Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 11, 2019

H. Chen
D. Cheng
Huawei Technologies
M. Toy
Verizon
Y. Yang
IBM
A. Wang
China Telecom
X. Liu
Volta Networks
Y. Fan
Casa Systems
L. Liu

January 7, 2019

# LS Flooding Reduction
# draft-cc-lsr-flooding-reduction-01

## 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 July 11, 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, providing an alternative connection between the two end nodes of a link on the flooding topology or between the two end nodes of a path crossing a node on the flooding topology. when a critical link goes down, the backup path for the link provides a connection to connect two parts of a split flooding topology. When a critical node goes down, the backup paths for the paths crossing the node connect all the split parts of the floooding topology 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 for 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 waves of flooding storms, choke of computing resources, slow convergence, oscillating topology changes, and 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.

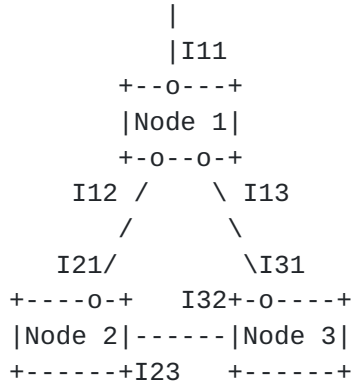



Figure 1

## 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 (i.e., the diameter of 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:

- Centralized Mode: One node in the network builds a flooding topology and floods the flooding topology to all the other nodes in the network (Note: 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. Protect 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. Without any protection against a flooding topology split when multiple failures on the flooding topology happen, we may have a slow convergence. For example, in centralized mode, it takes some time for the leader to detect the failures through receiving the link states, compute a new flooding topology and flood the new flooding topology. In addition, it takes some time for each of the other nodes to receive the new flooding topology (piece by piece), decode it and build it locally. It is better to have some simple and fast methods for protecting the flooding topology split. Thus the convergence is not slowed down.

In one way, when two or more failures on the current flooding topology occur almost in the same time, each of the nodes within a given distance (such as 3 hops) to a failure point, floods the link state (LS) that it receives to all the links (except for the one from which the LS is received) until a new flooding topology is built.

In another way, each node computes and maintains a small number of backup paths. For a backup path for a link L on the flooding topology, a node N computes and maintains it only if the backup path goes through node N. Node N stores the links (e.g., local link L1 and L2) attached to it and on the backup path. When link L fails and there are one or more other failures on the flooding topology, node N adds the links (e.g., L1 and L2) to the flooding topology temporarily until a new flooding topology is built. Suppose that the two end nodes of link L is A and B, and A's ID is smaller than B's. Node N

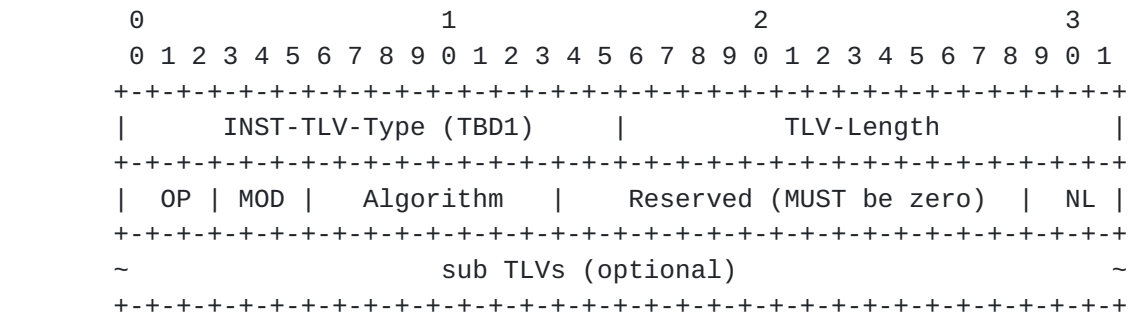
computes a path from A to B with minimum hops and whose links are not on the flooding topology. This path is a backup path for link L.

## 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 flooding reduction (or say flooding reduction in centralized mode).

### 6.1. Extensions for Operations

A new TLV is defined in OSPF RI LSA [RFC7770]. It contains instructions about flooding reduction, and is called Flooding Reduction Instruction TLV or Instruction TLV for short. This TLV is originated from only one node and has the format below.




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 an area to perform flooding reduction.

- 0x010 (N): Roll back to Normal flooding, which instructs the nodes in an area 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): Centralized Mode, which instructs: 1) the nodes in an area to select leaders (primary/designated leader, secondary/ backup leader, and so on); 2) the primary leader to compute a flooding topology and flood it to all the other nodes in the area;

3) every node in the area to receive and use the flooding topology originated by the primary leader.

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

- 0x011 (S): Static Mode, which instructs every node in an area 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 centralized 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 Centralized 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 Centralized 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

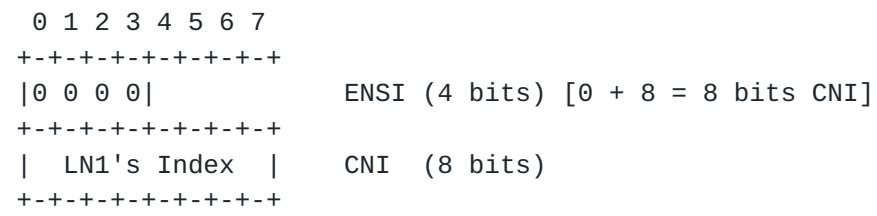
### 6.2.1. Message for Flooding Topology

A flooding topology can be represented by the links in the flooding topology. For the links between a local node and its adjacent (or remote) nodes, we can encode the local node and its adjacent nodes. 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.

Every node orders the nodes by their node IDs (router IDs in OSPF, system IDs in IS-IS) in ascending order, and generates the same sequence of the nodes in the area. The sequence of nodes have the index 0, 1, 2, and so on respectively. Every node in the encoded links is represented by its index.

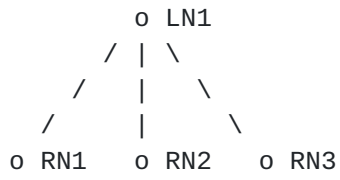
#### 6.2.1.1. Links Encoding

A local node can be encoded in two parts: encoded node index size indication (ENSI) of 4 bits and compact node index (CNI). ENSI value plus 8 gives the size of compact node index. For example, ENSI = 0 indicates that the size of CNIs is 8 bits. In the figure below, Local node LN1 is encoded as ENSI=0 using 4 bits and CNI=LN1's Index using 8 bits. LN1 is encoded in 12 bits in total.



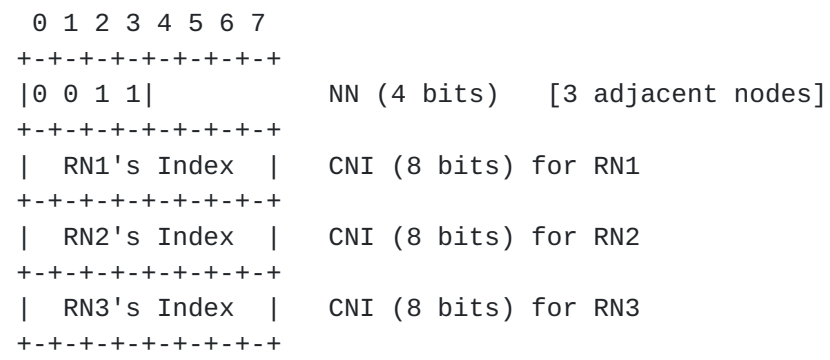

Encoding for local node LN1

The adjacent nodes can be encoded in two parts: Number of Nodes (NN) of 4 bits and compact node indexes (CNIs). The size of CNIs is the same as the local node. For example, local node LN1 has three adjacent nodes RN1, RN2 and RN3 in the following figure.



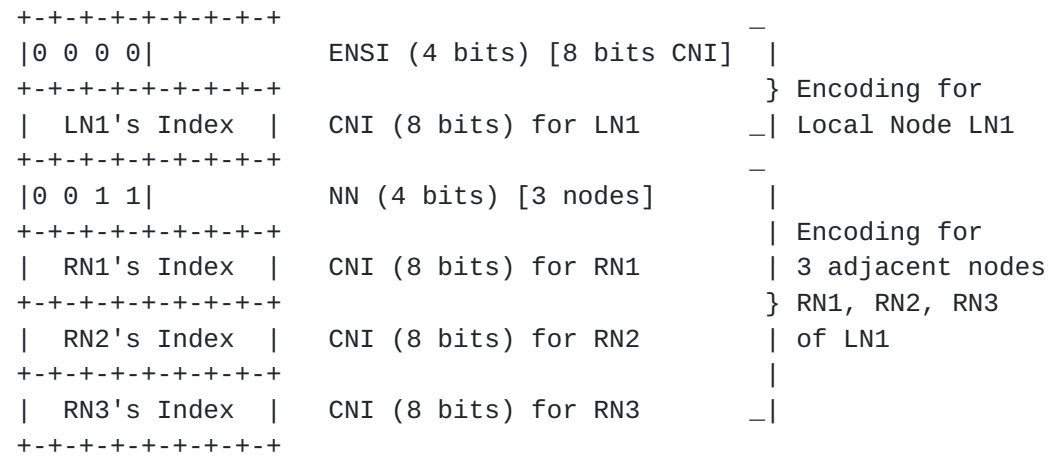

Links from LN1 to its adjacent nodes RN1, RN2 and RN3

These three adjacent nodes RN1, RN2 and RN3 are encoded below in 28 bits (i.e., 3.5 bytes).



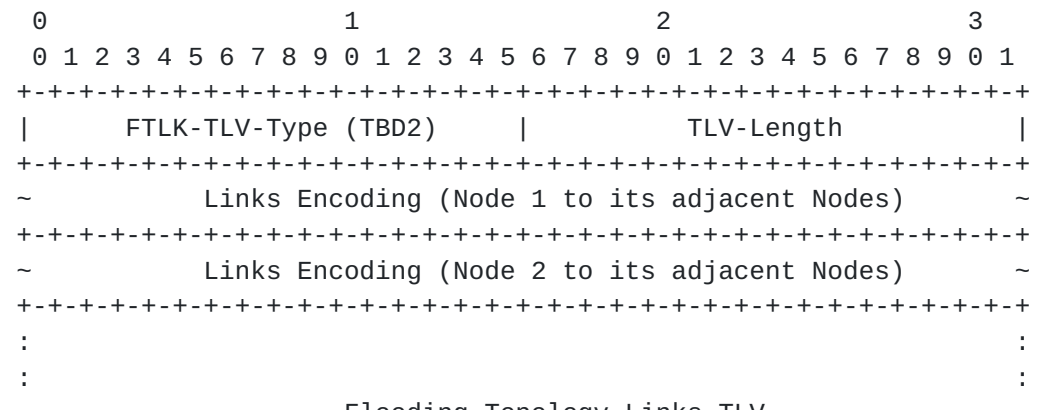

Encoding for three adjacent nodes RN1, RN2 and RN3

The links between a local node and 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 40 bits (i.e., 5 bytes).




Links Encoding for links from LN1 to RN1, RN2 and RN3

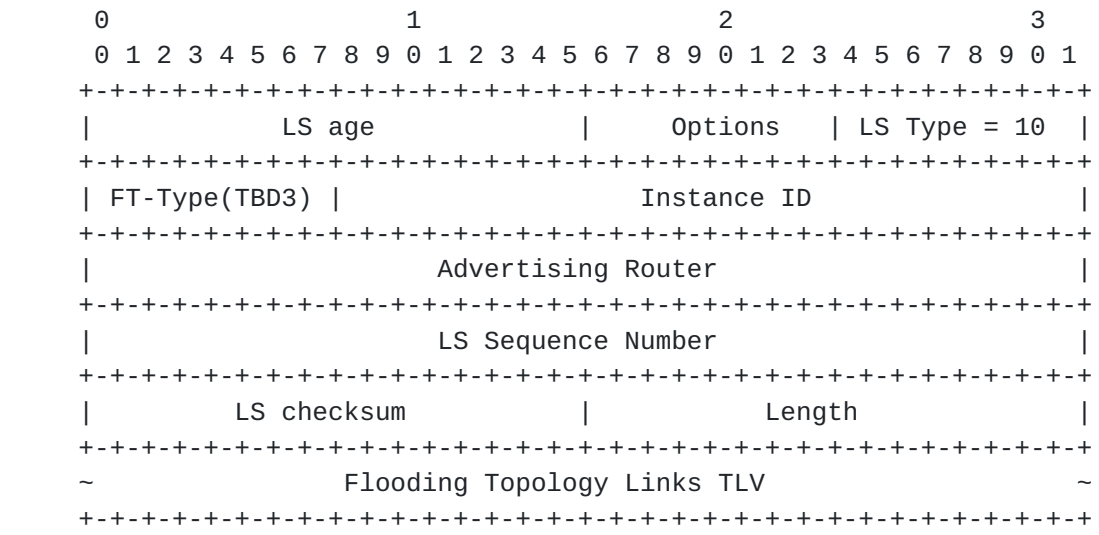
For a flooding topology computed by a leader of an area, it is represented by all the links on the flooding topology. A Type-Length-Value (TLV) of the following format for the links encodings is included in an LSA to represent the flooding topology (FT).




Flooding Topology Links TLV

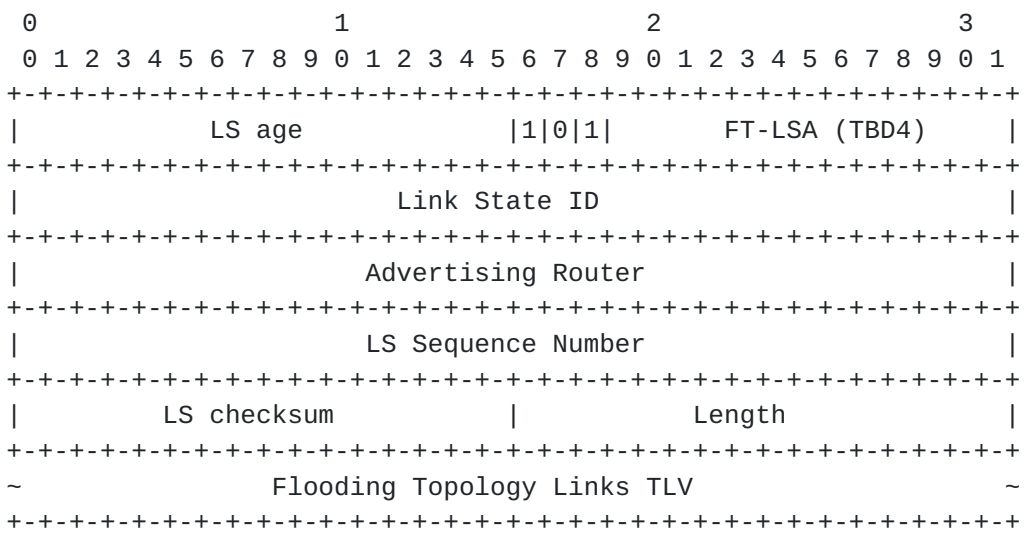
Note that a link between a local node LN and its adjacent node RN is encoded once and as a bi-directional link. That is that if it is encoded in a Links Encoding from LN to RN, then the link from RN to LN is implied or assumed.

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

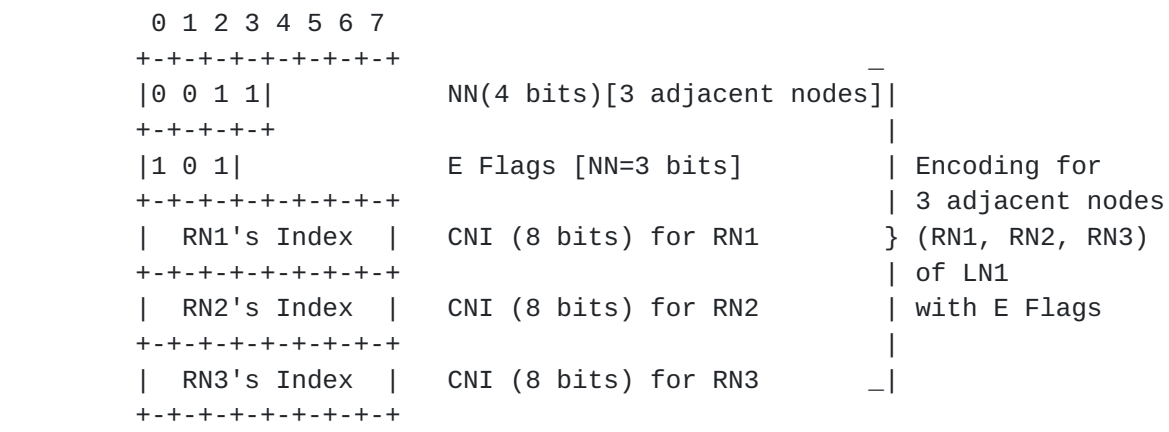
The U-bit is set to 1, and the scope is set to 01 for area-scoping.

#### 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 encoding of LN1's adjacent nodes RN1, RN2 and RN3 with E Flags of 3 bits, which is the value of the NN (the number of adjacent nodes).

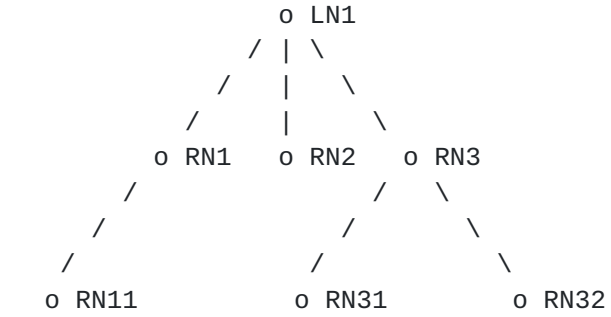



Encoding of LN1's Adjacent Nodes RN1, RN2 and RN3 with E Flags

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 4 bits, E Flags field of NN bits, and CNIs for the adjacent nodes of RNi.

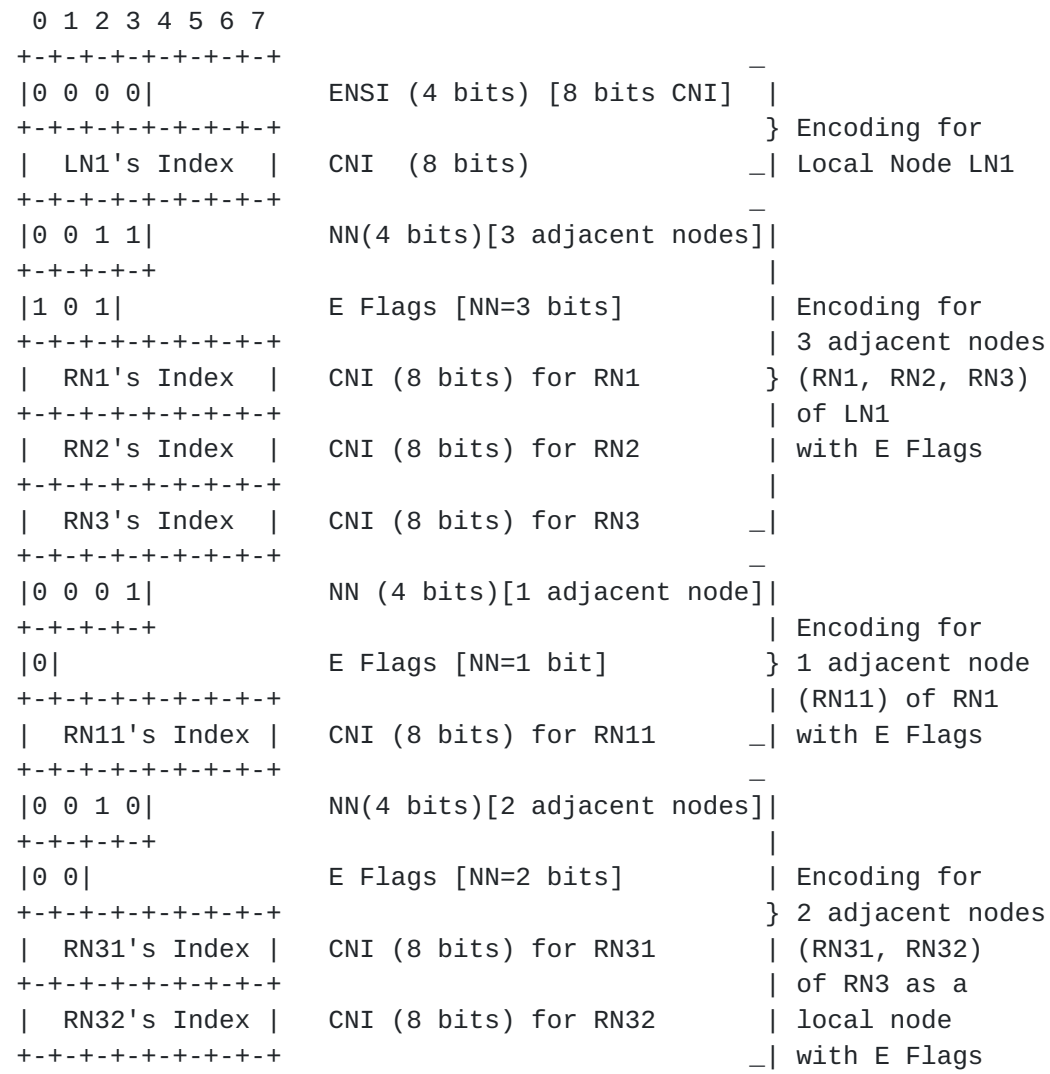
The following is an example of a block encoding for a flooding topology (FT) block below.




FT Block from LN1 to RN1, RN2 and RN3, and to RN11, RN31 and RN32

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 4 bits with value of 3, E Flags field of NN = 3 bits, and the indexes of adjacent nodes RN1, RN2 and RN3.



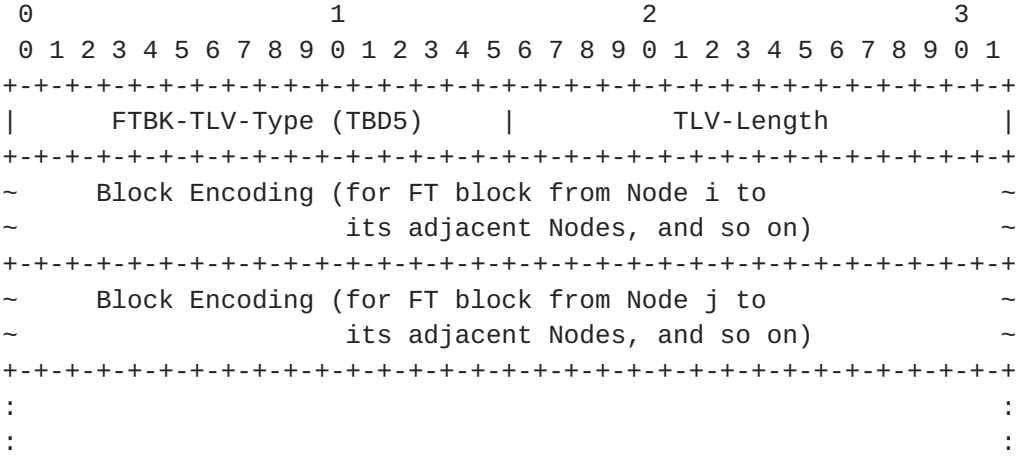

Block Encoding for FT block
from LN1 to RN1, RN2 and RN3, and to RN11, RN31 and RN32

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 4 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 4 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.

A new TLV is defined to contain a number of block encodings, and is called Flooding Topology Blocks TLV or Blocks TLV for short. Its format is illustrated below. This TLV may be used in the place of Links TLV in Section 6.2.1.1 for more efficiency.

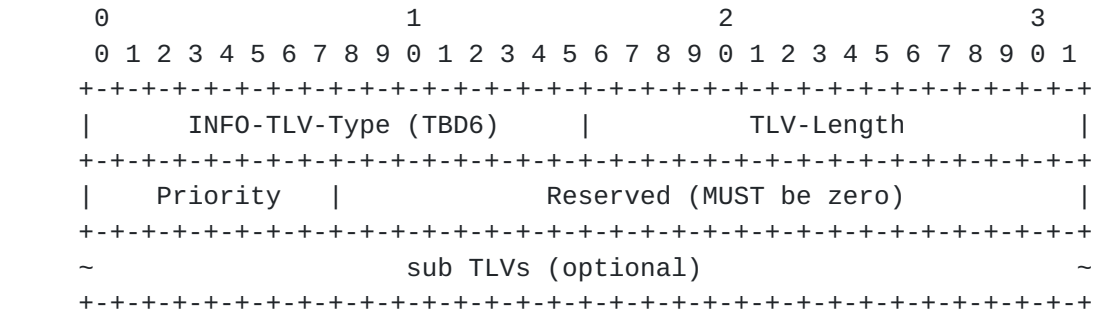



Flooding Topology Blocks TLV

### 6.2.2. 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.

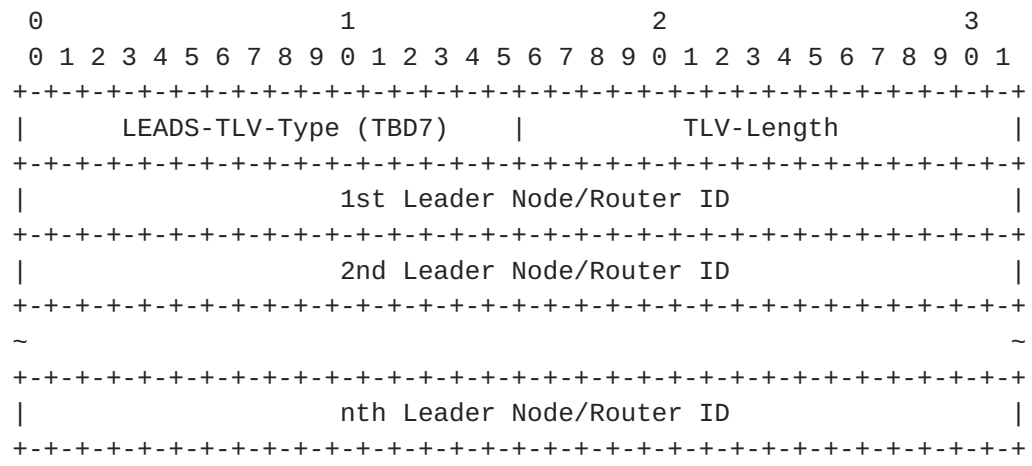
A new TLV is defined to include the information on flooding reduction of a node, which is called Flooding Reduction Information TLV or Information TLV for short. This TLV is generated by every node that supports flooding reduction in general. Every node originates a RI LSA with a Flooding Reduction Information 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 centralized 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.

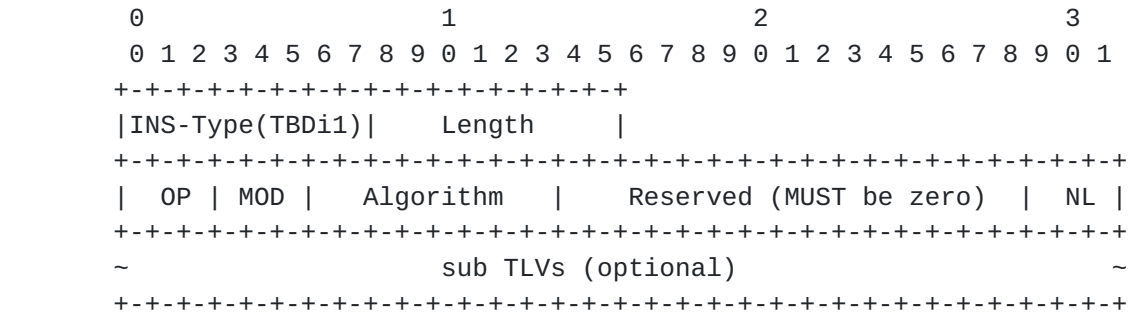
After the first leader node is down, the other leaders will be promoted. The secondary leader becomes the first leader, the third leader becomes the secondary leader, and so on. When a node selects itself as the n-th leader, it originates a RI LSA with a Leaders sub-TLV containing n leaders.

## 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.

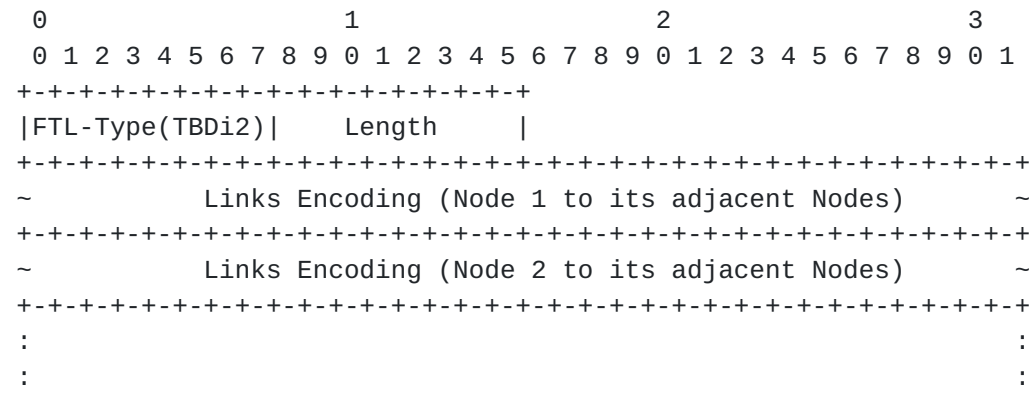



IS-IS Flooding Reduction Instruction TLV

## 7.2. Extensions for Centralized Mode

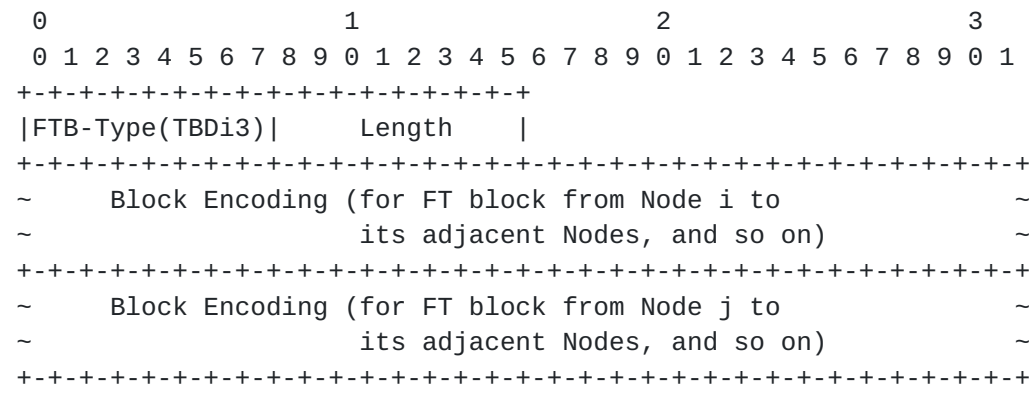
### 7.2.1. TLV for Flooding Topology

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

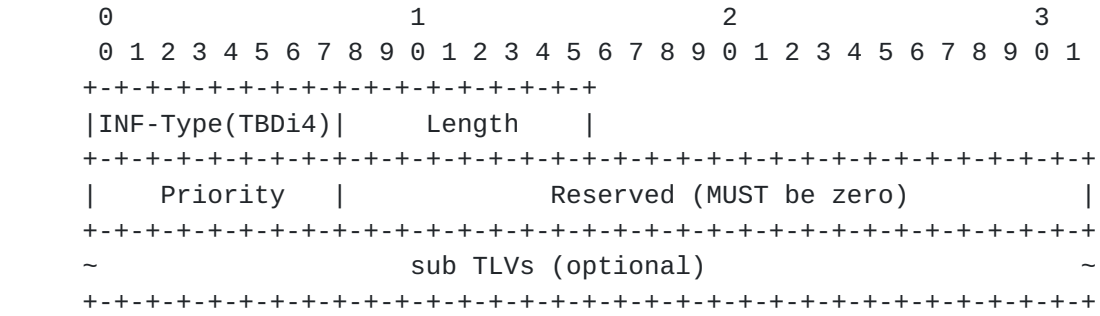
Another new TLV for the encodings of the blocks in the flooding topology is defined. It has the format below and contains the same contents as the Flooding Topology Blocks TLV defined in previous section.




ISIS Flooding Topology Blocks TLV

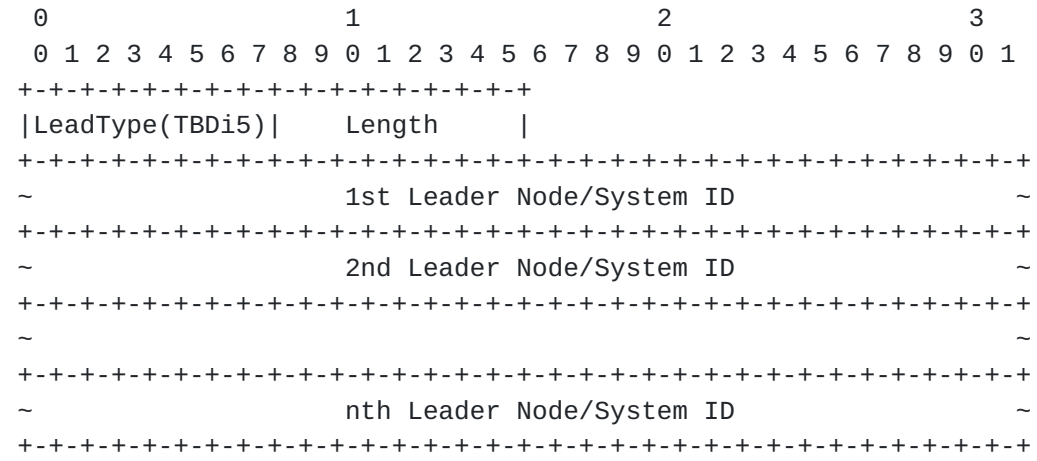
### 7.2.2. Leaders Selection

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

A sub-TLV called IS-IS leaders sub-TLV is defined. It has the following format and contains the contents similar to those in leaders sub-TLV in OSPF.




IS-IS Leaders sub-TLV

## 8. Flooding Behavior

This section describes the revised flooding behavior for a node. 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 such as link down 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 an existing node (say node E) on the flooding topology and the new node (say node N) which is not on the flooding topology. There is not any adjacency between node N and a node in the network area. The procedure for establishing the adjacency between E and N is the existing normal procedure unchanged.

When the adjacency between N and E is established, node E adds node N and the link between N and E to the flooding topology temporarily until a new flooding topology is built. New node N adds node N and

the link between N and E to the flooding topology temporarily until a new flooding topology is built.

#### 8.1.3.2. Adjacency to Existing Node

An adjacency to an existing node is an adjacency between two nodes (say nodes E and X) on the flooding topology. The procedure for establishing the adjacency between E and X is the existing normal procedure unchanged.

Both node E and node X assume that the link between E and X is not on the flooding topology until a new flooding topology is built. After the adjacency between E and X is established, node E does not send node X any new or updated LS that it receives or originates, and node X does not send node E any new or updated LS that it receives or originates until a new flooding topology is built.

### 8.2. An Exception Case

During an LS flooding, one or more 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. Multiple Failures

When two or more failures on the current flooding topology occur almost in the same time, each of the nodes within a given distance (such as 3 hops) to a failure point, floods the link state (LS) that it receives or originates to all its links (except for the one from which the LS is received) until a new flooding topology is built.

In other words, when the failures happen, each of the nodes within a given distance to a failure point, adds all its local links to the flooding topology temporarily until a new flooding topology is built.

In alternative way, each node computes and maintains a small number of backup paths. For a backup path for a link L on the flooding topology, a node N computes and maintains it only if the backup path goes through node N. Node N stores the links (e.g., local link L1 and L2) attached to it and on the backup path for link L. When link L fails and there are one or more other failures on the flooding topology or the flooding topology splits, node N adds the links (e.g., L1 and L2) to the flooding topology temporarily until a new flooding topology is built.

Similarly, for a backup path for a connection crossing a node M on the flooding topology, a node N computes and maintains it only if the backup path goes through node N. Node N stores the links (e.g., local link La and Lb) attached to it and on the backup path for node M.

When node M fails and there are one or more other failures on the flooding topology or the flooding topology splits, node N adds the links (e.g., La and Lb) to the flooding topology temporarily until a new flooding topology is built.

For one link/node failure that splits the current flooding topology, the above behavior is applied.

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.

### 8.2.2. Changes on Flooding Topology

After multiple failures split the current (old) flooding topology, some link states may be out of synchronization among some nodes. This can be resolved as follows.

After a node N computes or receives a new flooding topology, for a local link L attached to node N, if 1) link L is not on the current (old) flooding topology and is on the new flooding topology, and 2) there is a failure after the current (old) flooding topology is built, then node N sends a delta of the link states that it received or originated to its adjacent node over link L.

For node N, the delta of the link states is the link states with changes that node N received or originated during the period of time in which the current (old) flooding topology is split.

Suppose that Max_Split_Period is a number (in seconds), which is the maximum period of time in which a flooding topology is split. Tc is the time at which the current (old) flooding topology is built, Tn is the time at which the new flooding topology is built, and Ts is the bigger one between Tc and (Tn - Max_Split_Period). Node N sends its adjacent node over link L the link states with changes that it received or originated from Ts to Tn.

## 9. Operations on Flooding Reduction

### 9.1. Configuring Flooding Reduction

This section describes configurations for link state flooding reduction, including configurations for centralized flooding reduction (i.e., flooding reduction in centralized mode) and configurations for distributed flooding reduction (i.e., flooding reduction in distributed mode).

#### 9.1.1. Configurations for Centralized Flooding Reduction

At first, for each node, if it is eligible to become a leader for flooding reduction in centralized mode, a user configures a priority on the node for the leader election. The value range for the priority is from 0 to 255. A node with a priority set to zero cannot become a leader. The node with the higher priority has the higher precedence to be elected as the leader.

And then, a user selects the centralized mode on one node, which tells the other nodes in the area to use centralized flooding reduction.

#### 9.1.2. Configurations for Distributed Flooding Reduction

For distributed flooding reduction, an algorithm for computing a flooding topology needs to be configured. The algorithm and distributed mode are configured on one node, which tells the other nodes in the area the algorithm and the mode via advertising the number of the algorithm and the mode. Every node participating in the distributed flooding reduction uses this same algorithm.

### 9.2. Migration to Flooding Reduction

Migrating a OSPF or IS-IS area from normal flooding to flooding reduction smoothly takes a few steps or stages. This section describes the steps for migrating an area to centralized flooding reduction or distributed flooding reduction from normal flooding.

#### 9.2.1. Migration to Centralized Flooding Reduction

At first, a user configures a priority on every node that is eligible for the leader for centralized flooding reduction. In this stage, a node does not originate or advertise its priority.

Second, after configuring the priority, a user selects the centralized mode on one node, which tells the other nodes in the area to use centralized flooding reduction.

After a node knows that the centralized mode is used, it originates and advertises its priority. The leader election is started in the area. A user may check whether a leader is elected through showing the link state containing leaders. After the leader is elected, the centralized flooding reduction may be activated.

And then, a user activates the flooding reduction through using a configuration such as perform flooding Reduction on one node, which tells all the nodes in the area to use centralized flooding reduction. The node generates and advertises a link state with OP = R (indicating perform flooding Reduction) after it receives the configuration. After another node in the area receives the link state with OP = R, it also perform flooding reduction (i.e., floods link states using flooding topology). Thus, activating the flooding reduction on one node propagates to every node in the area, which migrates to flooding reduction.

### 9.2.2. Migration to Distributed Flooding Reduction

At first, a user selects the distributed mode on one node, which tells the other nodes in the area to use distributed flooding reduction.

After a node knows that the distributed mode is used, it advertises the algorithms it supports. A user may check whether every node advertises its supporting algorithms through showing the link state containing the algorithms.

And then, a user selects an algorithm and activates the flooding reduction through using configurations such as perform flooding Reduction on one node, which tells all the nodes in the area to use the given algorithm and start the distributed flooding reduction.

## 9.3. Roll Back to Normal Flooding

For rolling back from flooding reduction to normal flooding, a user de-activates the flooding reduction through configuring roll back to normal flooding on one node, which tells all the nodes in the area to roll back to normal flooding.

After receiving a configuration to roll back to normal flooding, the node floods link states using all its local links instead of the local links on the flooding topology. It also advertises the roll back to Normal flooding (i.e., OP = N) to all the other nodes in the area. When each of the other nodes receives the advertisement, it rolls back to normal flooding (i.e., floods link states using all its local links instead of the local links on the flooding topology).

In centralized mode, after rolling back to normal flooding, the leader of the area stops computing and advertising a flooding topology, the other nodes stops receiving and building the flooding topology. In distributed mode, every node in the area will not compute or build flooding topology.

### 9.4. Transfer from Distributed to Centralized Mode

When the distributed flooding reduction in an area is running, in order to transfer it to centralized flooding reduction, a user may take the following steps.

At first, the user rolls back from flooding reduction to normal flooding as described in section "Roll Back to Normal Flooding".

And then, the user migrates to centralized flooding reduction from normal flooding as described in section "Migration to Centralized Flooding Reduction".

Alternatively, the user may just change the flooding reduction mode from distributed mode to centralized mode on one node through a configuration. After receiving the configuration for changing the mode, the node transfers from distributed mode to centralized mode and tells the other nodes the change through advertising MOD = C (i.e., Centralized mode). After receiving the advertisement, each of the other nodes transfers from distributed mode to centralized mode.

Note that before changing the flooding reduction mode to centralized mode, the user needs to configure a priority on every node that is eligible for the leader for centralized flooding reduction.

While transferring from distributed mode to centralized mode, a node uses the distributed flooding reduction (i.e., floods the link states over its local links on the flooding topology computed and built by itself) until the centralized flooding reduction is fully functional for a given time such as 5 seconds. After this time, the node stops its distributed flooding reduction, i.e., stops computing and building its flooding topology, and using this flooding topology to flood the link states.

Each node in the area advertises its priority. A leader will be elected for the area. The leader starts to compute a flooding topology and floods it to all the other nodes. Every node builds the flooding topology computed by the leader and starts to flood the link states over its local links on this flooding topology.

### 9.5. Transfer from Centralized to Distributed Mode

When the centralized flooding reduction in an area is running, in order to transfer it to distributed flooding reduction, a user may take the following steps.

At first, the user rolls back from flooding reduction to normal flooding as described in section "Roll Back to Normal Flooding".

And then, the user migrates to distributed flooding reduction from normal flooding as described in section "Migration to Distributed Flooding Reduction".

Alternatively, the user may just change the flooding reduction mode from centralized mode to distributed mode on one node through a configuration. After receiving the configuration for changing the mode, the node transfers from centralized mode to distributed mode and tells the other nodes the change through advertising MOD = D (i.e., Distributed mode). After receiving the advertisement, each of the other nodes transfers from centralized mode to distributed mode.

While transferring from centralized mode to distributed mode, a node uses the centralized flooding reduction (i.e., floods the link states over its local links on the flooding topology computed by the leader of the area) until the distributed flooding reduction is fully functional for a given time. After this time, the node stops its centralized flooding reduction. The leader stops computing the flooding topology, advertising it to all the other routers, and using this flooding topology to flood the link states. Each of the other nodes stops receiving and building the flooding topology computed by the leader.

Every node starts to compute and build its flooding topology and to flood the link states over its local links on this flooding topology.

### 9.6. Adding a New Node to Network

If the centralized flooding reduction is used in an area, for adding a new node (say node N) to the area, a user configures a priority for this new node to become the leader of the area.

The other configurations on the new node are the existing normal ones unchanged.

When the new node N is connected via a link to a node (say E) on the flooding topology, there is not any adjacency between them (i.e., N and E) over the link. The procedure for establishing the adjacency between N and E is the existing normal procedure unchanged.

Node E adds node N and the link between N and E to the flooding topology temporarily until a new flooding topology is built.

New node N adds node N and the link between N and E to the flooding topology temporarily until a new flooding topology is built.

## 10. Manageability Considerations

Section 9 "Operations on Flooding Reduction" outlines the configuration process and deployment scenarios for link state flooding reduction. The configurable items include to set the priority of a node to become a leader of the area for link state flooding reduction in centralized mode. The flooding reduction function may be controlled by a policy module and assigned a suitable user privilege level to enable. A suitable model may be required to verify the flooding reduction status on routers participating in the flooding reduction, including their role as a leader in centralized mode or a normal node advertising link states using flooding topology. The mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those indicated in [RFC2328] and [RFC1195].

## 11. Security Considerations

A notable beneficial security aspect of link state flooding reduction is that the flooding topology in the centralized mode is advertised in a single area, and a link state is not advertised over every link, but over the links on the flooding topology. It should be noted that a malicious node could inject a fake flooding topology in the centralized mode, which could lead inconsistent link state databases among the nodes in an area. The malicious node could inject a link state with the OP field set to R or N, which could trigger the migration or roll back into/from a flooding reduction. Good security practice might reuse the IS-IS authentication in [RFC5304] as well as [RFC5310], and the OSPF authentication and other security mechanisms described in [RFC2328], [RFC4552] and [RFC7474] to mitigate this type of risk.

## 12. IANA Considerations

### 12.1. 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 as follows:

| Registry Value | Opaque Type | reference |
|---|---|---|
| 10 | FT LSA | This document |

IANA is requested to create and maintain new registries:

- 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 6.2.1.1 |
| 2 | FT Blocks TLV | see Section 6.2.1.2 |
| 3-32767 | Unassigned | |
| 32768-65535 | Reserved | |

## 12.2. OSPFv3

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

| Value | LSA Function Code Name | reference |
|---|---|---|
| 16 | FT 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 6.2.1.1 |
| 2 | FT Blocks TLV | see Section 6.2.1.2 |
| 3-32767 | Unassigned | |
| 32768-65535 | Reserved | |

### 12.3. 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 | FT Blocks TLV | see Section 7.2.1 |
| 153 | Instruction TLV | see Section 7.1 |
| 154 | Information TLV | see Section 7.2.2 |

## 13. 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.

## 14. References

### 14.1. Normative References

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <https://www.rfc-editor.org/info/rfc1195>.

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

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <https://www.rfc-editor.org/info/rfc4552>.

[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>.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <https://www.rfc-editor.org/info/rfc5304>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <https://www.rfc-editor.org/info/rfc5310>.

[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>.

[RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <https://www.rfc-editor.org/info/rfc7474>.

[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>.

## 14.2. 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-07 (work in progress), October 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
Boston
USA

Email: huaimo.chen@huawei.com

Dean Cheng
Huawei Technologies
Santa Clara
USA

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

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

Email: wangaj.bri@chinatelecom.cn

Xufeng Liu
Volta Networks
McLean, VA
USA

Email: xufeng.liu.ietf@gmail.com

Yanhe Fan
Casa Systems
USA

Email: yfan@casa-systems.com

Lei Liu
USA

Email: liulei.kddi@gmail.com