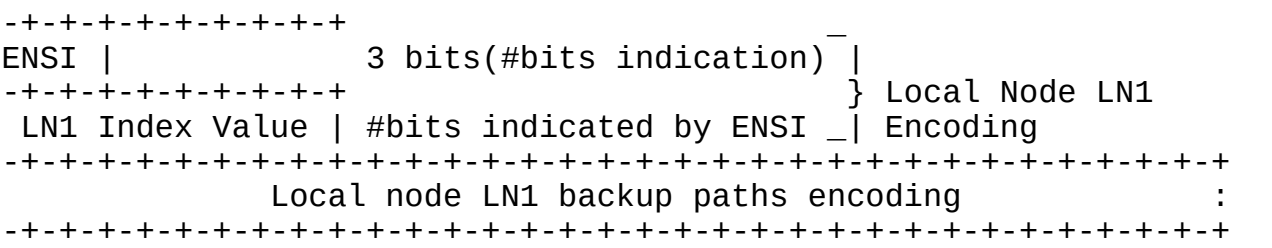


OSPFv3: Backup Paths LSA

The U-bit is set to 1, and the scope is set to 01 for area-scoping.

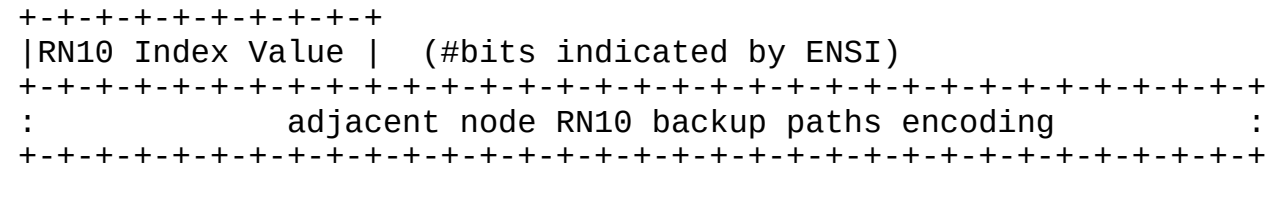
6.2.2.2. Backup Paths in Links TLV

A local node and its backup paths can be encoded in the following format. It is the local node (such as local node LN1) encoding followed by the local node backup paths encoding, which is the same as the node backup paths encoding described in Section 6.2.2.1.



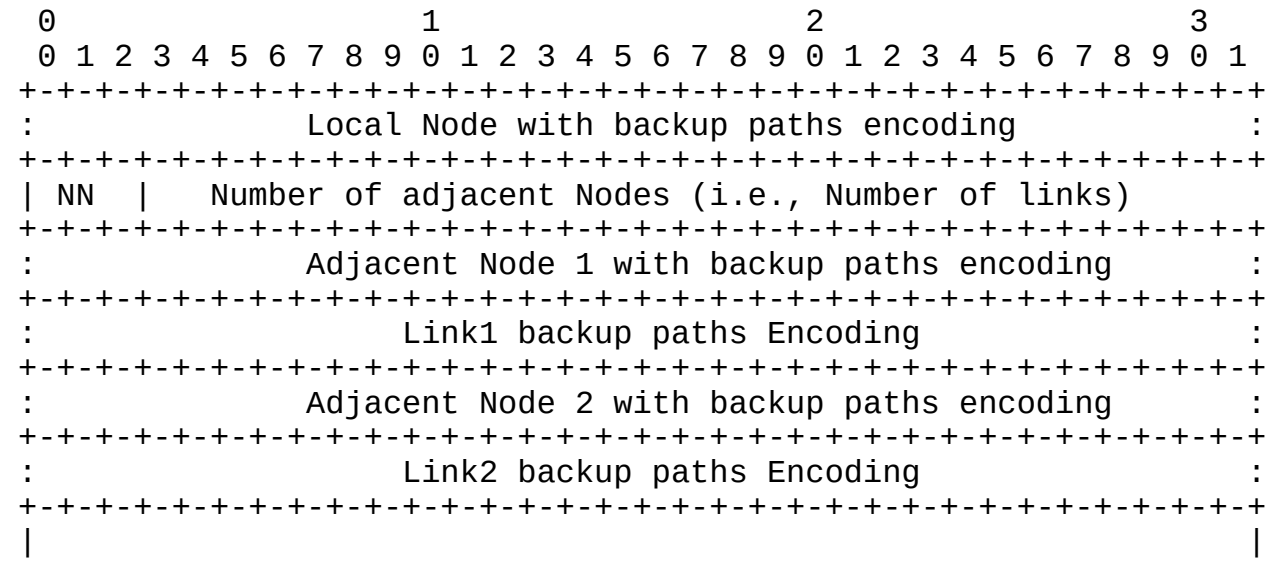
Local Node with Backup Paths Encoding

A adjacent node and its backup paths can be encoded in the following format. It is the adjacent node (such as adjacent node RN10) index value followed by the adjacent node backup paths encoding, which is the same as the node backup paths encoding described in Section 6.2.2.1.



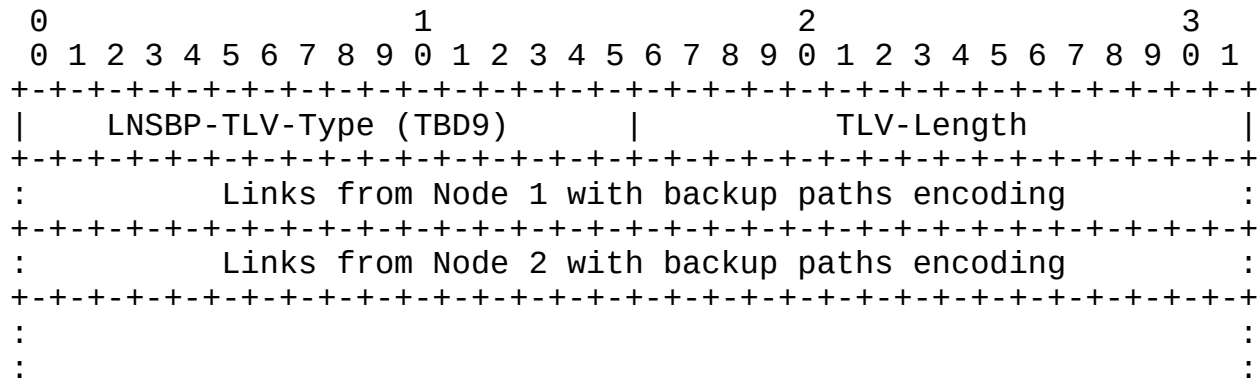
Adjacent Node with Backup Paths Encoding

The links between a local node and a number of its adjacent nodes, the backup paths for each of the nodes, and the backup paths for each of the links can be encoded in the following format. It is called Links from Node with Backup Paths Encoding.



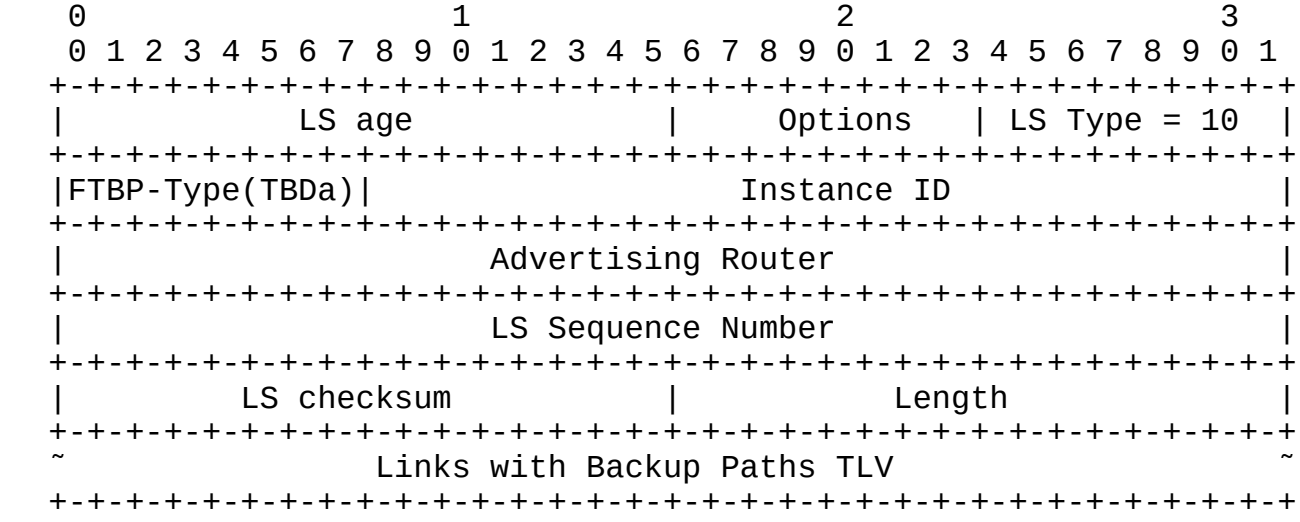
Links from Node with Backup Paths Encoding

A new TLV called Links with Backup Paths TLV is defined below. It includes a number of Links from Node with Backup Paths Encodings described above. This TLV contains both the flooding topology and the backup paths for the links and nodes on the flooding topology.



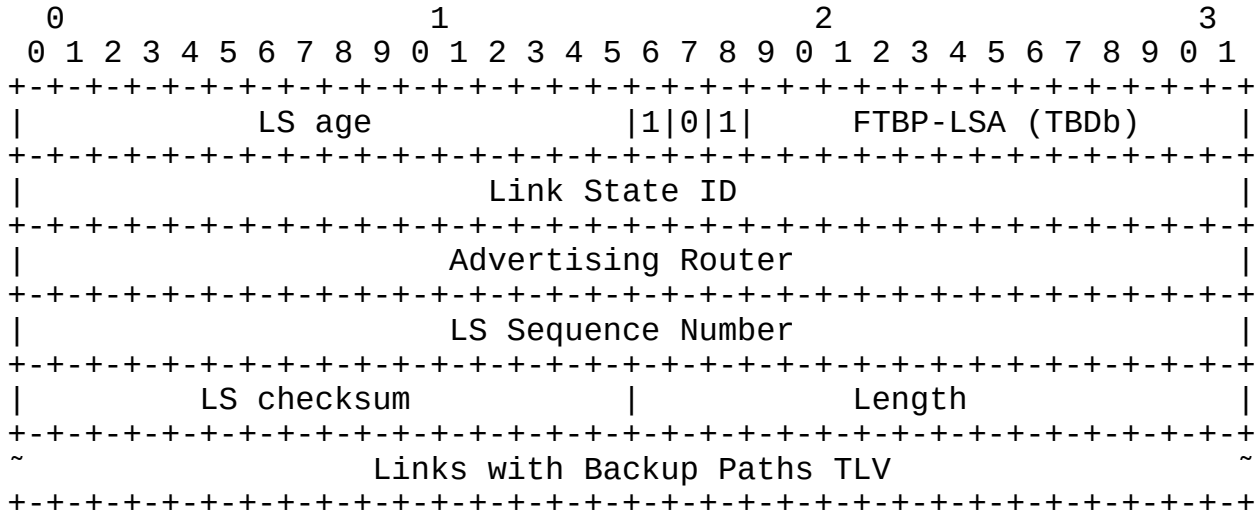
Links with Backup Paths TLV

For OSPFv2, an Opaque LSA of a new opaque type (TBDa), called Flooding Topology with Backup Paths (FTBP) Opaque LSA, containing a Links with Backup Paths TLV, is used to flood the flooding topology with backup paths from the leader of an area to all the other nodes in the area.



OSPFv2: Flooding Topology with Backup Paths (FTBP) Opaque LSA

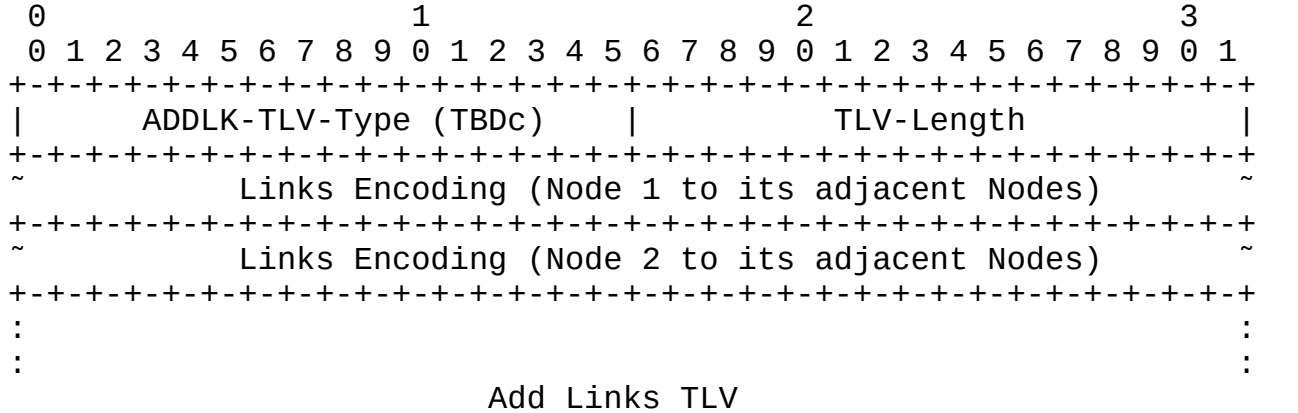
For OSPFv3, an area scope LSA of a new LSA function code (TBDb), containing a Links with Backup Paths TLV, is used to flood the flooding topology with backup paths from the leader of an area to all the other nodes in the area.



OSPFv3: Flooding Topology with Backup Paths (FTBP) LSA

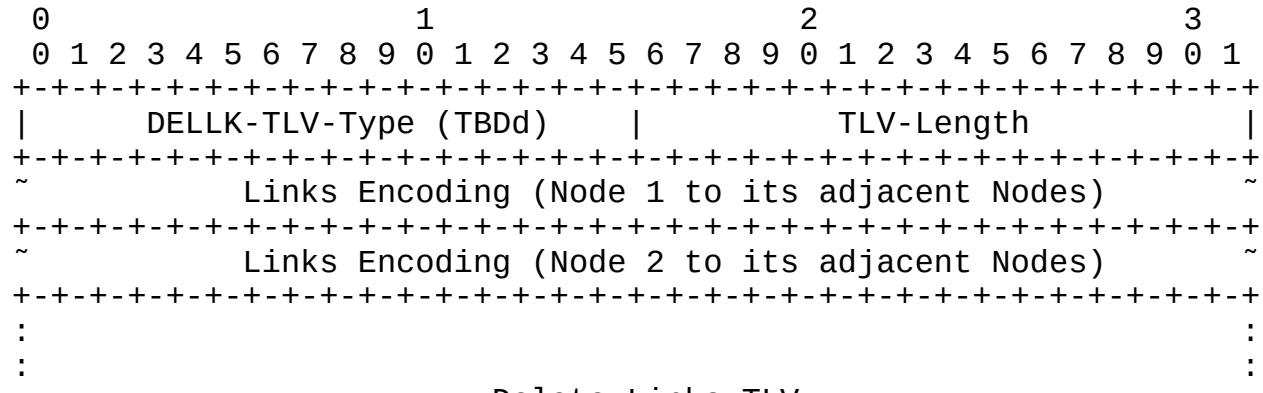
6.2.3. Message for Incremental Changes

For adding some links to the flooding topology, we define a new TLV called Add Links TLVs of the following format. When some new links are added to the flooding topology, the leader may not flood the whole flooding topology with the new links to all the other nodes. It may just flood these new links. After receiving these new links, each of the other nodes adds these new links into the existing flooding topology. When the leader floods the whole flooding topology with the new links to all the other nodes, it removes the LSA for the new links. When removing the LSA for these new links, each of the other nodes does not update the flooding topology (i.e., does not remove these links from the flooding topology).



Add Links TLV

For deleting some links from the flooding topology, we define a new TLV called Delete Links TLVs of the following format. When some old links are removed from the flooding topology, the leader may not flood the whole flooding topology without the old links to all the other nodes. It may just flood these old links. After receiving these old links, each of the other nodes deletes these old links from the existing flooding topology. When the leader floods the whole flooding topology without the old links to all the other nodes, it removes the LSA for the old links. When removing the LSA for these old links, each of the other nodes does not update the flooding topology (i.e., does not add these links into the flooding topology).



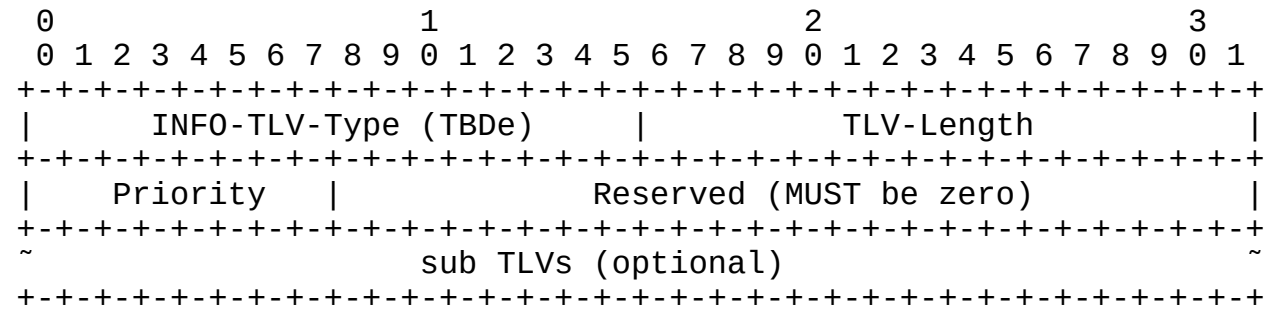
Delete Links TLV

The Add Links TLVs and Delete Links TLVs should be in a separate LSA instance. The LSA can be a Flooding Topology LSA defined above. Alternatively, we may define a new LSA for these TLVs.

6.2.4. Leaders Selection

The leader or Designated Router (DR) selection for a broadcast link is about selecting two leaders: a DR and Backup DR. This is generalized to select two or more leaders for an area: the primary/ first leader (or leader for short), the secondary leader, the third leader and so on.

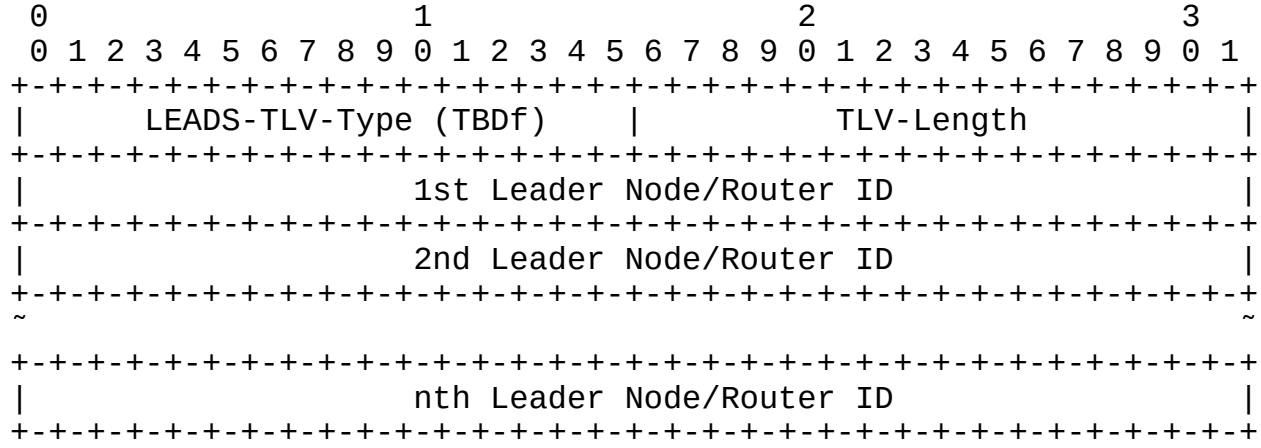
A new TLV is defined to include the information on flooding reduction of a node, which is called Flooding Reduction Information TLV or Information TLV for short. This TLV is generated by every node that supports flooding reduction in general. Every node originates a RI LSA with a Flooding Reduction Information TLV containing its priority to become a leader. The format of the TLV is as follows.



Flooding Reduction Information TLV

A Priority field of eight bits is defined in the TLV to indicate the priority of the node originating the TLV to become the leader node in central mode.

A sub-TLV called leaders sub-TLV is defined. It has the following format.



Leaders sub-TLV

When a node selects itself as a leader, it originates a RI LSA containing the leader in a leaders sub-TLV.

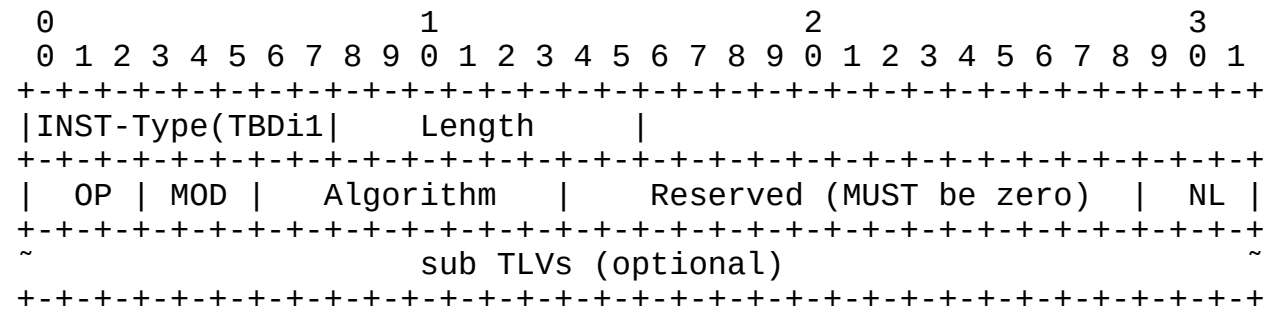
After the first leader node is down, the other leaders will be promoted. The secondary leader becomes the first leader, the third leader becomes the secondary leader, and so on. When a node selects itself as the n-th leader, it originates a RI LSA with a Leaders sub-TLV containing n leaders.

7. Extensions to IS-IS

The extensions to IS-IS is similar to OSPF.

7.1. Extensions for Operations

A new TLV for operations is defined in IS-IS LSP. It has the following format and contains the same contents as the Flooding Reduction Instruction TLV defined in OSPF RI LSA.

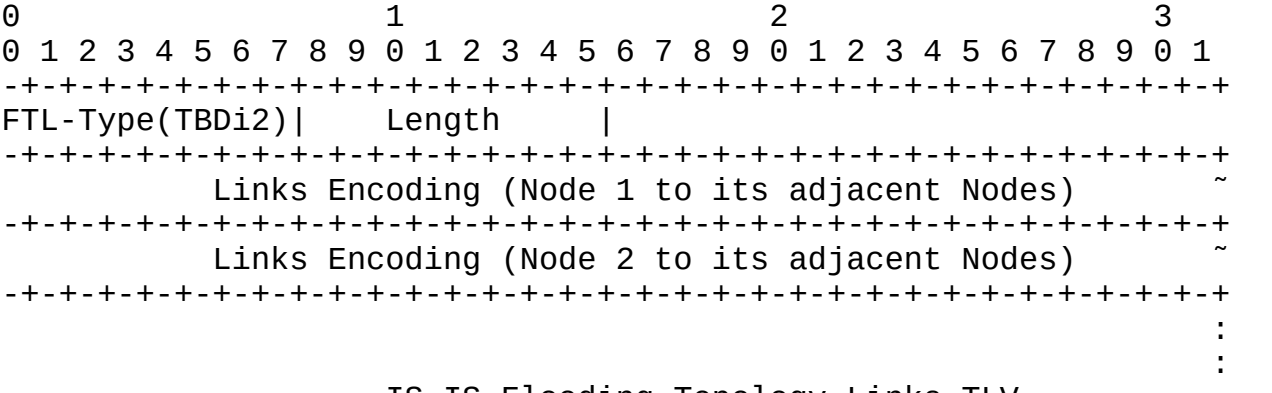


IS-IS Flooding Reduction Instruction TLV

7.2. Extensions for Centralized Mode

7.2.1. TLV for Flooding Topology

A new TLV for the encodings of the links in the flooding topology is defined. It has the following format and contains the same contents as the Flooding Topology Links TLV defined in OSPF Flooding Topology Opaque LSA.

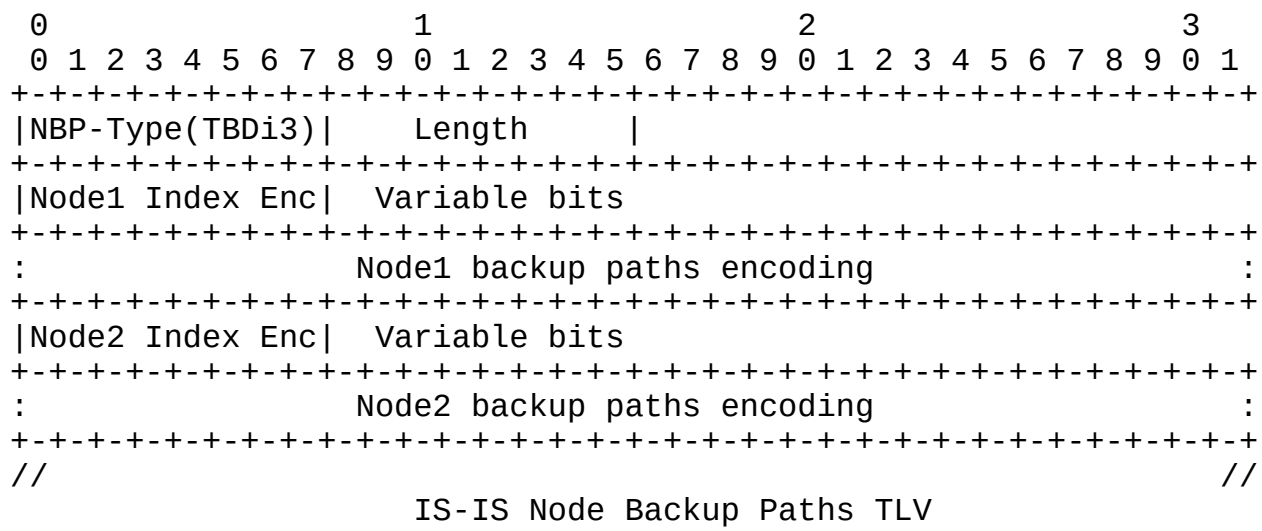


IS-IS Flooding Topology Links TLV

7.2.2. Encodings for Backup Paths

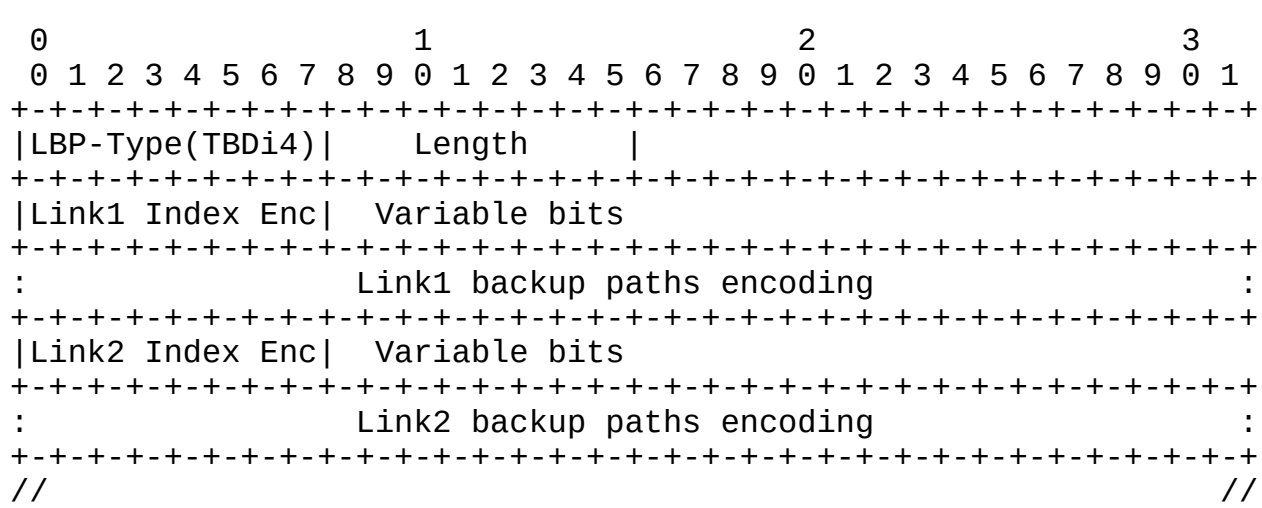
7.2.2.1. TLVs for Backup Paths

For flooding backup paths separately, we define two TLVs: IS-IS Node Backup Paths TLV and IS-IS Link Backup Path TLV. The former has the following format and contains the same contents as Node Backup Paths TLV in OSPF.



IS-IS Node Backup Paths TLV

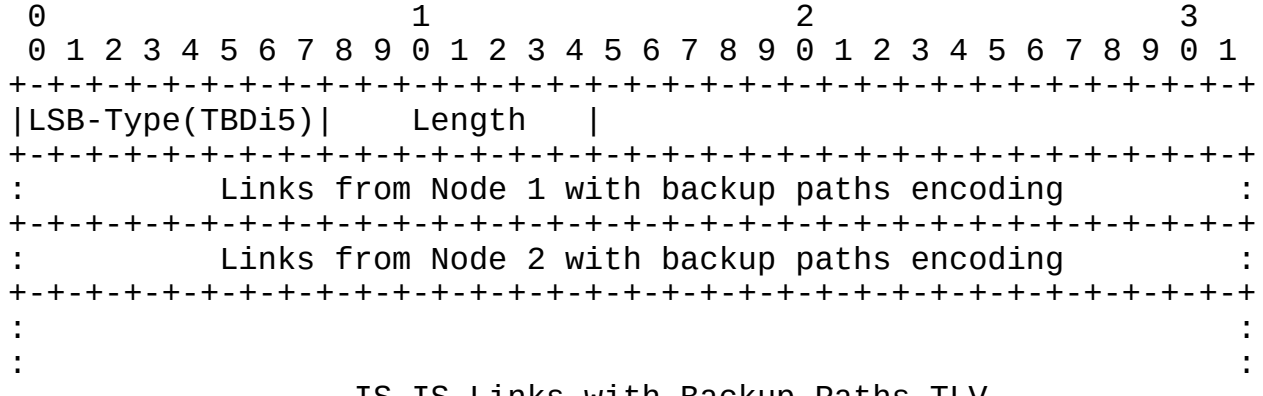
The latter has the following format and contains the same contents as Link Backup Paths TLV in OSPF.



IS-IS Link Backup Paths TLV

7.2.2.2. Backup Paths in Links TLV

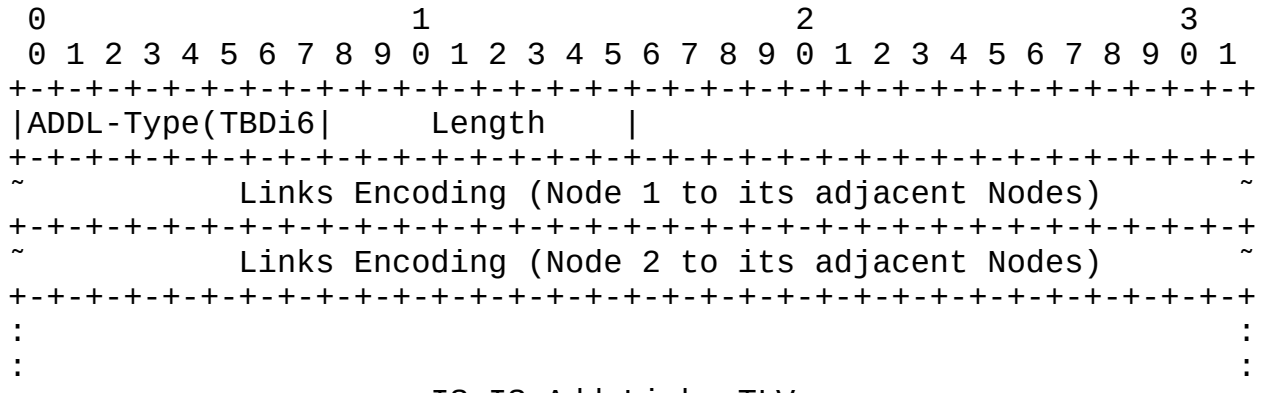
A new TLV is defined to integrate the backup paths with the links on the flooding topology. It has the following format and contains the same contents as the Links with Backup Paths TLV in OSPF.



IS-IS Links with Backup Paths TLV

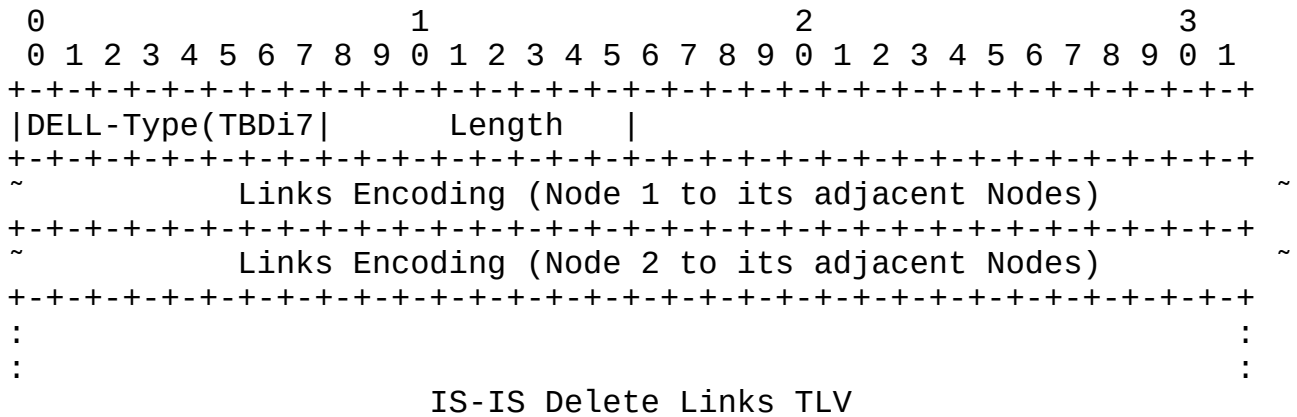
7.2.3. TLVs for Incremental Changes

Similar to Add Links TLV in OSPF, a new TLV called IS-IS Add Links TLV is defined. It has the following format and contains the same contents as Add Links TLV in OSPF.



IS-IS Add Links TLV

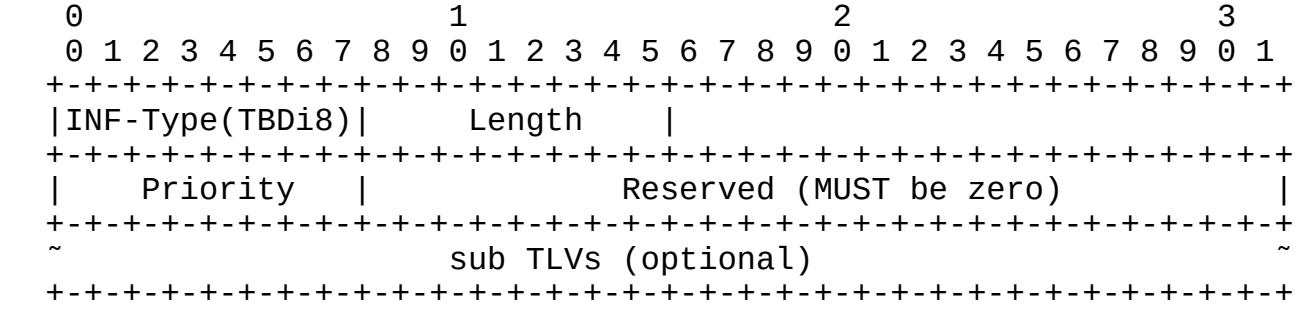
Similar to Delete Links TLV in OSPF, a new TLV called IS-IS Delete Links TLV is defined. It has the following format and contains the same contents as Delete Links TLV in OSPF.



IS-IS Delete Links TLV

7.2.4. Leaders Selection

Similar to Flooding Reduction Information TLV in OSPF, a new TLV called IS-IS Flooding Reduction Information TLV is defined. It has the following format and contains the same contents as Flooding Reduction Information TLV in OSPF.



IS-IS Flooding Reduction Information TLV

8. Flooding Behavior

This section describes the revised flooding behavior for a node having at least one link on the flooding topology. The revised flooding procedure MUST flood an LS to every node in the network in any case, as the standard flooding procedure does.

8.1. Nodes Perform Flooding Reduction without Failure

8.1.1. Receiving an LS

When a node receives a newer LS that is not originated by itself from one of its interfaces, it floods the LS only to all the other interfaces that are on the flooding topology.

When the LS is received from an interface on the flooding topology, it is flooded only to all the other interfaces that are on the flooding topology. When the LS is received on an interface that is not on the flooding topology, it is also flooded only to all the other interfaces that are on the flooding topology.

In any case, the LS must not be transmitted back to the receiving interface.

Note before forwarding a received LS, the node would do the normal processing as usual.

8.1.2. Originating an LS

When a node originates an LS, it floods the LS to its interfaces on the flooding topology if the LS is a refresh LS (i.e., there is no significant change in the LS comparing to the previous LS); otherwise (i.e., there are significant changes in the LS), it floods the LS to all its interfaces. Choosing flooding the LS with significant changes to all the interfaces instead of limiting to the interfaces on the flooding topology would speed up the distribution of the significant link state changes.

8.1.3. Establishing Adjacencies

Adjacencies being established can be classified into two categories: adjacencies to new nodes and adjacencies to existing nodes.

8.1.3.1. Adjacency to New Node

An adjacency to a new node is an adjacency between a node (say node A) on the flooding topology and the new node (say node Y) which is not on the flooding topology. There is not any adjacency between node Y and a node in the network area.

When new node Y is up and connected to node A, node A assumes that node Y and the link between node Y and node A are on the flooding topology until a new flooding topology is computed and built. Node A may determine whether node Y is a new node through checking if node Y is reachable or on the flooding topology.

The procedure for establishing the adjacency between node A and node Y is the existing normal procedure unchanged. After the status of the adjacency reaches to Exchange or Full, node A sends node Y every new or updated LS that node A receives or originates.

8.1.3.2. Adjacency to Existing Node

An adjacency to an existing node is an adjacency between a node (say node A) on the flooding topology and the existing node (say node X) which exists on the flooding topology. There are some adjacencies between node X and some nodes in the network area.

When existing node X is connected to node A after a link between node X and node A is up, node A assumes that the link connecting node A and node X is not on the flooding topology until a new flooding topology is computed and built. Node A may determine whether node X is an existing node through checking if node X is reachable or on the flooding topology.

The procedure for establishing the adjacency between node A and node X is the existing normal procedure unchanged. Node A does not send node X any new or updated LS that node A receives or originates even after the status of the adjacency reaches to Exchange or Full.

8.2. An Exception Case

During an LS flooding, one or multiple link and node failures may happen. Some failures do not split the flooding topology, thus do not affect the flooding behavior. For example, multiple failures of the links not on the flooding topology do not split the flooding topology and do not affect the flooding behavior. The sections below focus on the failures that may split the flooding topology.

8.2.1. A Critical Failure

For a link failure, if the link is a critical link on the flooding topology, then the LS is flooded through a backup path for the link and the remaining flooding topology until a new flooding topology is computed and built; otherwise, the flooding behavior in Section 8.1 follows.

Similarly, for a node failure, if the node is a critical node on the flooding topology, then the LS is flooded through backup paths for the node and the remaining flooding topology until a new flooding topology is computed and built; otherwise, the flooding behavior in Section 8.1 follows.

8.2.2. Multiple Failures

For multiple link failures, if the number of the failed links on the flooding topology is greater than or equal to two, then the LS is flooded through a backup path for each of the failed links on the flooding topology and the remaining flooding topology until a new

flooding topology is computed and built; otherwise, the flooding behavior in Section 8.1 follows.

If all the backup paths for some of the failed links are broken by some failures, the LS is flooded to all interfaces (except where it is received from) until a new flooding topology is computed and built.

For multiple node failures, the LS is flooded through the backup paths for each of the failed nodes and the remaining flooding topology until a new flooding topology is computed and built; otherwise, the flooding behavior in Section 8.1 follows.

If the backup paths for some of the failed nodes are broken by some failures, the LS is flooded to all interfaces (except where it is received from) until a new flooding topology is computed and built.

Note that if it can be quickly determined that the flooding topology is not split by the failures, the flooding behavior in Section 8.1 may follow.

9. Security Considerations

This document does not introduce any security issue.

10. IANA Considerations

10.1. OSPFv2

Under Registry Name: OSPF Router Information (RI) TLVs [RFC7770], IANA is requested to assign two new TLV values for OSPF flooding reduction as follows:

TLV Value	TLV Name	reference
11	Instruction TLV	This document
12	Information TLV	This document

Under the registry name "Opaque Link-State Advertisements (LSA) Option Types" [RFC5250], IANA is requested to assign new Opaque Type registry values for FT LSA, BP LSA, FTBP LSA as follows:

Registry Value	Opaque Type	reference
10	FT LSA	This document
11	BP LSA	This document
12	FTBP LSA	This document

IANA is requested to create and maintain new registries:

o OSPFv2 FT LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	OSPFv2 FT LSA TLV Name	Definition
0	Reserved	
1	FT Links TLV	see Section 6.2.1
2-32767	Unassigned	
32768-65535	Reserved	

o OSPFv2 BP LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	OSPFv2 TBPLSA TLV Name	Definition
0	Reserved	
1	Node Backup Paths TLV	see Section 6.2.2
2	Link Backup Paths TLV	see Section 6.2.2
3-32767	Unassigned	
32768-65535	Reserved	

o OSPFv2 FTBP LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	OSPFv2 FTBP LSA TLV Name	Definition
0	Reserved	
1	Links with Backup Paths TLV	see Section 6.2.2
2-32767	Unassigned	
32768-65535	Reserved	

10.2. OSPFv3

Under the registry name "OSPFv3 LSA Function Codes", IANA is requested to assign new registry values for FT LSA, BP LSA, FTBP LSA as follows:

Value	LSA Function Code Name	reference
16	FT LSA	This document
17	BP LSA	This document
18	FTBP LSA	This document

IANA is requested to create and maintain new registries:

- OSPFv3 FT LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	OSPFv3 FT LSA TLV Name	Definition
0	Reserved	
1	FT Links TLV	see Section 6.2.1
2-32767	Unassigned	
32768-65535	Reserved	

- OSPFv3 BP LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	OSPFv3 TBPLSA TLV Name	Definition
0	Reserved	
1	Node Backup Paths TLV	see Section 6.2.2
2	Link Backup Paths TLV	see Section 6.2.2
3-32767	Unassigned	
32768-65535	Reserved	

- OSPFv3 FTBP LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	OSPFv3 FTBP LSA TLV Name	Definition
0	Reserved	
1	Links with Backup Paths TLV	see Section 6.2.2
2-32767	Unassigned	
32768-65535	Reserved	

10.3. IS-IS

Under Registry Name: IS-IS TLV Codepoints, IANA is requested to assign new TLV values for IS-IS flooding reduction as follows:

Value	TLV Name	Definition
151	FT Links TLV	see Section 7.2.1
152	Node Backup Paths TLV	see Section 7.2.2
153	Link Backup Paths TLV	see Section 7.2.2
154	Links with Backup Paths TLV	see Section 7.2.2
155	Add Links TLV	see Section 7.2.3
156	Delete Links TLV	see Section 7.2.3
157	Instruction TLV	see Section 7.1
158	Information TLV	see Section 7.2.4

11. Acknowledgements

The authors would like to thank Acee Lindem, Zhibo Hu, Robin Li, Stephane Litkowski and Alvaro Retana for their valuable suggestions and comments on this draft.

12. References

12.1. Normative References

[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>.

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <https://www.rfc-editor.org/info/rfc5250>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <https://www.rfc-editor.org/info/rfc7770>.

12.2. Informative References

[I-D.li-dynamic-flooding]
Li, T. and P. Psenak, "Dynamic Flooding on Dense Graphs", draft-li-dynamic-flooding-05 (work in progress), June 2018.

[I-D.shen-isis-spine-leaf-ext]
Shen, N., Ginsberg, L., and S. Thyamagundalu, "IS-IS Routing for Spine-Leaf Topology", draft-shen-isis-spine-leaf-ext-06 (work in progress), June 2018.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

Appendix A. Algorithms to Build Flooding Topology

There are many algorithms to build a flooding topology. A simple and efficient one is briefed below.

- Select a node R according to a rule such as the node with the biggest/smallest node ID;
- Build a tree using R as root of the tree (details below); and then
- Connect k (k>=0) leaves to the tree to have a flooding topology (details follow).

A.1. Algorithms to Build Tree without Considering Others

An algorithm for building a tree from node R as root starts with a candidate queue Cq containing R and an empty flooding topology Ft:

1. Remove the first node A from Cq and add A into Ft
2. If Cq is empty, then return with Ft

3. Suppose that node Xi (i = 1, 2, ..., n) is connected to node A and not in Ft and X1, X2, ..., Xn are in a special order. For example, X1, X2, ..., Xn are ordered by the cost of the link between A and Xi. The cost of the link between A and Xi is less than the cost of the link between A and Xj (j = i + 1). If two costs are the same, Xi's ID is less than Xj's ID. In another example, X1, X2, ..., Xn are ordered by their IDs. If they are not ordered, then make them in the order.

4. Add Xi (i = 1, 2, ..., n) into the end of Cq, goto step 1.

Another algorithm for building a tree from node R as root starts with a candidate queue Cq containing R and an empty flooding topology Ft:

1. Remove the first node A from Cq and add A into Ft

2. If Cq is empty, then return with Ft

3. Suppose that node Xi (i = 1, 2, ..., n) is connected to node A and not in Ft and X1, X2, ..., Xn are in a special order. For example, X1, X2, ..., Xn are ordered by the cost of the link between A and Xi. The cost of the link between A and Xi is less than the cost of the link between A and Xj (j = i + 1). If two costs are the same, Xi's ID is less than Xj's ID. In another example, X1, X2, ..., Xn are ordered by their IDs. If they are not ordered, then make them in the order.

4. Add Xi (i = 1, 2, ..., n) into the front of Cq and goto step 1.

A third algorithm for building a tree from node R as root starts with a candidate list Cq containing R associated with cost 0 and an empty flooding topology Ft:

1. Remove the first node A from Cq and add A into Ft

2. If all the nodes are on Ft, then return with Ft

3. Suppose that node A is associated with a cost Ca which is the cost from root R to node A, node Xi (i = 1, 2, ..., n) is connected to node A and not in Ft and the cost of the link between A and Xi is LCi (i=1, 2, ..., n). Compute Ci = Ca + LCi, check if Xi is in Cq and if Cxi (cost from R to Xi) < Ci. If Xi is not in Cq, then add Xi with cost Ci into Cq; If Xi is in Cq, then If Cxi > Ci then replace Xi with cost Cxi by Xi with Ci in Cq; If Cxi == Ci then add Xi with cost Ci into Cq.

4. Make sure Cq is in a special order. Suppose that Ai (i=1, 2, ..., m) are the nodes in Cq, Cai is the cost associated with Ai,

and IDi is the ID of Ai. One order is that for any k = 1, 2, ..., m-1, Cak < Caj (j = k+1) or Cak = Caj and IDk < IDj. Goto step 1.

A.2. Algorithms to Build Tree Considering Others

An algorithm for building a tree from node R as root with consideration of others's support for flooding reduction starts with a candidate queue Cq containing R associated with previous hop PH=0 and an empty flooding topology Ft:

1. Remove the first node A that supports flooding reduction from the candidate queue Cq if there is such a node A; otherwise (i.e., if there is not such node A in Cq), then remove the first node A from Cq. Add A into the flooding topology Ft.

2. If Cq is empty or all nodes are on Ft, then return with Ft

3. Suppose that node Xi (i = 1, 2, ..., n) is connected to node A and not in the flooding topology Ft and X1, X2, ..., Xn are in a special order considering whether some of them that support flooding reduction (. For example, X1, X2, ..., Xn are ordered by the cost of the link between A and Xi. The cost of the link between A and Xi is less than that of the link between A and Xj (j = i + 1). If two costs are the same, Xi's ID is less than Xj's ID. The cost of a link is redefined such that 1) the cost of a link between A and Xi both support flooding reduction is much less than the cost of any link between A and Xk where Xk with F=0; 2) the real metric of a link between A and Xi and the real metric of a link between A and Xk are used as their costs for determining the order of Xi and Xk if they all (i.e., A, Xi and Xk) support flooding reduction or none of Xi and Xk support flooding reduction.

4. Add Xi (i = 1, 2, ..., n) associated with previous hop PH=A into the end of the candidate queue Cq, and goto step 1.

Another algorithm for building a tree from node R as root with consideration of others' support for flooding reduction starts with a candidate queue Cq containing R associated with previous hop PH=0 and an empty flooding topology Ft:

1. Remove the first node A that supports flooding reduction from the candidate queue Cq if there is such a node A; otherwise (i.e., if there is not such node A in Cq), then remove the first node A from Cq. Add A into the flooding topology Ft.

2. If Cq is empty or all nodes are on Ft, then return with Ft.

3. Suppose that node Xi (i = 1, 2, ..., n) is connected to node A and not in the flooding topology Ft and X1, X2, ..., Xn are in a special order considering whether some of them support flooding reduction. For example, X1, X2, ..., Xn are ordered by the cost of the link between A and Xi. The cost of the link between A and Xi is less than the cost of the link between A and Xj (j = i + 1). If two costs are the same, Xi's ID is less than Xj's ID. The cost of a link is redefined such that 1) the cost of a link between A and Xi both support flooding reduction is much less than the cost of any link between A and Xk where Xk does not support flooding reduction; 2) the real metric of a link between A and Xi and the real metric of a link between A and Xk are used as their costs for determining the order of Xi and Xk if they all (i.e., A, Xi and Xk) support flooding reduction or none of Xi and Xk supports flooding reduction.

4. Add Xi (i = 1, 2, ..., n) associated with previous hop PH=A into the front of the candidate queue Cq, and goto step 1.

A third algorithm for building a tree from node R as root with consideration of others' support for flooding reduction (using flag F = 1 for support, and F = 0 for not support in the following) starts with a candidate list Cq containing R associated with low order cost Lc=0, high order cost Hc=0 and previous hop ID PH=0, and an empty flooding topology Ft:

1. Remove the first node A from Cq and add A into Ft.

2. If all the nodes are on Ft, then return with Ft

3. Suppose that node A is associated with a cost Ca which is the cost from root R to node A, node Xi (i = 1, 2, ..., n) is connected to node A and not in Ft and the cost of the link between A and Xi is LCi (i=1, 2, ..., n). Compute Ci = Ca + LCi, check if Xi is in Cq and if Cxi (cost from R to Xi) < Ci. If Xi is not in Cq, then add Xi with cost Ci into Cq; If Xi is in Cq, then If Cxi > Ci then replace Xi with cost Cxi by Xi with Ci in Cq; If Cxi == Ci then add Xi with cost Ci into Cq.

4. Suppose that node A is associated with a low order cost LCa which is the low order cost from root R to node A and a high order cost HCa which is the high order cost from R to A, node Xi (i = 1, 2, ..., n) is connected to node A and not in the flooding topology Ft and the real cost of the link between A and Xi is Ci (i=1, 2, ..., n). Compute LCxi and HCxi: LCxi = LCa + Ci if both A and Xi have flag F set to one, otherwise LCxi = LCa HCxi = HCa + Ci if A or Xi does not have flag F set to one, otherwise HCxi = HCa If Xi is not in Cq, then add Xi associated with LCxi, HCxi and PH = A

into Cq; If Xi associated with LCxi' and HCxi' and PHxi' is in Cq, then If HCxi' > HCxi then replace Xi with HCxi', LCxi' and PHxi' by Xi with HCxi, LCxi and PH=A in Cq; otherwise (i.e., HCxi' == HCxi) if LCxi' > LCxi , then replace Xi with HCxi', LCxi' and PHxi' by Xi with HCxi, LCxi and PH=A in Cq; otherwise (i.e., HCxi' == HCxi and LCxi' == LCxi) if PHxi' > PH, then replace Xi with HCxi', LCxi' and PHxi' by Xi with HCxi, LCxi and PH=A in Cq.

5. Make sure Cq is in a special order. Suppose that Ai (i=1, 2, ..., m) are the nodes in Cq, HCai and LCai are low order cost and high order cost associated with Ai, and IDi is the ID of Ai. One order is that for any k = 1, 2, ..., m-1, HCak < HCaj (j = k+1) or HCak = HCaj and LCak < LCaj or HCak = HCaj and LCak = LCaj and IDk < IDj. Goto step 1.

A.3. Connecting Leaves

Suppose that we have a flooding topology Ft built by one of the algorithms described above. Ft is like a tree. We may connect k (k >=0) leaves to the tree to have a enhanced flooding topology with more connectivity.

Suppose that there are m (0 < m) leaves directly connected to a node X on the flooding topology Ft. Select k (k <= m) leaves through using a deterministic algorithm or rule. One algorithm or rule is to select k leaves that have smaller or larger IDs (i.e., the IDs of these k leaves are smaller/bigger than the IDs of the other leaves directly connected to node X). Since every node has a unique ID, selecting k leaves with smaller or larger IDs is deterministic.

If k = 1, the leaf selected has the smallest/largest node ID among the IDs of all the leaves directly connected to node X.

For a selected leaf L directly connected to a node N in the flooding topology Ft, select a connection/adjacency to another node from node L in Ft through using a deterministic algorithm or rule.

Suppose that leaf node L is directly connected to nodes Ni (i = 1,2,...,s) in the flooding topology Ft via adjacencies and node Ni is not node N, IDi is the ID of node Ni, and Hi (i = 1,2,...,s) is the number of hops from node L to node Ni in the flooding topology Ft.

One Algorithm or rule is to select the connection to node Nj (1 <= j <= s) such that Hj is the largest among H1, H2, ..., Hs. If there is another node Na (1 <= a <= s) and Hj = Ha, then select the one with smaller (or larger) node ID. That is that if Hj == Ha and IDj < IDa then select the connection to Nj for selecting the one with smaller

node ID (or if Hj == Ha and IDj < IDa then select the connection to Na for selecting the one with larger node ID).

Suppose that the number of connections in total between leaves selected and the nodes in the flooding topology Ft to be added is NLc. We may have a limit to NLc.

Authors' Addresses

Huaimo Chen
Huawei Technologies

Email: huaimo.chen@huawei.com

Dean Cheng
Huawei Technologies

Email: dean.cheng@huawei.com

Mehmet Toy
Verizon
USA

Email: mehmet.toy@verizon.com

Yi Yang
IBM
Cary, NC
United States of America

Email: yyietf@gmail.com