Network Working Group
Internet-Draft
Intended status: Informational
Expires: June 2, 2021

R. White
S. Hegde
Juniper Networks
S. Zandi
LinkedIn

November 29, 2020

IS-IS Optimal Distributed Flooding for Dense Topologies
draft-white-lsr-distoptflood-00

Abstract

In dense topologies, such as data center fabrics based on the Clos and butterfly fabric topologies, flooding mechanisms designed for sparse topologies, when used in these dense topologies, can "overflood," or carry too many copies of topology and reachability to fabric devices. This results in slower convergence times and higher resource utilization. The modifications to the flooding mechanism in the Intermediate System to Intermediate System (IS-IS) link state protocol described in this document reduce resource utilization to a minimum, while increaseing convergence performance in dense topologies.

Note that a Clos fabric is used as the primary example of a desne flooding topology throughout this document. However, the flooding optimizations described in this document apply to any dense topology.

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 June 2, 2021.

Table of Contents

1. Introduction

1.1. Goals

The goal of this draft is to solve one specific set of problems involved in operating a link state protocol in a dense mesh topology. The problem with such topologies is the connectivity density, which causes too much information to be flooded (or too much repeated state to be flooded). Analysis and experiment show, for instance, that in a butterfly fabric of around 2500 intermediate systems, each intermediate system will receive 40+ copies of any changed LSP fragment. This not only wastes bandwidth and processor time, this dramatically slows convergence speed.

This document describes a set of modifications to existing IS-IS flooding mechanisms which minimize the number of LSP framgents received by individual intermediate systems, potentially to one copy per intermediate system. The mechanisms described in this document are similar to those implemented in OSPF to support mobile ad-hoc networks, as described in [RFC5449], [RFC5614], and [RFC7182]. These mechanisms have been widely deployed and tested.

1.2. Contributors

The following people have contributed to this draft: Nikos Triantafillis, Ivan Pepelnjak, Christian Franke, Hannes Gredler, Les Ginsberg, Naiming Shen, Uma Chunduri, Nick Russo, and Rodny Molina.

1.3. Experience

The modifications described in this draft have been implemented in the FR Routing open source routing stack, and hence are available for testing and modification. The implementation is part of the openfabric daemon, which can be conditionally compiled from isisd. Note openfabricd has further modifications are not described in this document.

Lab testing shows these modifications reduce flooding in a large scale emulated butterfly network topology; without these modifications, intermediate systems receive, on average, 40 copies of any changed LSP fragment. With these modifications, intermediate systems recieve, on average, two copies of any changed LSP fragment. In many cases, each intermediate system receives one copy of each changed LSP. In terms of performance, the modifications described here reduce convergence times by around 50%. A network that converges in about 30-40 seconds without these modifications converged in 15-20 seconds with these modifications. Processor load times were not checked, as this was an emulated environment.

1.4. Sample Network

The following spine and leaf fabric will be used to describe these modifications.

Figure 1

To reduce confusion (spine and leaf fabrics are difficult to draw in plain text art), this diagram does not contain the connections between devices. The reader should assume that each device in a given layer is connected to every device in the layer above it. For instance:

- 5A is connected to 4A, 4B, 4C, 4D, 4E, and 4F
- 5B is connected to 4A, 4B, 4C, 4D, 4E, and 4F
- 4A is connected to 3A, 3B, 3C, 3D, 3E, 3F, 5A, 5B, 5C, 5D, 5E, and 5F
- 4B is connected to 3A, 3B, 3C, 3D, 3E, 3F, 5A, 5B, 5C, 5D, 5E, and 5F
- etc.

The tiers or stages of the fabric are also marked for easier reference. T0 is assumed to be connected to application servers, or rather they are Top of Rack (ToR) intermediate systems. The remaining tiers, T1 and T2, are connected only to the fabric itself.

2. Flooding Modifications

Flooding is perhaps the most challenging scaling issue for a link state protocol running on a dense, large scale fabric. This section describes modifications to the IS-IS flooding process to reduce flooding load on a dense or mesh topology.

2.1. Optimizing Flooding

To reduce the flooding of link state information in the form of Link State Protocol Data Units (LSPs), the following tables are required to compute a set of reflooders:

- Neighbor List (NL) list: The set of neighbors
- Neighbor's Neighbors (NN) list: The set of neighbor's neighbors; this can be calculated by running SPF truncated to two hops
- Do Not Reflood (DNR) list: The set of neighbors who should have LSPs (or fragments) who should not reflood LSPs
- Reflood (RF) list: The set of neighbors who should flood LSPs (or fragments) to their adjacent neighbors to ensure synchronization

NL is set to contain all neighbors, and sorted deterministically (for instance, from the highest IS identifier to the lowest). All intermediate systems within a single fabric SHOULD use the same mechanism for sorting the NL list. NN is set to contain all neighbor's neighbors, or all intermediate systems that are two hops away, as determined by performing a truncated SPF. The DNR and RF tables are initially empty. To begin, the following steps are taken to reduce the size of NN and NL:

- Remove all intermediate systems from NL and NN that in the shortest path to the IS that originated the LSP

Then, for every IS in NL:

- If the current entry in NL is connected to any entries in NN:
  - Move the IS to RF
  - Remove the intermediate systems connected to the IS from NN
- Else move the IS to DNR

The calculation terminates when the NL is empty.

When flooding, LSPs transmitted to adjacent neighbors on the RF list will be transmitted normally. Adjacent intermediate systems on this list will reflood received LSPs into the next stage of the topology, ensuring database synchronization. LSPs transmitted to adjacent neighbors on the DNR list, however, MUST be transmitted using a circuit scope PDU as described in [RFC7356].

2.2. Flooding Failures

It is possible in some failure modes for flooding to be incomplete because of the flooding optimizations outlined. Specifically, if a reflooder fails, or is somehow disconnected from all the links across which it should be reflooding, it is possible an LSP is only partially flooded through the fabric. To prevent such situations, any IS receiving an LSP transmitted using DNR SHOULD:

- Set a short timer; the default should be less than one second
- When the timer expires, send a Complete Sequence Number Packet (CSNP) to all neighbors
- Process any Partial Sequence Number Packets (PSNPs) as required to resynchronize
- If a resynchronization is required, notify the network operator through a network management system

3. Use of Flooding Leaders and Flooding Mechanism Advertisements

[I-D.ietf-lsr-dynamic-flooding], section 5.1.1, describes the election of a flooding domain leader, which can advertise the kind of flooding reduction mechanism used in the flooding domain. Implementations of this draft MAY implement the election and advertisement of a flooding domain leader as described in section 5.1.1 of [I-D.ietf-lsr-dynamic-flooding]. If the election of a flooding domain leader is implemented, implementations SHOULD also advertise the flooding mechanism using the IS-IS Dynamic Flooding Sub-TLV described in section 5.1.2 of [I-D.ietf-lsr-dynamic-flooding], using Algorithm number (TBD).

4. Security Considerations

This document outlines modifications to the IS-IS protocol for operation on high density network topologies. Implementations SHOULD implement IS-IS cryptographic authentication, as described in [RFC5304], and should enable other security measures in accordance with best common practices for the IS-IS protocol.

5. References

5.1. Normative References

[I-D.ietf-lsr-dynamic-flooding]
Li, T., Psenak, P., Ginsberg, L., Chen, H., Przygienda, T., Cooper, D., Jalil, L., Dontula, S., and G. Mishra, "Dynamic Flooding on Dense Graphs", draft-ietf-lsr-dynamic-flooding-07 (work in progress), June 2020.

[ISO10589]
International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Second Edition, 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>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <https://www.rfc-editor.org/info/rfc2629>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/info/rfc5120>.

[RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <https://www.rfc-editor.org/info/rfc5301>.

[RFC5303] Katz, D., Saluja, R., and D. Eastlake 3rd, "Three-Way Handshake for IS-IS Point-to-Point Adjacencies", RFC 5303, DOI 10.17487/RFC5303, October 2008, <https://www.rfc-editor.org/info/rfc5303>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

[RFC5309] Shen, N., Ed. and A. Zinin, Ed., "Point-to-Point Operation over LAN in Link State Routing Protocols", RFC 5309, DOI 10.17487/RFC5309, October 2008, <https://www.rfc-editor.org/info/rfc5309>.

[RFC5311] McPherson, D., Ed., Ginsberg, L., Previdi, S., and M. Shand, "Simplified Extension of Link State PDU (LSP) Space for IS-IS", RFC 5311, DOI 10.17487/RFC5311, February 2009, <https://www.rfc-editor.org/info/rfc5311>.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316, December 2008, <https://www.rfc-editor.org/info/rfc5316>.

[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>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

5.2. Informative References

[I-D.ietf-isis-segment-routing-extensions] Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-25 (work in progress), May 2019.

[RFC3277] McPherson, D., "Intermediate System to Intermediate System (IS-IS) Transient Blackhole Avoidance", RFC 3277, DOI 10.17487/RFC3277, April 2002, <https://www.rfc-editor.org/info/rfc3277>.

[RFC3719] Parker, J., Ed., "Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)", RFC 3719, DOI 10.17487/RFC3719, February 2004, <https://www.rfc-editor.org/info/rfc3719>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[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>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[RFC5449] Baccelli, E., Jacquet, P., Nguyen, D., and T. Clausen, "OSPF Multipoint Relay (MPR) Extension for Ad Hoc Networks", RFC 5449, DOI 10.17487/RFC5449, February 2009, <https://www.rfc-editor.org/info/rfc5449>.

[RFC5614] Ogier, R. and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding", RFC 5614, DOI 10.17487/RFC5614, August 2009, <https://www.rfc-editor.org/info/rfc5614>.

[RFC5820] Roy, A., Ed. and M. Chandra, Ed., "Extensions to OSPF to Support Mobile Ad Hoc Networking", RFC 5820, DOI 10.17487/RFC5820, March 2010, <https://www.rfc-editor.org/info/rfc5820>.

[RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <https://www.rfc-editor.org/info/rfc5837>.

[RFC6232] Wei, F., Qin, Y., Li, Z., Li, T., and J. Dong, "Purge Originator Identification TLV for IS-IS", RFC 6232, DOI 10.17487/RFC6232, May 2011, <https://www.rfc-editor.org/info/rfc6232>.

[RFC7182] Herberg, U., Clausen, T., and C. Dearlove, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)", RFC 7182, DOI 10.17487/RFC7182, April 2014, <https://www.rfc-editor.org/info/rfc7182>.

[RFC7921] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", RFC 7921, DOI 10.17487/RFC7921, June 2016, <https://www.rfc-editor.org/info/rfc7921>.

Appendix A. Flooding Optimization Operation

Recent testing has shown that flooding is largely a "non-issue" in terms of scaling when using high speed links connecting intermediate systems with reasonable processing power and memory. However, testing has also shown that flooding will impact convergence speed even in such environments, and flooding optimization has a major impact on the performance of a link state protocol in resource constrained environments. Some thoughts on flooding optimization in general, and the flooding optimization contained in this document, follow.

There are two general classes of flooding optimization available for link state protocols. The first class of optimization relies on a centralized service or server to gather the link state information and redistribute it back into the intermediate systems making up the fabric. Such solutions are attractive in many, but not all, environments; hence these systems compliment, rather than compete with, the system described here. Systems relying on a service or server necessarily also rely on connectivity to that service or server, either through an out-of-band network or connectivity through the fabric itself. Because of this, these mechanisms do not apply to all deployments; some deployments require underlying reachability regardless of connectivity to an outside service or server.

The second possibility is to create a fully distributed system that floods the minimal amount of information possible to every intermediate system. The system described in this draft is an example of such a system. Again, there are many ways to accomplish this goal, but simplicity is a primary goal of the system described in this draft.

The system described here divides the work into two different parts; forward and reverse optimization. The forward optimization begins by finding the set of intermediate systems two hops away from the flooding device, and choosing a subset of connected neighbors that will successfully reach this entire set of intermediate systems, as shown in the diagram below.

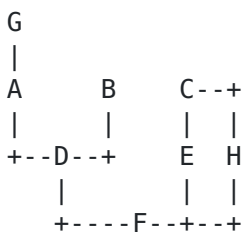

Figure 2

If F is flooding some piece of information, then it will find the entire set of intermediate systems within two hops by discovering its neighbors and their neighbors from the local LSDB. This will include A, B, C, D, and E--but not G. From this set, F can determine that D can reach A and B, while a single flood to either E or H will reach C. Hence F can flood to D and either E or H to reach C. F can choose to flood to D and E normally. Because H still needs to receive this new LSP (or fragment!), but does not need to reflood to C, F can send the LSP using link local signaling. In this case, H will receive and process the new LSP, but not reflood it.

Rather than carrying the information necessary through hello extensions, as is done in [RFC5820], the neighbors are allowed to complete initial synchronization, and then a truncated shortest path tree is built to determine the "two hop neighborhood." This has the advantage of using mechanisms already used in IS-IS, rather than adding new processes. The risk with this process is any LSPs flooded through the network before this initial calculation takes place will be suboptimal. This "two hop neighborhood" process has been used in OSPF deployments for a number of years, and has proven stable in practice.

Rather than setting a timer for reflooding, the implementation described here uses IS-IS' ability to describe the entire database using a CSNP to ensure flooding is successful. This adds some small amount of overhead, so there is some balance between optimal flooding and ensuring flooding is complete.

The reverse optimization is simpler. It relies on the observation that any intermediate system between the local IS and the origin of the LSP, other than in the case of floods removing an LSP from the shared LSDB, should have already received a copy of the LSP. For instance, if F originates an LSP in the figure above, and E refloods the LSP to C, C does not need to reflood back to F if F is on its shortest path tree towards F. It is obvious this is not a "perfect" optimization. A perfect optimization would block flooding back along a directed acyclic graph towards the originator. Using the SPT,

however, is a quick way to reduce flooding without performing more calculations.

The combination of these two optimizations have been seen, in testing, to reduce the number of copies any IS receives from the tens to precisely one.

Authors' Addresses

Russ White
Juniper Networks

Email: russ@riw.us

Shraddha Hegde
Juniper Networks

Email: shraddha@juniper.net

Shawn Zandi
LinkedIn

Email: szandi@linkedin.com