LSR Working Group
Internet Draft
Intended status: Standards Track
Expires: April 2019

Dave Allan
Ericsson

October 2018

A Distributed Algorithm for Constrained Flooding of IGP Advertisements
draft-allan-lsr-flooding-algorithm-00

Abstract

This document describes a distributed algorithm that can be applied to the problem of constraining IGP flooding in dense mesh topologies. The flooding topology utilizes two node-diverse spanning trees in order to provide complete coverage in the presence of any single failure while constraining the number of LSAs received by any IGP speaker connected to the flooding topology.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire in March 2019.

Copyright and License Notice

Copyright (c) 2018 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 (http://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

1. Introduction

This memo describes an algorithm suitable for reducing the quantity of IGP flooding in dense mesh networks. The only property that the algorithm is dependent upon is that there are at least two equal and diverse shortest paths between any pair of IGP speakers in order to meet the requirements elucidated in [Li]. The algorithm uses a re-purposing of the tie breaking algorithm used in 802.1aq Shortest Path Bridging as an element of construction of the flooding topology. It is not the intention of this memo to specify a complete solution, but to offer a foundation of an eventual solution.

1.1. Authors

David Allan

1.2. 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 RFC2119 [RFC2119].

2. Conventions used in this document

2.1. Terminology

Member Adjacency - An adjacency that has been determined to part of the flooding topology.

Member Node - A Participant node that is connected to the flooding topology.

Participant Adjacency - An adjacency between two participating nodes. It may be a member adjacency or a non-member adjacency

Non-Participant Adjacency - An adjacency where at least one of the two nodes is a not a Participating Node

Participating Node - An IGP speaker that has advertised the capability, and hence the intention, to participate in a flooding topology

3. Solution Overview

3.1. The Flooding Topology

A flooding topology is composed of a contiguously connected set of participating nodes.

The flooding topology constructed from two diversely rooted spanning trees. A participating node that is connected to the physical topology with a degree of two or greater and has at least two participating adjacencies will be bi-connected to the flooding topology.

The resulting flooding topology diameter will typically be two times the depth of the tree hierarchy. The compromise in this approach is that a subset of nodes in the network will not see a reduction of the replication burden from current practice when flooding LSAs as the degree of a subset of nodes in the flooding topology will correspond to the degree of the physical topology.

The protocol structure of flooded information is unmodified. A participant node may relay a received LSA onto member links of both spanning trees. Specific forwarding rules prevent undue flooding, the result being that every participant node that is bi-connected to the flooding topology will receive two copies of any flooded LSA in a fault free network. Participating nodes that due to network degradation are only singly connected will receive one copy. The forwarding rules are described in section 4.4.

3.2. Solution Applicability

This algorithm has been considered in the context of pure bipartite graphs, bipartite graphs modified with the addition of intra-tier adjacencies, and hierarchical variations of the above. Applicability to other network designs is for further study.

For all graphs the link costs are assumed to be common for all inter-tier links and common for any intra-tier links. Inter-tier and intra-tier links do not have to have the same cost.

3.3. Algorithm

The algorithm borrows from 802.1aq for the construction of the spanning trees used in this application. This is described in clause 28.5 of [802.1Q].

3.3.1. Algorithm Basics

The key component of the 802.1aq employed is the tie breaking algorithm. The original application of the algorithm was to produce a symmetrically congruent mesh of multicast trees and unicast forwarding whereby the path between any two nodes in the network was symmetric in both directions and congruent for both unicast and multicast traffic.

For this application the algorithm is used in the generation of two diversely rooted spanning trees that define the flooding topology.

As part of tree construction, the algorithm tie breaks between equal cost paths. When a tie is identified as part of a Dijkstra computation, a path-id is constructed for each equal cost path. A path-id is expressed as a lexicographically sorted list of the node-ids in the path. The set of equal cost paths is ranked, and the lowest selected. As an example:

Path-id 23-39-44-68-85 is ranked lower than

Path-id 23-44-59-63-90

When the path-ids are of unequal length, the path-ids with the fewest hops are ranked superior to the longer paths, and tie breaking is applied to select between the shorter path-ids. This is not expected to apply in the general case of the dense graphs this application is targeted at.

The node-ids used would be the loopback address of each node, therefore each path-id will be unique.

3.3.2. Generating Diverse Trees

The algorithm includes the concept of an "algorithm-mask", which is a value XOR'd with the node-ids prior to sorting into path IDs and ranking the paths. This permits the construction of diverse trees in a dense topology.

Two algorithm masks are used (zero and -1). When computing two trees from the same root, when there are at least two nodes to choose from at each distance from the root, fully diverse trees will be generated. When computing two trees from diverse roots in a tree architecture, diverse nodes will be selected in each tier in the hierarchy as the relay nodes to the next tier.

3.3.3. Desirable Properties Computation Wise

The algorithm has the property of permitting the pruning of intermediate state as a Dijkstra progresses as ties can be immediately evaluated, and the all but the selected path removed from further consideration. This is desirable when computing a Dijkstra in a dense graph as all path permutations do not need to be carried forward during computation. This permits the computation to be quite fast.

The resulting computational complexity would still be expressed as 2N(lnN).

4. Applying the Algorithm

4.1. Tree Generation

Each IGP speaker in the network has knowledge of each of the two spanning tree roots and the algorithm mask associated with each. This memo does not specify how root selection is performed and disseminated through the network, but does discuss selection requirements in section 4.5.

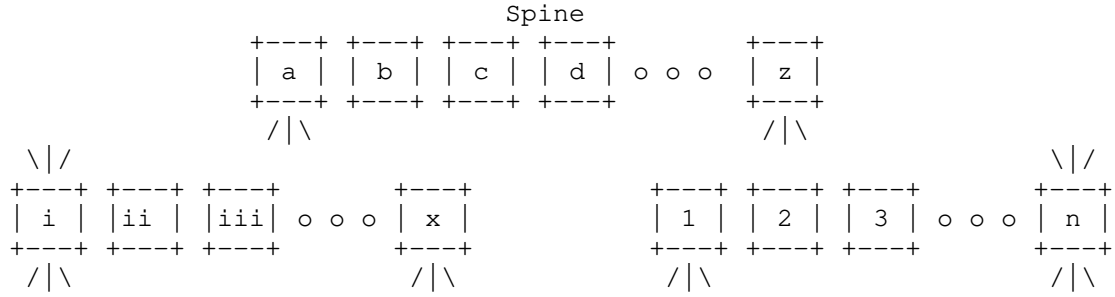
Each root has one of the two algorithm masks associated with it.

Each participating IGP speaker in the network computes a spanning tree from each the two roots (using the algorithm mask associated with each root) and from that can determine its own role in the flooding topology. The two spanning trees are designated the "low spanning tree" and the "high spanning tree".

The spanning trees are a starting point for a redundant topology. Unlike the commonly accepted operation of a spanning tree, in this application the distinction between upstream and downstream adjacencies is important and is an input to how a member node further relays any LSAs received. Upstream member adjacencies are in the direction of a root, and downstream member adjacencies are in the direction away from the root.

4.2. Illustrating the result

The following diagram illustrates the general layout of the flooding graph constructed using the algorithm as applied to a bi-partite style of tree (no intra tier links):



In the example, there are two tiers of switches. The spine (nodes a..z), and the next tier with two groups of nodes (i..x) and (1..n). The algorithm will select the node with the lowest node ID in each tier as the replicating node for the low spanning tree; 'a' and 'i' for the set of nodes connecting the spine and the next tier. The algorithm will select the nodes with highest node ID in the same set of nodes for the high spanning tree; 'z' and 'n' for the same set of nodes.

In the flooding topology:

- Node 'a' is connected to nodes i..x and 1..n for the low spanning tree.

- Node 'z' is connected to the same set of nodes for the high spanning tree.

- Node 'i' is connected to nodes 'a'..'z' for the low spanning tree, and

- Node 'n' is connected to the same nodes for the high spanning tree.

- All other nodes are bi-connected to the flooding topology

If there was a further tier added below nodes i..x, then 'i' and 'x' would be selected as the replicating nodes for the low and high spanning tree respectively. This is similarly true for nodes 1..n.

4.3. Interactions between Participating and Non-Participating Nodes

This solution proposes primarily only nodal behaviors with respect to constraining flooding to member adjacencies. To address the scenario

where the participating nodes were a subset of a larger network, it would be necessary to advertise the capability to participate in flood reduction.

This would then require that each participating node use this information to be able to identify the set of participating adjacencies and confine the spanning tree computation to the set of participating adjacencies in order to identify local set of member adjacencies. Interactions with non-participant adjacencies would conform to current practice.

4.4. Flooding of LSAs

The design of the protocol elements that are flooded is unmodified by this solution. Therefore, there is no additional information available to associate a received LSA with a given tree, nor is such information needed; the two spanning trees are not treated as unique entities in the flooding topology.

As per current practice, a node does not relay LSAs that it has already seen.

A new LSA received from an upstream member adjacency is flooded on:

- All downstream member adjacencies exclusive of the adjacency of arrival, irrespective of which tree the adjacencies are part of.

- All non-participant adjacencies

A new LSA received from a downstream member adjacency is flooded on:

- All other member adjacencies exclusive of the adjacency of arrival irrespective of which tree the adjacencies are part of.

- All non-participant adjacencies

A new LSA received from a member adjacency where upstream and downstream is ambiguous (it is an upstream member on one of the spanning trees and a downstream member on the other), is flooded on:

- All other member adjacencies exclusive of the adjacency of arrival irrespective of which adjacency the links are part of.

- All Non-Participant adjacencies

A new LSA received from a non-member adjacency is flooded on all member adjacency irrespective of which tree the adjacencies are part of (see sections 5.1 and 5.5).

4.5. Root Selection

The algorithm depends on tie breaking between sets of node IDs to produce diverse paths, therefore it does place some restrictions on root selection.

A root SHOULD be selected so that the root"s node-id when XORd with the associated algorithm mask is the lowest ranked node in the local tier in the tree hierarchy. This would be analogous to path-id ranking where the paths were all of length 1.

The root MUST NOT be selected such that the node-ID when XORd with the other root"s algorithm mask is the lowest ranked node. This would result in the root also being a transit node for the other spanning tree and produce a scenario whereby a single failure could render both spanning trees incomplete.

Roots MUST NOT be directly connected for either of the low or high spanning trees. If the topology does not permit this to be satisfied purely by root selection, then the inter-root adjacency must be pruned from the graph prior to spanning tree computation to ensure that diverse paths between the roots are used.

For a true bipartite graph, there are no other restrictions on node selection.

For a bipartite graph modified with inter-tier links, the roots MUST be placed in different tiers to ensure a pathological combination of link weights and node-ids does not result in a scenario where a single failure would render the flooding topology incomplete.

Other sources of failure may exist that may require an administrative component to root selection. This, for example, would ensure that both roots were not selected from a common shared risk group.

See also section 5.5.

4.6. Node Additions

A participating node that is added to the topology will initially not be served by the flooding topology. A participating node adjacent to that node is required to treat it as a non-participating node until such time as tree re-optimization has completed. At the end of tree

optimization, typically two adjacent participating nodes will have member adjacencies with the new node, so the ability to flood LSAs between the new node and the flooding topology will have been uninterrupted during the process.

5. Further work

5.1. Thoughts on Coexistence in the Context of a Larger Network

A node that had a combination of participating and non-participating adjacencies would be required to do the following:

- For any new LSA received on a participating adjacency, in addition to the rules for member adjacencies, it would also flood the LSA on all non-participating adjacencies.

- For any new LSA received on a non-participating adjacency, it would flood the LSA on all member adjacencies.

This is reflected in the forwarding rules described in section 4.4.

5.1.1. Multiple flooding Domains and the Severing of Flooding Domains

It is possible to envision several scenarios whereby there are sets of participating nodes that are not contiguously connected via participating adjacencies in a given IGP domain.

1. A node has been incorrectly configured as a participating node but has no participating adjacencies.

2. A participating node or set of nodes has become severed from the flooding topology but is still connected to other nodes in the network. Nodes in this set would still be able to compute a local extension of the flooding topology, but it would only be useful if the set was sufficiently large that a majority of the nodes were not connected to non-participants.

3. Procedures are designed to permit more than one flooding topology in an IGP domain. In which case participating nodes would have to be administratively configured to associate with a flooding topology instance.

5.2. Thoughts on Flooding Topology Re-Optimization

After a topology change, it is desirable that the flooding topology remain stable until the network has stabilized. However a single failure may render one of the spanning trees incomplete, such that a

further single failure could make the flooding topology incomplete. Therefore procedures should include re-optimization of the flooding topology after a topology change. In order to maintain complete coverage it would make sense not to recompute the spanning trees simultaneously.

One approach that would appear to make sense to separate in time network convergence, re-optimization of the low spanning tree and re-optimization of the high spanning tree.

The ideal would be to reoptimize an incomplete tree first, however this would require the participating nodes to maintain a complete map of all member adjacencies so that a common determination of the most degraded spanning tree and hence the order of re-optimization could be made.

5.3. Thoughts on Node and Network Initialization

A participating node at power up will be not be able to establish member links until it has synchronized with the network and the network is stable in the new topology. This suggests it simply treats power up similarly to how a topology change and network re-optimization is treated. The only difference being that it will flood all LSAs received or originated as per current practice until both spanning trees have stabilized.

5.4. Thoughts on Loop Prevention

802.1aq included additional mechanisms to prevent looping, a reverse path forwarding check, and digest exchange across adjacencies to ensure IGP synchronization.

Routing LSAs are not relayed if they are a duplicate, therefore destructive looping cannot occur and additional mitigation mechanisms are not required.

5.5. Thoughts on Pathological Failure Scenarios

While in a stable fault free network with sufficient mesh density of the types considered, the flooding topology used by this solution would ensure that no single failure rendered both spanning trees incomplete, it is also useful to consider multiple failure scenarios and if they can be mitigated.

Preliminary analysis suggests that in a tree network of sufficient mesh density, the only dual link failure that can render the flooding topology incomplete is if a participant node has failures in both

upstream member adjacencies. This can be partially mitigated if the node recognizes this scenario and reverts to flooding on all adjacencies. If the suggested procedures of 5.1.1 above are adopted, surrounding participating nodes that receive the LSA on a non-member adjacency will introduce the LSA into the flooding topology.

The pathological scenario is the simultaneous failure of both roots. This does suggest that root selection should place the roots two hops apart so there will be a constituency of participants that would observe a simultaneous failure of both upstream member adjacencies and revert to normal flooding.

6. Acknowledgements

The author would like to acknowledge Jerome Chiabaut for his original algorithm work that underpins this memo.

7. Security Considerations

For a future version of this document.

8. IANA Considerations

This memo requires no IANA allocations

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[802.1Q] 802.1Q (2014) IEEE Standard for Local and Metropolitan Area Networks--Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks

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

10. Author's Address

Dave Allan
Ericsson
2455 Augustine Drive
Santa Clara, CA 95054
USA
Email: david.i.allan@ericsson.com