Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 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

March 10, 2019

LS Distributed Flooding Reduction
draft-cc-lsr-flooding-reduction-02

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. In this approach, every node in the area automatically calculates a flooding topology by using a same algorithm concurrently.

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/.

Copyright Notice

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.

In this approach, every node in the network automatically calculates a flooding topology by using a same algorithm concurrently at almost the same time. It floods a link state on the flooding topology. There may be multiple algorithms for computing a flooding topology. Users can select one they prefer, and smoothly switch from one to another. The approach is applicable to any network topology in a single area. It is backward compatible.

2. Terminology

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.

FTC: Flooding Topology Computation.

3. 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.

3.1. Flooding Topology Construction

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 three different ways to construct a flooding topology for a given network topology: centralized, distributed and static mode. This document focuses on the distributed mode, in which each node in the network automatically calculates a flooding topology by using a same algorithm concurrently at almost the same time.

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.

3.2. Scheduling for Flooding Topology Computation

In a network consisting of routers from multiple vendors, there are different schedulers for SPF. Using different schedulers is in favor of creating more micro routing loops during the convergence process due to discrepancies of schedulers than using a same scheduler. More micro routing loops will lead to more traffic lose. Service providers are already aware to use similar timers (values and behavior), but sometimes it is not possible due to limitations of schedulers [I-D.ietf-rtgwg-spf-uloop-pb-statement]. In order to let every node run a flooding topology computation (FTC) at almost the same time, we need to have a same scheduler for FTC to be implemented by multiple vendors.

Two schedulers are described below. One uses a constant delay such as 200ms for each of multiple consecutive FTCs. For example, for four consecutive FTCs, the second FTC will be triggered 200ms after the first FTC; the third FTC will be triggered 200ms after the second FTC; and the forth FTC will be triggered 200ms after the third FTC.

The other uses an exponential delay starting from a given hold time such as 100ms for consecutive FTCs. For example, for four consecutive FTCs, the second FTC will be triggered 2 x 100ms = 200ms after the first FTC; the third FTC will be triggered 2 x 200ms = 400ms after the second FTC; and the forth FTC will be triggered 2 x 400ms = 800ms after the second FTC.

If these two schedulers are used in a network, it is almost impossible to let every node in the network run FTC at almost same time for multiple consecutive FTCs.

3.2.1. Scheduler with Exponential Delay

There are three parameters for the scheduler: initial-delay, minimum-hold-time and maximum-wait-time. The initial-delay is the delay in milliseconds from detecting a topology change to triggering FTC. Its default value is 50ms.

The minimum-hold-time is the minimum hold time in milliseconds between two consecutive FTCs. Its default value is 100ms.

The maximum-wait-time is the maximum wait time in milliseconds for triggering FTC. Its default value is 2000ms.

The behavior of the scheduler with these parameters is described as follows:

1. When FTC is to be called first time, initial-delay for FTC.
2. When FTC is to be called n-th (n > 1) time consecutively, delay T = minimum-hold-time x 2^(n-2) for FTC if T is less than maximum-wait-time
3. When T = hold-time x 2^(n-2) reaches maximum-wait-time, delay maximum-wait-time for FTC. Then repeats step 1 (i.e., the next FTC call is considered as first time again).

Scheduler with Exponential Delay

3.2.2. Scheduler with Constant Delay

There are three parameters for the scheduler: constant-delay, number-of-runs and maximum-wait-time. The constant-delay is the constant time to delay in milliseconds from detecting a topology change to triggering FTC. Its default value is 200ms.

The number-of-runs is the maximum number of times that FTC can run consecutively. Its default value is 5.

The maximum-wait-time is the maximum wait time in milliseconds for triggering FTC. Its default value is 2000ms.

The behavior of the scheduler with these parameters is described as follows:

1. When FTC is to be called first time, constant-delay for FTC.
2. When FTC is to be called n-th (n > 1) time consecutively,
   delay constant-delay for FTC if n <= number-of-runs
3. If n == number-of-runs,
   delay maximum-wait-time, and then repeats step 1
   (i.e., the next FTC call is considered as first time again).

Scheduler with Constant Delay

3.3. Flooding Topology Consistency

The flooding topology computed by one node needs to be the same as the one computed by another node. When two flooding topologies computed by two nodes are different, this inconsistency needs to be detected and handled accordingly.

3.4. Flooding Topology Protection

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. 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 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 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 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.

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.

4. Protocol Extensions

The extensions comprises two parts: one part is for operations on flooding reduction, the other is for flooding topology consistency.

4.1. Extensions for Operations

4.1.1. Extensions to OSPF

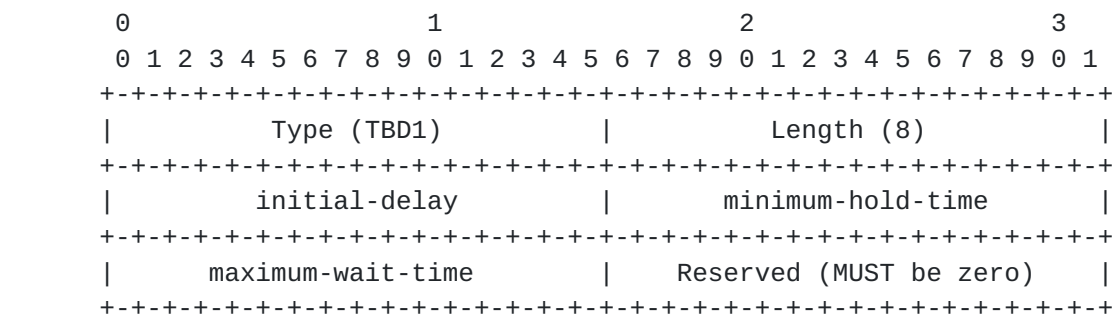
The OSPF Dynamic Flooding sub-TLV and area leader sub-TLV are defined in [I-D.ietf-lsr-dynamic-flooding]. The former may contains a number of algorithms. The latter contains instructions about flooding reduction.

Every node supporting the distributed flooding reduction MUST indicate its algorithms for flooding topology computation in a OSPF Dynamic Flooding sub-TLV. This sub-TLV in a RI LSA will be advertised to the area leader.

When the distributed flooding reduction is selected, every node MUST receive the OSPF area leader sub-TLV in a RI LSA from the area leader, which indicates the distributed mode and an algorithm to be used. It SHOULD receive the parameters needed for the algorithm and the distributed mode.

The parameters for the distributed mode include those three parameters configured for the scheduler for flooding topology computation. Through configuring these parameters on one place such as the area leader and automatically advertising them to every node in the network, we simplify the operation on flooding reduction and reduce the errors on configurations (comparing to manually configuring these parameters on every node).

A new sub-TLV, called OSPF Scheduler Parameters sub-TLV, is defined for advertising the three parameters configured for the scheduler. Its format is illustrated below.



OSPF Scheduler Parameters sub-TLV

Type: TBD1 for Scheduler Parameters is to be assigned by IANA.

Length: 8.

initial-delay: 2 octets' field representing the initial delay in milliseconds.

minimum-hold-time: 2 octets' field representing the minimum hold time in milliseconds.

maximum-wait-time: 2 octets' field representing the maximum wait time in milliseconds.

Reserved: MUST be set to 0 while sending and ignored on receipt.

In the case where the distributed flooding reduction is selected and an algorithm for flooding topology computation is given already, there are some operational changes. These changes include:

1 the algorithm given is changed to another algorithm;

2 the distributed flooding reduction is rolled back to normal flooding; and

3 the distributed flooding reduction is changed to centralized flooding reduction.

For the first change, every node MUST receive the OSPF area leader sub-TLV in a RI LSA from the leader, which indicates that another algorithm is to be used. After receiving the sub-TLV, it uses the new algorithm to compute a new flooding topology, and floods link states over both the flooding topology computed by the old algorithm and the new flooding topology for a given time. And then it will floods link states over the flooding topology computed by the new algorithm.

For the second change, every node MUST receive the OSPF area leader sub-TLV in a RI LSA from the leader, which indicates the current flooding reduction is to be rolled back to normal flooding. After receiving the sub-TLV, it stops computing flooding topology and flooding link states over a flooding topology. It floods link states using all its local links instead of the local links on the flooding topology.

Note that the OSPF area leader sub-TLV defined in [I-D.ietf-lsr-dynamic-flooding] needs to be extended to allow users to roll back to normal flooding. The Flooding Reduction Instruction sub-TLV defined in the previous version of this draft supports this.

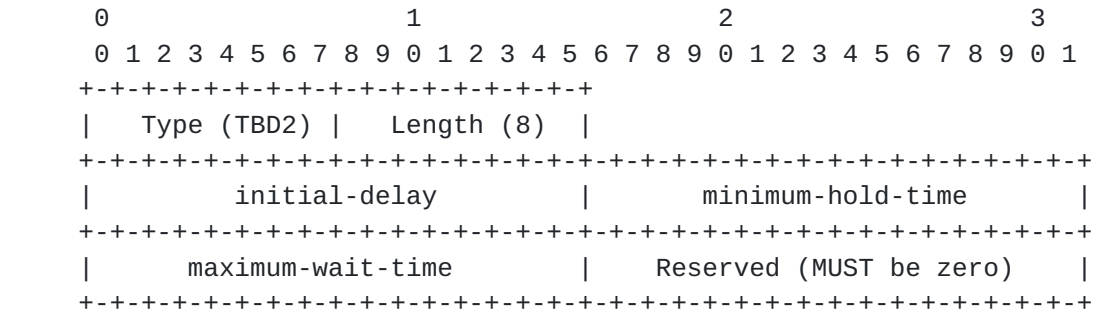
4.1.2. Extensions to IS-IS

Similar to OSPF, the IS-IS Dynamic Flooding sub-TLV and area leader sub-TLV are also defined in [I-D.ietf-lsr-dynamic-flooding].

Every node supporting the distributed flooding reduction MUST indicate its algorithms for flooding topology computation in an IS-IS Dynamic Flooding sub-TLV in an LSP to be advertised to the leader.

When the distributed flooding reduction is selected, every node MUST receive the IS-IS area leader sub-TLV in an LSP, which indicates the distributed mode and an algorithm to be used. It SHOULD receive the parameters needed for the algorithm and the distributed mode.

A new sub-TLV, called IS-IS Scheduler Parameters sub-TLV, is defined for advertising the three parameters configured for the scheduler. Its format is illustrated below.



IS-IS Scheduler Parameters sub-TLV

Type: TBD1 for Scheduler Parameters is to be assigned by IANA.

Length: 8.

initial-delay: 2 octets' field representing the initial delay in milliseconds.

minimum-hold-time: 2 octets' field representing the minimum hold time in milliseconds.

maximum-wait-time: 2 octets' field representing the maximum wait time in milliseconds.

Reserved: MUST be set to 0 while sending and ignored on receipt.

4.2. Extensions for Consistency

4.2.1. Extensions to OSPF

RFC 5613 defines a TLV called Extended Options and Flag (EOF) TLV. A OSPF Hello may contain this TLV in link-local signaling (LLS) data block. A new flag bit (bit 30 suggested), called link on flooding topology (FT-bit for short), is defined in EOF TLV.

When a node B receives a Hello from its adjacent node A over a link, FT-bit set to one in the Hello indicates that the link is on the flooding topology (FT) from node A's point of view.

For a link between node A and node B, not on the current FT, after node A computes a new FT and the link is on the new FT, it sends a Hello with FT-bit set to one to node B. Similarly, after node B computes a new FT and the link is on the new FT, it sends a Hello with FT-bit set to one to node A. Note that Hello may include FT-bit after the state of the adjacency between A and B is FULL.

For a link between node A and node B, on the current FT, after node A computes a new FT and the link is not on the new FT, it sends a Hello with FT-bit set to zero to node B. Similarly, after node B computes a new FT and the link is not on the new FT, it sends a Hello with FT-bit set to zero to node A.

If the Hellos from the two nodes have the same FT-bit value, then the FT for the link between the two nodes is consistent; otherwise, it is not consistent.

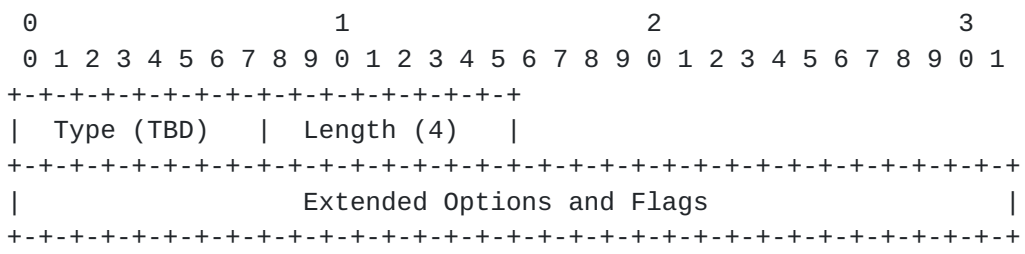
If one of the two nodes receives the Hellos with FT-bit set to one from the other, but sends the Hellos with FT-bit set to zero for a number of Hellos such as 5 Hellos or a given time, then the FT for the link between the two nodes is not consistent.

When an inconsistency on the FT for a link is detected, a warning is issued or logged, and the node receiving the Hellos with FT-bit set

to one from the other node assumes that the link is on the FT temporarily and floods the link states over the link.

4.2.2. Extensions to IS-IS

A new TLV, called Extended Options and Flag (EOF) TLV, is defined. It may be included in an IS-IS Hello. Its format is shown below.



EOF-TLV in IS-IS Hello

Similar to OSPF, a new flag bit (bit 31 suggested), called link on flooding topology (FT-bit for short), is defined in EOF TLV. When a node B receives a Hello from its adjacent node A over a link, FT-bit set to one in the Hello indicates that the link is on the FT from node A's point of view.

5. 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.

5.1. Nodes Perform Flooding Reduction without Failure

5.1.1. 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.

5.1.2. 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.

5.1.3. Establishing Adjacencies

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

5.1.3.1. 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.

5.1.3.2. 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.

5.2. An Exception Case

5.2.1. Multiple Failures

When a node detects that two or more failures on the current flooding topology occur before a new flooding topology is built, it enables one flooding topology protection method in section 3.4.

5.2.2. Changes on Flooding Topology

After one or more 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 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.

6. Operations on Flooding Reduction

6.1. Configuring Flooding Reduction

This section describes configurations for distributed flooding reduction (i.e., flooding reduction in distributed mode).

6.1.1. 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 a node such as the area leader, 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.

6.2. 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 distributed flooding reduction from normal flooding.

6.2.1. Migration to Distributed Flooding Reduction

At first, a user selects the distributed mode on a node such as the area leader 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, which tells all the nodes in the area to use the given algorithm and start the distributed flooding reduction.

6.3. 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 a 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 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 distributed mode, every node in the area will not compute or build flooding topology.

7. Manageability Considerations

Section 6 "Operations on Flooding Reduction" outlines the configuration process and deployment scenarios for link state flooding reduction. 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 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].

8. Security Considerations

A notable beneficial security aspect of link state flooding reduction is that a link state is not advertised over every link, but over the links on the flooding topology. The malicious node could inject a link state with a different algorithm, which could trigger the flooding topology computation using the algorithm. 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.

9. IANA Considerations

9.1. OSPF

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

TLV Value	TLV Name	reference
11	Scheduler Parameters TLV	This document

9.2. IS-IS

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

TLV Value	TLV Name	reference
151	Scheduler Parameters TLV	This document
152	Extended Options and Flag TLV	This document

10. 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.

11. References

11.1. Normative References

[I-D.ietf-lsr-dynamic-flooding]
Li, T., Psenak, P., Ginsberg, L., Przygienda, T., Cooper, D., Jalil, L., and S. Dontula, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-00 (work in progress), February 2019.

[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>.

11.2. Informative References

[I-D.ietf-rtgwg-spf-uloop-pb-statement]
Litkowski, S., Decraene, B., and M. Horneffer, "Link State protocols SPF trigger and delay algorithm impact on IGP micro-loops", draft-ietf-rtgwg-spf-uloop-pb-statement-10 (work in progress), January 2019.

[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