

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: August 29, 2019

T. Li, Ed.  
Arista Networks  
P. Psenak, Ed.  
L. Ginsberg  
Cisco Systems, Inc.  
T. Przygienda  
Juniper Networks, Inc.  
D. Cooper  
CenturyLink  
L. Jalil  
Verizon  
S. Dontula  
ATT  
February 25, 2019

**Dynamic Flooding on Dense Graphs**  
**draft-ietf-lsr-dynamic-flooding-00**

Abstract

Routing with link state protocols in dense network topologies can result in sub-optimal convergence times due to the overhead associated with flooding. This can be addressed by decreasing the flooding topology so that it is less dense.

This document discusses the problem in some depth and an architectural solution. Specific protocol changes for IS-IS, OSPFv2, and OSPFv3 are described in this document.

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 August 29, 2019.

## Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">1.1.</a>	Requirements Language . . . . .	<a href="#">4</a>
<a href="#">2.</a>	Problem Statement . . . . .	<a href="#">5</a>
<a href="#">3.</a>	Solution Requirements . . . . .	<a href="#">5</a>
<a href="#">4.</a>	Dynamic Flooding . . . . .	<a href="#">5</a>
<a href="#">4.1.</a>	Applicability . . . . .	<a href="#">7</a>
<a href="#">4.2.</a>	Leader election . . . . .	<a href="#">7</a>
<a href="#">4.3.</a>	Computing the Flooding Topology . . . . .	<a href="#">8</a>
<a href="#">4.4.</a>	Topologies on Complete Bipartite Graphs . . . . .	<a href="#">9</a>
<a href="#">4.4.1.</a>	A Minimal Flooding Topology . . . . .	<a href="#">9</a>
<a href="#">4.4.2.</a>	Xia Topologies . . . . .	<a href="#">9</a>
<a href="#">4.4.3.</a>	Optimization . . . . .	<a href="#">10</a>
<a href="#">4.5.</a>	Encoding the Flooding Topology . . . . .	<a href="#">11</a>
<a href="#">5.</a>	Protocol Elements . . . . .	<a href="#">11</a>
<a href="#">5.1.</a>	IS-IS TLVs . . . . .	<a href="#">11</a>
<a href="#">5.1.1.</a>	IS-IS Area Leader Sub-TLV . . . . .	<a href="#">11</a>
<a href="#">5.1.2.</a>	IS-IS Dynamic Flooding Sub-TLV . . . . .	<a href="#">12</a>
<a href="#">5.1.3.</a>	IS-IS Area System IDs TLV . . . . .	<a href="#">13</a>
<a href="#">5.1.4.</a>	IS-IS Flooding Path TLV . . . . .	<a href="#">14</a>
<a href="#">5.1.5.</a>	IS-IS Flooding Request TLV . . . . .	<a href="#">15</a>
<a href="#">5.2.</a>	OSPF LSAs and TLVs . . . . .	<a href="#">16</a>
<a href="#">5.2.1.</a>	OSPF Area Leader Sub-TLV . . . . .	<a href="#">17</a>
<a href="#">5.2.2.</a>	OSPF Dynamic Flooding Sub-TLV . . . . .	<a href="#">17</a>
<a href="#">5.2.3.</a>	OSPFv2 Dynamic Flooding Opaque LSA . . . . .	<a href="#">18</a>
<a href="#">5.2.4.</a>	OSPFv3 Dynamic Flooding LSA . . . . .	<a href="#">19</a>
<a href="#">5.2.5.</a>	OSPF Area Router IDs TLV . . . . .	<a href="#">20</a>
<a href="#">5.2.6.</a>	OSPF Flooding Path TLV . . . . .	<a href="#">21</a>
<a href="#">5.2.7.</a>	OSPF Flooding Request Bit . . . . .	<a href="#">22</a>
<a href="#">6.</a>	Behavioral Specification . . . . .	<a href="#">23</a>
<a href="#">6.1.</a>	Terminology . . . . .	<a href="#">23</a>
<a href="#">6.2.</a>	Flooding Topology . . . . .	<a href="#">23</a>



<a href="#">6.3.</a>	<a href="#">Leader Election . . . . .</a>	<a href="#">24</a>
<a href="#">6.4.</a>	<a href="#">Area Leader Responsibilities . . . . .</a>	<a href="#">24</a>
<a href="#">6.5.</a>	<a href="#">Distributed Flooding Topology Calculation . . . . .</a>	<a href="#">24</a>
<a href="#">6.6.</a>	<a href="#">Flooding Behavior . . . . .</a>	<a href="#">25</a>
<a href="#">6.7.</a>	<a href="#">Treatment of Topology Events . . . . .</a>	<a href="#">25</a>
<a href="#">6.7.1.</a>	<a href="#">Temporary Addition of Link to Flooding Topology . . . . .</a>	<a href="#">26</a>
<a href="#">6.7.2.</a>	<a href="#">Local Link Addition . . . . .</a>	<a href="#">26</a>
<a href="#">6.7.3.</a>	<a href="#">Node Addition . . . . .</a>	<a href="#">27</a>
<a href="#">6.7.4.</a>	<a href="#">Failures of Link Not on Flooding Topology . . . . .</a>	<a href="#">27</a>
<a href="#">6.7.5.</a>	<a href="#">Failures of Link On the Flooding Topology . . . . .</a>	<a href="#">28</a>
<a href="#">6.7.6.</a>	<a href="#">Node Deletion . . . . .</a>	<a href="#">28</a>
<a href="#">6.7.7.</a>	<a href="#">Local Link Addition to the Flooding Topology . . . . .</a>	<a href="#">28</a>
<a href="#">6.7.8.</a>	<a href="#">Local Link Deletion from the Flooding Topology . . . . .</a>	<a href="#">29</a>
<a href="#">6.7.9.</a>	<a href="#">Treatment of Disconnected Adjacent Nodes . . . . .</a>	<a href="#">29</a>
<a href="#">6.7.10.</a>	<a href="#">Failure of the Area Leader . . . . .</a>	<a href="#">29</a>
<a href="#">6.7.11.</a>	<a href="#">Recovery from Multiple Failures . . . . .</a>	<a href="#">30</a>
<a href="#">7.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">30</a>
<a href="#">7.1.</a>	<a href="#">IS-IS . . . . .</a>	<a href="#">30</a>
<a href="#">7.2.</a>	<a href="#">OSPF . . . . .</a>	<a href="#">31</a>
<a href="#">7.2.1.</a>	<a href="#">OSPF Dynamic Flooding LSA TLVs Registry . . . . .</a>	<a href="#">32</a>
<a href="#">7.3.</a>	<a href="#">IGP . . . . .</a>	<a href="#">33</a>
<a href="#">8.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">33</a>
<a href="#">9.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">33</a>
<a href="#">10.</a>	<a href="#">References . . . . .</a>	<a href="#">34</a>
<a href="#">10.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">34</a>
<a href="#">10.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">35</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">36</a>

## [1.](#) Introduction

In recent years, there has been increased focus on how to address the dynamic routing of networks that have a bipartite (a.k.a. spine-leaf or leaf-spine), Clos [[Clos](#)], or Fat Tree [[Leiserson](#)] topology. Conventional Interior Gateway Protocols (IGPs, i.e., IS-IS [[ISO10589](#)], OSPFv2 [[RFC2328](#)], and OSPFv3 [[RFC5340](#)]) under-perform, redundantly flooding information throughout the dense topology, leading to overloaded control plane inputs and thereby creating operational issues. For practical considerations, network architects have resorted to applying unconventional techniques to address the problem, e.g., applying BGP in the data center [[RFC7938](#)]. However it is very clear that using an Exterior Gateway Protocol as an IGP is sub-optimal, if only due to the configuration overhead.

The primary issue that is demonstrated when conventional mechanisms are applied is the poor reaction of the network to topology changes. Normal link state routing protocols rely on a flooding algorithm for state distribution within an area. In a dense topology, this flooding algorithm is highly redundant, resulting in unnecessary



overhead. Each node in the topology receives each link state update multiple times. Ultimately, all of the redundant copies will be discarded, but only after they have reached the control plane and been processed. This creates issues because significant link state database updates can become queued behind many redundant copies of another update. This delays convergence as the link state database does not stabilize promptly.

In a real world implementation, the packet queues leading to the control plane are necessarily of finite size, so if the flooding rate exceeds the update processing rate for long enough, the control plane will be obligated to drop incoming updates. If these lost updates are of significance, this will further delay stabilization of the link state database and the convergence of the network.

This is not a new problem. Historically, when routing protocols have been deployed in networks where the underlying topology is a complete graph, there have been similar issues. This was more common when the underlying link layer fabric presented the network layer with a full mesh of virtual connections. This was addressed by reducing the flooding topology through IS-IS Mesh Groups [[RFC2973](#)], but this approach requires careful configuration of the flooding topology.

Thus, the root problem is not limited to massively scalable data centers. It exists with any dense topology at scale.

This problem is not entirely surprising. Link state routing protocols were conceived when links were very expensive and topologies were sparse. The fact that those same designs are sub-optimal in a dense topology should not come as a huge surprise. The fundamental premise that was addressed by the original designs was an environment of extreme cost and scarcity. Technology has progressed to the point where links are cheap and common. This represents a complete reversal in the economic fundamentals of network engineering. The original designs are to be commended for continuing to provide correct operation to this point, and optimizations for operation in today's environment are to be expected.

### **1.1. Requirements Language**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].



## **2. Problem Statement**

In a dense topology, the flooding algorithm that is the heart of conventional link state routing protocols causes a great deal of redundant messaging. This is exacerbated by scale. While the protocol can survive this combination, the redundant messaging is unnecessary overhead and delays convergence. Thus, the problem is to provide routing in dense, scalable topologies with rapid convergence.

## **3. Solution Requirements**

A solution to this problem must then meet the following requirements:

Requirement 1     Provide a dynamic routing solution. Reachability must be restored after any topology change.

Requirement 2     Provide a significant improvement in convergence.

Requirement 3     The solution should address a variety of dense topologies. Just addressing a complete bipartite topology such as K5,8 is insufficient. Multi-stage Clos topologies must also be addressed, as well as topologies that are slight variants. Addressing complete graphs is a good demonstration of generality.

Requirement 4     There must be no single point of failure. The loss of any link or node should not unduly hinder convergence.

Requirement 5     Dense topologies are subgraphs of much larger topologies. Operational efficiency requires that the dense subgraph not operate in a radically different manner than the remainder of the topology. While some operational differences are permissible, they should be minimized. Changes to nodes outside of the dense subgraph are not acceptable. These situations occur when massively scaled data centers are part of an overall larger wide-area network. Having a second protocol operating just on this subgraph would add much more complexity at the edge of the subgraph where the two protocols would have to inter-operate.

## **4. Dynamic Flooding**

We have observed that the combination of the dense topology and flooding on the physical topology in a scalable network is sub-optimal. However, if we decouple the flooding topology from the physical topology and only flood on a greatly reduced portion of that topology, we can have efficient flooding and retain all of the resilience of existing protocols. A node that supports flooding on the decoupled flooding topology is said to support dynamic flooding.





In this idea, the flooding topology is computed within an IGP area with the dense topology either centrally on an elected node, termed the Area Leader, or in a distributed manner on all nodes that are supporting Dynamic Flooding. If the flooding topology is computed centrally, it is encoded into and distributed as part of the normal link state database. We call this the centralized mode of operation. If the flooding topology is computed in a distributed fashion, we call this the distributed mode of operation. Nodes within such an IGP area would only flood on the flooding topology. On links outside of the normal flooding topology, normal database synchronization mechanisms (i.e., OSPF database exchange, IS-IS CSNPs) would apply, but flooding may not. Details are described in [Section 6](#). New link state information that arrives from outside of the flooding topology suggests that the sender has a different or no flooding topology information and that the link state update should be flooded on the flooding topology as well.

The flooding topology covers the full set of nodes within the area, but excludes some of the links that standard flooding would employ.

Since the flooding topology is computed prior to topology changes, it does not factor into the convergence time and can be done when the topology is stable. The speed of the computation and its distribution, in the case of a centralized mode, is not a significant issue.

If a node does not have any flooding topology information when it receives new link state information, it should flood according to standard flooding rules. This situation will occur when the dense topology is first established, but is unlikely to recur.

When centralized mode is used and if, during a transient, there are multiple flooding topologies being advertised, then nodes should flood link state updates on all of the flooding topologies. Each node should locally evaluate the election of the Area Leader for the IGP area and first flood on its flooding topology. The rationale behind this is straightforward: if there is a transient and there has been a recent change in Area Leader, then propagating topology information promptly along the most likely flooding topology should be the priority.

During transients, it is possible that loops will form in the flooding topology. This is not problematic, as the legacy flooding rules would cause duplicate updates to be ignored. Similarly, during transients, it is possible that the flooding topology may become disconnected. [Section 6.7.11](#) discusses how such conditions are handled.



#### **4.1. Applicability**

In a complete graph, this approach is appealing because it drastically decreases the flooding topology without the manual configuration of mesh groups. By controlling the diameter of the flooding topology, as well as the maximum degree node in the flooding topology, convergence time goals can be met and the stability of the control plane can be assured.

Similarly, in a massively scaled data center, where there are many opportunities for redundant flooding, this mechanism ensures that flooding is redundant, with each leaf and spine well connected, while ensuring that no update need make too many hops and that no node shares an undue portion of the flooding effort.

In a network where only a portion of the nodes support Dynamic Flooding, the remaining nodes will continue to perform standard flooding. This is not an issue for correctness, as no node can become isolated.

Flooding that is initiated by nodes that support Dynamic Flooding will remain within the flooding topology until it reaches a legacy node, which will resume legacy flooding. Standard flooding will be bounded by nodes supporting Dynamic Flooding, which can help limit the propagation of unnecessary flooding. Whether or not the network can remain stable in this condition is unknown and may be very dependent on the number and location of the nodes that support Dynamic Flooding.

During incremental deployment of dynamic flooding an area will consist of one or more sets of connected nodes that support dynamic flooding and one or more sets of connected nodes that do not, i.e., nodes that support standard flooding. The flooding topology is the union of these sets of nodes. Each set of nodes that does not support dynamic flooding needs to be part of the flooding topology and such a set of nodes may provide connectivity between two or more sets of nodes that support dynamic flooding.

#### **4.2. Leader election**

A single node within the dense topology is elected as an Area Leader.

A generalization of the mechanisms used in existing Designated Router (OSPF) or Designated Intermediate-System (IS-IS) elections suffices. The elected node is known as the Area Leader.

In the case of centralized mode, the Area Leader is responsible for computing and distributing the flooding topology. When a new Area



Leader is elected and has distributed new flooding topology information, then any prior Area Leaders should withdraw any of their flooding topology information from their link state database entries.

In the case of distributed mode, the distributed algorithm advertised by the Area Leader **MUST** be used by all nodes that participate in Dynamic Flooding.

Not every node needs to be a candidate to be Area Leader within an area, as a single candidate is sufficient for correct operation. For redundancy, however, it is strongly **RECOMMENDED** that there be multiple candidates.

#### **4.3. Computing the Flooding Topology**

There is a great deal of flexibility in how the flooding topology may be computed. For resilience, it needs to at least contain a cycle of all nodes in the dense subgraph. However, additional links could be added to decrease the convergence time. The trade-off between the density of the flooding topology and the convergence time is a matter for further study. The exact algorithm for computing the flooding topology in the case of the centralized computation need not be standardized, as it is not an interoperability issue. Only the encoding of the result needs to be documented. In the case of distributed mode, all nodes in the IGP area need to use the same algorithm to compute the flooding topology. It is possible to use private algorithms to compute flooding topology, so long as all nodes in the IGP area use the same algorithm.

While the flooding topology should be a covering cycle, it need not be a Hamiltonian cycle where each node appears only once. In fact, in many relevant topologies this will not be possible e.g., K5,8. This is fortunate, as computing a Hamiltonian cycle is known to be NP-complete.

A simple algorithm to compute the topology for a complete bipartite graph is to simply select unvisited nodes on each side of the graph until both sides are completely visited. If the number of nodes on each side of the graph are unequal, then revisiting nodes on the less populated side of the graph will be inevitable. This algorithm can run in  $O(N)$  time, so is quite efficient.

While a simple cycle is adequate for correctness and resiliency, it may not be optimal for convergence. At scale, a cycle may have a diameter that is half the number of nodes in the graph. This could cause an undue delay in link state update propagation. Therefore it may be useful to have a bound on the diameter of the flooding topology. Introducing more links into the flooding topology would



reduce the diameter, but at the trade-off of possibly adding redundant messaging. The optimal trade-off between convergence time and graph diameter is for further study.

Similarly, if additional redundancy is added to the flooding topology, specific nodes in that topology may end up with a very high degree. This could result in overloading the control plane of those nodes, resulting in poor convergence. Thus, it may be optimal to have an upper bound on the degree of nodes in the flooding topology. Again, the optimal trade-off between graph diameter, node degree, and convergence time, and topology computation time is for further study.

If the leader chooses to include a multi-node broadcast LAN segment as part of the flooding topology, all of the connectivity to that LAN segment should be included as well. Once updates are flooded onto the LAN, they will be received by every attached node.

#### **4.4. Topologies on Complete Bipartite Graphs**

Complete bipartite graph topologies have become popular for data center applications and are commonly called leaf-spine or spine-leaf topologies. In this section, we discuss some flooding topologies that are of particular interest in these networks.

##### **4.4.1. A Minimal Flooding Topology**

We define a Minimal Flooding Topology on a complete bipartite graph as one in which the topology is connected and each node has at least degree two. This is of interest because it guarantees that the flooding topology has no single points of failure.

In practice, this implies that every leaf node in the flooding topology will have a degree of two. As there are usually more leaves than spines, the degree of the spines will be higher, but the load on the individual spines can be evenly distributed.

This type of flooding topology is also of interest because it scales well. As the number of leaves increases, we can construct flooding topologies that perform well. Specifically, for  $n$  spines and  $m$  leaves, if  $m \geq n(n/2-1)$ , then there is a flooding topology that has a diameter of four.

##### **4.4.2. Xia Topologies**

We define a Xia Topology on a complete bipartite graph as one in which all spine nodes are bi-connected through leaves with degree two, but the remaining leaves all have degree one and are evenly distributed across the spines.





Constructively, we can create a Xia topology by iterating through the spines. Each spine can be connected to the next spine by selecting any unused leaf. Since leaves are connected to all spines, all leaves will have a connection to both the first and second spine and we can therefore choose any leaf without loss of generality. Continuing this iteration across all of the spines, selecting a new leaf at each iteration, will result in a path that connects all spines. Adding one more leaf between the last and first spine will produce a cycle of  $n$  spines and  $n$  leaves.

At this point,  $m-n$  leaves remain unconnected. These can be distributed evenly across the remaining spines, connected by a single link.

Xia topologies represent a compromise that trades off increased risk and decreased performance for lower flooding amplification. Xia topologies will have a larger diameter. For  $m$  spines, the diameter will be  $m + 2$ .

In a Xia topology, some leaves are singly connected. This represents a risk in that in some failures, convergence may be delayed. However, there may be some alternate behaviors that can be employed to mitigate these risks. If a leaf node sees that its single link on the flooding topology has failed, it can compensate by performing a database synchronization check with a different spine. Similarly, if a leaf determines that its connected spine on the flooding topology has failed, it can compensate by performing a database synchronization check with a different spine. In both of these cases, the synchronization check is intended to ameliorate any delays in link state propagation due to the fragmentation of the flooding topology.

The benefit of this topology is that flooding load is easily understood. Each node in the spine cycle will never receive an update more than twice. For  $m$  leaves and  $n$  spines, a spine never transmits more than  $(m/n + 1)$  updates.

#### **4.4.3. Optimization**

If two nodes are adjacent on the flooding topology and there are a set of parallel links between them, then any given update MUST be flooded over a single one of those links. Selection of the specific link is implementation specific.



#### **4.5. Encoding the Flooding Topology**

There are a variety of ways that the flooding topology could be encoded efficiently. If the topology was only a cycle, a simple list of the nodes in the topology would suffice. However, this is insufficiently flexible as it would require a slightly different encoding scheme as soon as a single additional link is added. Instead, we choose to encode the flooding topology as a set of intersecting paths, where each path is a set of connected edges.

Other encodings are certainly possible. We have attempted to make a useful trade off between simplicity, generality, and space.

### **5. Protocol Elements**

#### **5.1. IS-IS TLVs**

The following TLVs/sub-TLVs are added to IS-IS:

1. A sub-TLV that an IS may inject into its LSP to indicate its preference for becoming Area Leader.
2. A sub-TLV that an IS may inject into its LSP to indicate that it supports Dynamic Flooding and the algorithms that it supports for distributed mode, if any.
3. A TLV to carry the list of system IDs that compromise the flooding topology for the area.
4. A TLV to carry a path which is part of the flooding topology
5. A TLV that requests flooding from the adjacent node

##### **5.1.1. IS-IS Area Leader Sub-TLV**

The Area Leader Sub-TLV allows a system to:

1. Indicate its eligibility and priority for becoming Area Leader.
2. Indicate whether centralized or distributed mode is to be used to compute the flooding topology in the area.
3. Indicate the algorithm identifier for the algorithm that is used to compute the flooding topology in distributed mode.

Intermediate Systems (nodes) that are not advertising this Sub-TLV are not eligible to become Area Leader.



The Area Leader is the node with the numerically highest Area Leader priority in the area. In the event of ties, the node with the numerically highest system ID is the Area Leader. Due to transients during database flooding, different nodes may not agree on the Area Leader.

The Area Leader Sub-TLV is advertised as a Sub-TLV of the IS-IS Router Capability TLV-242 that is defined in [\[RFC7981\]](#) and has the following format:

```

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      | Priority      | Algorithm  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD1

Length: 2

Priority: 0-255, unsigned integer

Algorithm: a numeric identifier in the range 0-255 that identifies the algorithm used to calculate the flooding topology. The following values are defined:

0: Centralized computation by the Area Leader.

1-127: Standardized distributed algorithms. Individual values are to be assigned according to the "Specification Required" policy defined in [\[RFC8126\]](#) (see [Section 7.3](#)).

128-254: Private distributed algorithms. Individual values are to be assigned according to the "Private Use" policy defined in [\[RFC8126\]](#) (see [Section 7.3](#)).

255: Reserved

#### **[5.1.2](#). IS-IS Dynamic Flooding Sub-TLV**

The Dynamic Flooding Sub-TLV allows a system to:

1. Indicate that it supports Dynamic Flooding. This is indicated by the advertisement of this Sub-TLV.
2. Indicate the set of algorithms that it supports for distributed mode, if any.



In incremental deployments, understanding which nodes support Dynamic Flooding can be used to optimize the flooding topology. In distributed mode, knowing the capabilities of the nodes can allow the Area Leader to select the optimal algorithm.

The Dynamic Flooding Sub-TLV is advertised as a Sub-TLV of the IS-IS Router Capability TLV (242) [[RFC7981](#)] and has the following format:

```

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+
      |      Type      |      Length      | Algorithm... |
      +-+-+-+-+-+-+-+-+

```

Type: TBD7

Length: 0-255; number of Algorithms

Algorithm: zero or more numeric identifiers in the range 0-255 that identifies the algorithm used to calculate the flooding topology, as described in [Section 5.1.1](#).

### 5.1.3. IS-IS Area System IDs TLV

IS-IS Area System IDs TLV is only used in centralized mode.

The Area System IDs TLV is used by the Area Leader to enumerate the system IDs that it has used in computing the flooding topology. Conceptually, the Area Leader creates a list of system IDs for all nodes in the area, assigning indices to each system, starting with index 0.

Because the space in a single TLV is small, more than one TLV may be required to encode all of the system IDs in the area. This TLV may be present in multiple LSPs.

The format of the Area System IDs TLV is:

```

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+
      |      Type      |      Length      | Starting Index      |
      +-+-+-+-+-+-+-+-+
      |L| Reserved    | System IDs ...
      +-+-+-+-+-+-+-+-+
      System IDs continued ....
      +-+-+-+-+-+-+-+-+

```





Type: TBD2

Length:  $3 + (\text{System ID length} * (\text{number of System IDs}))$

Starting index: The index of the first system ID that appears in this TLV.

L (Last): This bit is set if the index of the last system ID that appears in this TLV is equal to the last index in the full list of system IDs for the area.

System IDs: A concatenated list of system IDs for the area.

If there are multiple IS-IS Area System IDs TLVs with the L bit set advertised by the same node, the TLV which specifies the smaller maximum index is used and the other TLV(s) with L bit set are ignored. TLVs which specify system IDs with indices greater than that specified by the TLV with the L bit set are also ignored.

#### **5.1.4. IS-IS Flooding Path TLV**

IS-IS Flooding Path TLV is only used in centralized mode.

The Flooding Path TLV is used to denote a path in the flooding topology. The goal is an efficient encoding of the links of the topology. A single link is a simple case of a path that only covers two nodes. A connected path may be described as a sequence of indices: (I1, I2, I3, ...), denoting a link from the system with index 1 to the system with index 2, a link from the system with index 2 to the system with index 3, and so on.

If a path exceeds the size that can be stored in a single TLV, then the path may be distributed across multiple TLVs by the replication of a single system index.

Complex topologies that are not a single path can be described using multiple TLVs.

The Flooding Path TLV contains a list of system indices relative to the systems advertised through the Area System IDs TLV. At least 2 indices must be included in the TLV. Due to the length restriction of TLVs, this TLV can contain at most 126 system indices.

The Flooding Path TLV has the format:



```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      | Starting Index      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Index 2        | Additional indices ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD3

Length: 2 \* (number of indices in the path)

Starting index: The index of the first system in the path.

Index 2: The index of the next system in the path.

Additional indices (optional): A sequence of additional indices to systems along the path.

#### 5.1.5. IS-IS Flooding Request TLV

The Flooding Request TLV allows a system to request an adjacent node to enable flooding towards it on a specific link in the case where the connection to adjacent node is not part of the existing flooding topology.

Nodes that support Dynamic Flooding MAY include the Flooding Request TLV in its IIH PDUs.

The Flooding Request TLV has the format:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      | Circuit Type |R| Scope      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|R| ...          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD9

Length: 1 + number of advertised Flooding Scopes

Circuit Type - circuit type as specified in IS-IS [[ISO10589](#)]

R bit: MUST be 0 and is ignored on receipt.



Scope: Flooding Scope for which the flooding is requested as defined by LSP Flooding Scope Identifier Registry defined by [RFC7356]. Inclusion of flooding scopes is optional and is only necessary if [RFC7356] is supported.

Circuit Flooding Scope MUST NOT be sent in the Flooding Request TLV and MUST be ignore if received.

If flooding was disabled on the received link due to Dynamic Flooding, then flooding MUST be temporarily enabled over the link for the specified Circuit Type(s) and Flooding Scope(s) received in the in the Flooding Request TLV. Flooding MUST be enabled until the Circuit Type or Flooding Scope is no longer advertised in the Flooding Request TLV or the TLV no longer appears in IIH PDUs received on the link.

When the flooding is temporarily enabled on the link for any Circuit Type or Flooding Scope due to received Flooding Request TLV, the receiver MUST perform standard database synchronization for the corresponding Circuit Type(s) and Flooding Scope(s) on the link. In the case of IS-IS, this results in setting SRM bit for all related LSPs on the link and sending CSNPs.

So long as the Flooding Request TLV is being received flooding MUST not be disabled for any of the Circuit Types or Flooding Scopes present in the Flooding Request TLV even if the connection between the neighbors is removed from the flooding topology. Flooding for such Circuit Types or Flooding Scopes MUST continue on the link and be considered as temporarily enabled.

## **5.2. OSPF LSAs and TLVs**

This section defines new LSAs and TLVs for both OSPFv2 and OSPFv3.

Following objects are added:

1. A TLV that is used to advertise the preference for becoming Area Leader.
2. A TLV that is used to indicate the support for Dynamic Flooding and the algorithms that the advertising node supports for distributed mode, if any.
3. OSPFv2 Opaque LSA and OSPFv3 LSA to advertise the flooding topology for centralized mode.
4. A TLV to carry the list of system IDs that compromise the flooding topology for the area.



The OSPF Dynamic Flooding Sub-TLV is advertised as a top-level TLV of the RI LSA that is defined in [RFC7770] and has the following format:





```

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Algorithm ... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD8

Length: number of Algorithms

Algorithm: as defined in [Section 5.1.1](#).

### 5.2.3. OSPFv2 Dynamic Flooding Opaque LSA

The OSPFv2 Dynamic Flooding Opaque LSA is only used in centralized mode.

The OSPFv2 Dynamic Flooding Opaque LSA is used to advertise additional data related to the dynamic flooding in OSPFv2. OSPFv2 Opaque LSAs are described in [\[RFC5250\]](#).

Multiple OSPFv2 Dynamic Flooding Opaque LSAs can be advertised by an OSPFv2 router. The flooding scope of the OSPFv2 Dynamic Flooding Opaque LSA is area-local.

The format of the OSPFv2 Dynamic Flooding Opaque LSA is as follows:

```

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     | Options | LS Type |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      TBD5      | Opaque ID |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Advertising Router |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     LS sequence number |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      LS checksum      | Length |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |
+-+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     TLVs |
|                                     ... |

```

OSPFv2 Dynamic Flooding Opaque LSA



The opaque type used by OSPFv2 Dynamic Flooding Opaque LSA is TBD. The opaque type is used to differentiate the various type of OSPFv2 Opaque LSAs and is described in [section 3 of \[RFC5250\]](#). The LS Type is 10. The LSA Length field [[RFC2328](#)] represents the total length (in octets) of the Opaque LSA including the LSA header and all TLVs (including padding).

The Opaque ID field is an arbitrary value used to maintain multiple Dynamic Flooding Opaque LSAs. For OSPFv2 Dynamic Flooding Opaque LSAs, the Opaque ID has no semantic significance other than to differentiate Dynamic Flooding Opaque LSAs originated by the same OSPFv2 router.

The format of the TLVs within the body of the OSPFv2 Dynamic Flooding Opaque LSA is the same as the format used by the Traffic Engineering Extensions to OSPF [[RFC3630](#)].

The Length field defines the length of the value portion in octets (thus a TLV with no value portion would have a length of 0). The TLV is padded to 4-octet alignment; padding is not included in the length field (so a 3-octet value would have a length of 3, but the total size of the TLV would be 8 octets). Nested TLVs are also 32-bit aligned. For example, a 1-octet value would have the length field set to 1, and 3 octets of padding would be added to the end of the value portion of the TLV. The padding is composed of zeros.

#### **[5.2.4.](#) OSPFv3 Dynamic Flooding LSA**

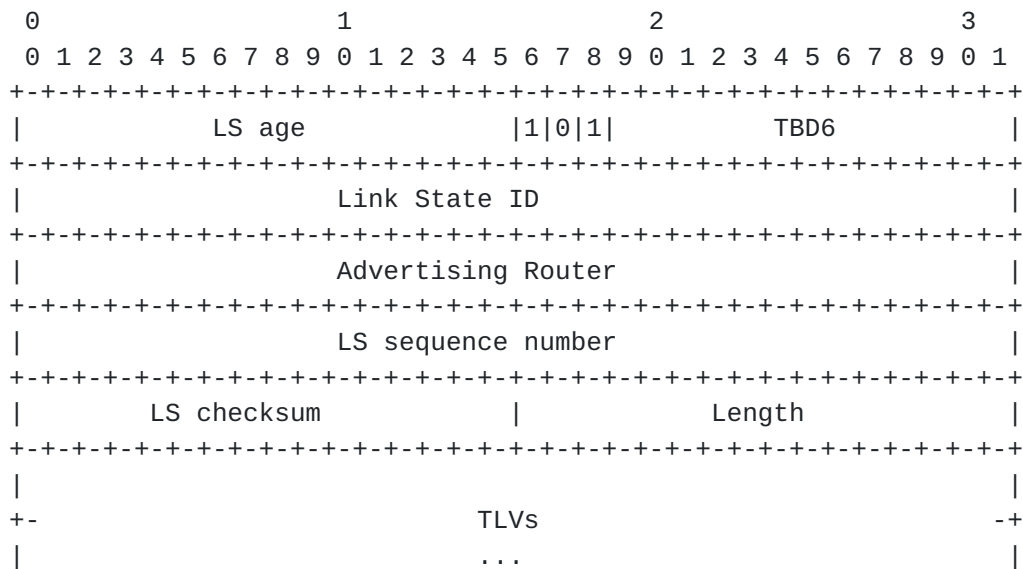
The OSPFv3 Dynamic Flooding Opaque LSA is only used in centralized mode.

The OSPFv3 Dynamic Flooding LSA is used to advertise additional data related to the dynamic flooding in OSPFv3.

The OSPFv3 Dynamic Flooding LSA has a function code of TBD. The flooding scope of the OSPFv3 Dynamic Flooding LSA is area-local. The U bit will be set indicating that the OSPFv3 Dynamic Flooding LSA should be flooded even if it is not understood. The Link State ID (LSID) value for this LSA is the Instance ID. OSPFv3 routers MAY advertise multiple Dynamic Flooding Opaque LSAs in each area.

The format of the OSPFv3 Dynamic Flooding LSA is as follows:





OSPFv3 Dynamic Flooding LSA

#### [5.2.5.](#) OSPF Area Router IDs TLV

The OSPF Area Router IDs TLV is a top level TLV of the OSPFv2 Dynamic Flooding Opaque LSA and OSPFv3 Dynamic Flooding LSA.

The OSPF Area Router IDs TLV is used by the Area Leader to enumerate the Router IDs that it has used in computing the flooding topology. Conceptually, the Area Leader creates a list of Router IDs for all routers in the area, assigning indices to each router, starting with index 0.

Because the space in a single OSPF Area Router IDs TLV is limited, more than one TLV may be required to encode all of the Router IDs in the area. This TLV may also recur in multiple OSPFv2 Dynamic Flooding Opaque LSAs or OSPFv3 Dynamic Flooding LSA, so that all Router IDs can be advertised.

The format of the Area Router IDs TLV is:









If FR-bit is signalled for an area for which the flooding on the link was disabled due to Dynamic Flooding, the flooding MUST be temporarily enabled over such link and area. Flooding MUST be enabled until FR-bit is no longer advertised in the OSPF LLS Extended



Options and Flags TLV or the OSPF LLS Extended Options and Flags TLV no longer appears in the OSPF Hellos.

When the flooding is temporarily enabled on the link for any area due to received FR-bit in OSPF LLS Extended Options and Flags TLV, the receiver MUST perform standard database synchronization for the corresponding area(s) on the link. If the adjacency is already in the FULL state, mechanism specified in [\[RFC4811\]](#) MUST be used for database resynchronization.

So long as the FR-bit is being received in the OSPF LLS Extended Options and Flags TLV for an area, flooding MUST not be disabled in such area even if the connection between the neighbors is removed from the flooding topology. Flooding for such area MUST continue on the link and be considered as temporarily enabled.

## **6. Behavioral Specification**

In this section, we specify the detailed behaviors of the nodes participating in the IGP.

### **6.1. Terminology**

We define some terminology here that is used in the following sections:

A node is considered reachable if it is part of the connected network graph. Note that this is independent of any constraints which may be considered when performing IGP SPT calculation (e.g., link metrics, OL bit state, etc.). Two-way-connectivity check MUST be performed before including an edge in the connected network graph.

Node is connected to the flooding topology, if it has at least one local link, which is part of the flooding topology.

Node is disconnected from the flooding topology when it is not connected to the flooding topology.

Current flooding topology - latest version of the flooding topology received (in case of the centralized mode) or calculated locally (in case of the distributed mode).

### **6.2. Flooding Topology**

The flooding topology MUST include all reachable nodes in the area.



If a node's reachability changes, the flooding topology **MUST** be recalculated. In centralized mode, the Area Leader **MUST** advertise a new flooding topology.

If a node becomes disconnected from the current flooding topology but is still reachable then a new flooding topology **MUST** be calculated. In centralized mode the Area Leader **MUST** advertise the new flooding topology.

The flooding topology **SHOULD** be bi-connected.

### **6.3. Leader Election**

Any node that is capable **MAY** advertise its eligibility to become Area Leader.

Nodes that are not reachable are not eligible as Area Leader. Nodes that do not advertise their eligibility to become Area Leader are not eligible. Amongst the eligible nodes, the node with the numerically highest priority is the Area Leader. If multiple nodes all have the highest priority, then the node with the numerically highest system identifier in the case of IS-IS, or Router-ID in the case of OSPFv2 and OSPFv3 is the Area Leader.

### **6.4. Area Leader Responsibilities**

If the Area Leader operates in centralized mode, it **MUST** advertise algorithm 0 in its Area Leader Sub-TLV. It also **MUST** compute and advertise a flooding topology for the area. The Area Leader may update the flooding topology at any time, however, it should not destabilize the network with undue or overly frequent topology changes.

If the Area Leader operates in centralized mode and needs to advertise a new flooding topology, it floods a new flooding topology on both the new and old flooding topologies.

### **6.5. Distributed Flooding Topology Calculation**

If the Area Leader advertises a non-zero algorithm in its Area Leader Sub-TLV, all nodes in the area that support Dynamic Flooding and the value of algorithm advertised by the Area Leader **MUST** compute the flooding topology based on the Area Leader's advertised algorithm.

Nodes that do not support the value of algorithm advertised by the Area Leader **MUST** continue to use standard flooding mechanism as defined by the protocol.



Nodes that do not support the value of algorithm advertised by the Area Leader MUST be considered as Dynamic Flooding incapable nodes by the Area Leader.

If the value of the algorithm advertised by the Area Leader is from the range 128-254 (private distributed algorithms), it is the responsibility of the network operator to guarantee that all nodes in the area have a common understanding of what the given algorithm value represents.

#### **6.6. Flooding Behavior**

Nodes that support Dynamic Flooding MUST use the flooding topology for flooding when possible, and MUST NOT revert to standard flooding when a valid flooding topology is available.

In some cases a node that supports Dynamic Flooding may need to add a local link(s) to the flooding topology temporarily, even though the link(s) is not part of the calculated flooding topology. This is termed "temporary flooding" and is discussed in [Section 6.7.1](#).

The flooding topology is calculated locally in the case of distributed mode. In centralized mode the flooding topology is advertised in the area link state database. Received link state updates, whether received on a link that is in the flooding topology or on a link that is not in the flooding topology, MUST be flooded on all links that are in the flooding topology, except for the link on which the update was received.

In centralized mode, if multiple flooding topologies are present in the area link state database, the node SHOULD flood on the on each of these topologies.

When the flooding topology changes on a node, either as a result of the local computation in distributed mode or as a result of the advertisement from the Area Leader in centralized mode, the node MUST continue to flood on both the old and new flooding topology for a limited amount of time. This is required to provide all nodes sufficient time to migrate to the new flooding topology.

#### **6.7. Treatment of Topology Events**

In this section, we explicitly consider a variety of different topological events in the network and how Dynamic Flooding should address them.





#### **6.7.1. Temporary Addition of Link to Flooding Topology**

In some cases a node that supports Dynamic Flooding may need to add a local link(s) to the flooding topology temporarily, even though the link(s) is not part of the calculated flooding topology. We refer to this as "temporary flooding" on the link.

When temporary flooding is enabled on the link, the flooding needs to be enabled from both directions on such link. To achieve that, the following steps MUST be performed:

Link State Database needs to be re-synchronised on the link. This is done using the standard protocol mechanisms. In the case of IS-IS, this results in setting SRM bit for all LSPs on the circuit and sending complete set of CSNPs on it. In OSPF, the mechanism specified in [[RFC4811](#)] is used.

Flooding is enabled locally on the link.

Flooding is requested from the neighbor using the mechanism specified in section [Section 5.1.5](#) or [Section 5.2.7](#).

The request for temporary flooding is withdrawn on the link when all of the following conditions are met:

Node itself is connected to the current flooding topology.

Adjacent node is connected to the current flooding topology.

Any change in the flooding topology MUST result in evaluation of the above conditions for any link on which the temporary flooding was enabled.

Temporary flooding is stopped on the link when both adjacent nodes stop requesting temporary flooding on the link.

#### **6.7.2. Local Link Addition**

If a local link is added to the topology, the protocol will form a normal adjacency on the link and update the appropriate link state advertisements for the nodes on either end of the link. These link state updates will be flooded on the flooding topology.

In centralized mode, the Area Leader, upon receiving these updates, may choose to retain the existing flooding topology or may choose to modify the flooding topology. If it elects to change the flooding topology, it will update the flooding topology in the link state database and flood it using the new flooding topology.



In distributed mode, any change in the topology, including the link addition, MUST trigger the flooding topology recalculation. This is done to ensure that all nodes converge to the same flooding topology, regardless of the time of the calculation.

Temporary flooding MUST be enabled on the newly added local link, if at least one of the following conditions are met:

The node on which the local link was added is not connected to the current flooding topology.

The new adjacent node is not connected to the current flooding topology.

Note that in this case there is no need to perform a database synchronization as part of the enablement of the temporary flooding, because it has been part of the adjacency bring-up itself.

If multiple local links are added to the topology before the flooding topology is updated, temporary flooding MUST be enabled on a subset of these links.

#### **6.7.3. Node Addition**

If a node is added to the topology, then at least one link is also added to the topology. [Section 6.7.2](#) applies.

#### **6.7.4. Failures of Link Not on Flooding Topology**

If a link that is not part of the flooding topology fails, then the adjacent nodes will update their link state advertisements and flood them on the flooding topology.

In centralized mode, the Area Leader, upon receiving these updates, may choose to retain the existing flooding topology or may choose to modify the flooding topology. If it elects to change the flooding topology, it will update the flooding topology in the link state database and flood it using the new flooding topology.

In distributed mode, any change in the topology, including the failure of the link that is not part of the flooding topology MUST trigger the flooding topology recalculation. This is done to ensure that all nodes converge to the same flooding topology, regardless of the time of the calculation.



#### **6.7.5. Failures of Link On the Flooding Topology**

If there is a failure on the flooding topology, the adjacent nodes will update their link state advertisements and flood them. If the original flooding topology is bi-connected, the flooding topology should still be connected despite a single failure.

If the failed local link represented the only connection to the flooding topology on the node where the link failed, the node **MUST** enable temporary flooding on a subset of its local links. This allows the node to send its updated link state advertisement(s) and also keep receiving link state updates from other nodes in the network before the new flooding topology is calculated and distributed (in the case of centralized mode).

In centralized mode, the Area Leader will notice the change in the flooding topology, recompute the flooding topology, and flood it using the new flooding topology.

In distributed mode, all nodes supporting dynamic flooding will notice the change in the topology and recompute the new flooding topology.

#### **6.7.6. Node Deletion**

If a node is deleted from the topology, then at least one link is also removed from the topology. The two sections above apply.

#### **6.7.7. Local Link Addition to the Flooding Topology**

If the new flooding topology is received in the case of centralized mode, or calculated locally in the case of distributed mode and the local link on the node that was not part of the flooding topology has been added to the flooding topology, the node **MUST**:

Re-synchronize the Link State Database over the link. This is done using the standard protocol mechanisms. In the case of IS-IS, this results in setting SRM bit for all LSPs on the circuit and sending a complete set of CSNPs. In OSPF, the mechanism specified in [[RFC4811](#)] is used.

Make the link part of the flooding topology and start flooding over it



#### **6.7.8. Local Link Deletion from the Flooding Topology**

If the new flooding topology is received in the case of centralized mode, or calculated locally in the case of distributed mode and the local link on the node that was part of the flooding topology has been removed from the flooding topology, the node **MUST** remove the link from the flooding topology.

The node **MUST** keep flooding on such link for a limited amount of time to allow other nodes to migrate to the new flooding topology.

If the removed local link represented the only connection to the flooding topology on the node, the node **MUST** enable temporary flooding on a subset of its local links. This allows the node to send its updated link state advertisement(s) and also keep receiving link state updates from other nodes in the network before the new flooding topology is calculated and distributed (in the case of centralized mode).

#### **6.7.9. Treatment of Disconnected Adjacent Nodes**

Every time there is a change in the flooding topology a node **MUST** check if there are any adjacent nodes that are disconnected from the current flooding topology. Temporary flooding **MUST** be enabled towards a subset of the disconnected nodes.

#### **6.7.10. Failure of the Area Leader**

The failure of the Area Leader can be detected by observing that it is no longer reachable. In this case, the Area Leader election process is repeated and a new Area Leader is elected.

In the centralized mode, the new Area Leader will compute a new flooding topology and flood it using the new flooding topology.

As an optimization, applicable to centralized mode, the new Area Leader **MAY** compute a new flooding topology that has as much in common as possible with the old flooding topology. This will minimize the risk of over-flooding.

In the distributed mode, the new flooding topology will be calculated on all nodes that support the algorithm that is advertised by the new Area Leader. Nodes that do not support the algorithm advertised by the new Area Leader will no longer participate in Dynamic Flooding and will revert to standard flooding.





#### **6.7.11. Recovery from Multiple Failures**

In the unlikely event of multiple failures on the flooding topology, it may become partitioned. The nodes that remain active on the edges of the flooding topology partitions will recognize this and will try to repair the flooding topology locally by enabling temporary flooding towards the nodes that they consider disconnected from the flooding topology until a new flooding topology becomes connected again.

Nodes where local failure was detected update their own link state advertisements and flood them on the remainder of the flooding topology.

In centralized mode, the Area Leader will notice the change in the flooding topology, recompute the flooding topology, and flood it using the new flooding topology.

In distributed mode, all nodes that actively participate in Dynamic Flooding will compute the new flooding topology.

Note that this is very different from the area partition because there is still a connected network graph between the nodes in the area. The area may remain connected and forwarding may still be effective.

### **7. IANA Considerations**

#### **7.1. IS-IS**

This document requests the following code point from the "sub-TLVs for TLV 242" registry (IS-IS Router CAPABILITY TLV).

Type: TBD1

Description: IS-IS Area Leader Sub-TLV

Reference: This document ([Section 5.1.1](#))

Type: TBD7

Description: IS-IS Dynamic Flooding Sub-TLV

Reference: This document ([Section 5.1.2](#))

This document requests that IANA allocate and assign two code points from the "IS-IS TLV Codepoints" registry. One for each of the following TLVs:



Type: TBD2

Description: IS-IS Area System IDs TLV

Reference: This document ([Section 5.1.3](#))

Type: TBD3

Description: IS-IS Flooding Path TLV

Reference: This document ([Section 5.1.4](#))

Type: TBD9

Description: IS-IS Flooding Request TLV

Reference: This document ([Section 5.1.5](#))

## **[7.2.](#) OSPF**

This document requests the following code points from the "OSPF Router Information (RI) TLVs" registry:

Type: TBD4

Description: OSPF Area Leader Sub-TLV

Reference: This document ([Section 5.2.1](#))

Type: TBD8

Description: OSPF Dynamic Flooding Sub-TLV

Reference: This document ([Section 5.2.2](#))

This document requests the following code point from the "Opaque Link-State Advertisements (LSA) Option Types" registry:

Type: TBD5

Description: OSPFv2 Dynamic Flooding Opaque LSA

Reference: This document ([Section 5.2.3](#))

This document requests the following code point from the "OSPFv3 LSA Function Codes" registry:

Type: TBD6



Description: OSPFv3 Dynamic Flooding LSA

Reference: This document ([Section 5.2.4](#))

This document requests a new bit in LLS Type 1 Extended Options and Flags registry:

Bit Position: TBD10

Description: Flooding Request bit

Reference: This document ([Section 5.2.7](#))

#### **[7.2.1](#). OSPF Dynamic Flooding LSA TLVs Registry**

This specification also requests one new registry - "OSPF Dynamic Flooding LSA TLVs". New values can be allocated via IETF Review or IESG Approval

The "OSPF Dynamic Flooding LSA TLVs" registry will define top-level TLVs for the OSPFv2 Dynamic Flooding Opaque LSA and OSPFv3 Dynamic Flooding LSAs. It should be added to the "Open Shortest Path First (OSPF) Parameters" registries group.

The following initial values are allocated:

Type: 0

Description: Reserved

Reference: This document

Type: 1

Description: OSPF Area Router IDs TLV

Reference: This document ([Section 5.2.5](#))

Type: 2

Description: OSPF Flooding Path TLV

Reference: This document ([Section 5.2.6](#))

Types in the range 32768-33023 are for experimental use; these will not be registered with IANA, and MUST NOT be mentioned by RFCs.



Types in the range 33024-65535 are not to be assigned at this time. Before any assignments can be made in the 33024-65535 range, there MUST be an IETF specification that specifies IANA Considerations that covers the range being assigned.

### **7.3. IGP**

IANA is requested to set up a registry called "IGP Algorithm Type For Computing Flooding Topology" under an existing "Interior Gateway Protocol (IGP) Parameters" IANA registries.

Values in this registry come from the range 0-255.

The initial values in the IGP Algorithm Type For Computing Flooding Topology registry are:

0: Reserved for centralized mode. Individual values are to be assigned according to the "Specification Required" policy defined in [[RFC8126](#)]

1-127: Available for standards action. Individual values are to be assigned according to the "Private Use" policy defined in [[RFC8126](#)]

128-254: Reserved for private use.

255: Reserved.

## **8. Security Considerations**

This document introduces no new security issues. Security of routing within a domain is already addressed as part of the routing protocols themselves. This document proposes no changes to those security architectures.

It is possible that an attacker could become Area Leader and introduce a flawed flooding algorithm into the network thus compromising the operation of the protocol. Authentication methods as describe in [[RFC5304](#)] and [[RFC5310](#)] for IS-IS, [[RFC2328](#)] and [[RFC7474](#)] for OSPFv2 and [[RFC5340](#)] and [[RFC4552](#)] for OSPFv3 SHOULD be used to prevent such attack.

## **9. Acknowledgements**

The authors would like to thank Sarah Chen for her contribution to this work.





The authors would like to thank Zeqing (Fred) Xia, Naiming Shen, Adam Sweeney and Olufemi Komolafe for their helpful comments.

The authors would like to thank Tom Edsall for initially introducing them to the problem.

## **10. References**

### **10.1. Normative References**

- [ISO10589] International Organization for Standardization, "Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Nov. 2002.
- [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>>.



- [RFC5613] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", [RFC 5613](#), DOI 10.17487/RFC5613, August 2009, <<https://www.rfc-editor.org/info/rfc5613>>.
- [RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", [BCP 100](#), [RFC 7120](#), DOI 10.17487/RFC7120, January 2014, <<https://www.rfc-editor.org/info/rfc7120>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", [RFC 7356](#), DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [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>>.
- [RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", [RFC 7981](#), DOI 10.17487/RFC7981, October 2016, <<https://www.rfc-editor.org/info/rfc7981>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 8126](#), DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

## **10.2. Informative References**

- [Clos] Clos, C., "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal Vol. 32(2), DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953, <<http://dx.doi.org/10.1002/j.1538-7305.1953.tb01433.x>>.
- [Leiserson]  
Leiserson, C., "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing", IEEE Transactions on Computers 34(10):892-901, 1985.



- [RFC2973] Balay, R., Katz, D., and J. Parker, "IS-IS Mesh Groups", [RFC 2973](#), DOI 10.17487/RFC2973, October 2000, <<https://www.rfc-editor.org/info/rfc2973>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", [RFC 3630](#), DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC4811] Nguyen, L., Roy, A., and A. Zinin, "OSPF Out-of-Band Link State Database (LSDB) Resynchronization", [RFC 4811](#), DOI 10.17487/RFC4811, March 2007, <<https://www.rfc-editor.org/info/rfc4811>>.
- [RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", [RFC 7938](#), DOI 10.17487/RFC7938, August 2016, <<https://www.rfc-editor.org/info/rfc7938>>.

#### Authors' Addresses

Tony Li (editor)  
Arista Networks  
5453 Great America Parkway  
Santa Clara, California 95054  
USA

Email: [tony.li@tony.li](mailto:tony.li@tony.li)

Peter Psenak (editor)  
Cisco Systems, Inc.  
Eurovea Centre, Central 3  
Pribinova Street 10  
Bratislava 81109  
Slovakia

Email: [ppsenak@cisco.com](mailto:ppsenak@cisco.com)

Les Ginsberg  
Cisco Systems, Inc.  
510 McCarthy Blvd.  
Milpitas, California 95035  
USA

Email: [ginsberg@cisco.com](mailto:ginsberg@cisco.com)



Tony Przygienda  
Juniper Networks, Inc.  
1194 N. Mathilda Ave  
Sunnyvale, California 94089  
USA

Email: [prz@juniper.net](mailto:prz@juniper.net)

Dave Cooper  
CenturyLink  
1025 Eldorado Blvd  
Broomfield, Colorado 80021  
USA

Email: [Dave.Cooper@centurylink.com](mailto:Dave.Cooper@centurylink.com)

Luay Jalil  
Verizon  
Richardson, Texas 75081  
USA

Email: [luay.jalil@verizon.com](mailto:luay.jalil@verizon.com)

Srinath Dontula  
ATT  
200 S Laurel Ave  
Middletown, New Jersey 07748  
USA

Email: [sd947e@att.com](mailto:sd947e@att.com)



