Abstract

There is significant overhead in OSPF when a router has to establish adjacencies with every peer with whom it can verify 2-way connectivity. OSPF supports the broadcast network type for these scenarios, where you only have to peer with the designated router (DR). However, a full mesh of connectivity is required for proper operation and this doesn’t help in networks with overlapping partial meshes of connectivity. This draft proposes a technique to reduce the number of adjacencies based on shortest path tree (SPT) reachability information.
Table of Contents

1. Introduction .................................................. 3
2. Previous Related Work ........................................ 4
3. Smart Peering ................................................... 5
   3.1. SPT Reachability Heuristics ............................. 5
   3.2. State Machine ............................................ 5
4. Advertise the 2-way link in Router-LSA ..................... 7
   4.1. Unsynchronized Adjacencies ............................... 7
   4.2. Unsynchronized SPT ....................................... 9
   4.3. Flooding Considerations .................................. 10
   4.4. Overlapping Relay [OR] election impact ................. 10
5. Security Considerations ....................................... 11
6. IANA Considerations ......................................... 12
7. References .................................................... 13
   7.1. Normative References .................................... 13
   7.2. Informative References .................................. 13
Appendix A. Contributors ....................................... 14
Appendix B. Acknowledgments ................................... 15
Author’s Address ............................................... 16
Intellectual Property and Copyright Statements ............... 17
1. Introduction

In OSPF [OSPFV2] [OSPFV3], nodes establish an adjacency by first verifying 2-way connectivity between them and then synchronizing their link state databases. Once the peering relationship is complete and the adjacency is established, the nodes will continue to advertise each other in their LSAs. As a result, the peers are maintained in the link state database and are included in all SPF calculations. During the reliable flooding process, a node must ensure that each peer has indeed received the flooded routing update via an acknowledgment and retransmission mechanism.

Consequently, maintaining an adjacency for a particular peer is a tradeoff between the added redundancy in routing paths and network reachability versus the associated overhead (memory consumption, SPF computations, routing overhead, and network convergence).

Consider the possibility of reducing the number of adjacencies that a node maintains without compromising reachability and redundancy. This will have direct implications on network scalability and is especially attractive in environments where the network topology is dynamic. For example, in a Mobile Ad-Hoc Network (MANET) where nodes are mobile and the topology is constantly changing, it seems highly desirable to 'intelligently' become adjacent with only selected peers and not establish a peering session with every node that comes within transmission range. Selective peering can be particularly useful in avoiding the peering process for unstable nodes, i.e., nodes that come in and out of transmission range.
2. Previous Related Work

The formation of a FULL adjacency requires the discovery (2-way relationship) and the database synchronization. To prevent from achieving the FULL state, others have taken the approach of modifying link state protocols to use periodic advertisements (instead of a database exchange). The result is that neighbor discovery is still required, but routing information is learned over time. An example of this approach is:

- OSPFv2 Wireless Interface Type [WINTF]

  * where the use of periodic advertisements "eliminates the formation of full adjacencies on wireless interfaces; all neighbor states beyond 2-Way are not reached, and no database synchronization is performed". 

What we propose in this specification goes a step further by not requiring the formation and maintenance of neighbor state (2-way, or other) *and* without changing the route distribution mechanisms in the link state protocols. In other words, the mechanism described is completely backwards compatible.
3. Smart Peering

Two routers are defined synchronized when they have identical link state database. To limit the number of neighbors that are formed, an algorithm is needed to select which neighbors with whom to peer.

The algorithm MUST provide reachability to every possible destination in the network, just like when normal adjacency formation processes are used. We should always peer with a neighbor if it provides our only path to currently unreachable destinations.

3.1. SPT Reachability Heuristics

The peering decision is really a local matter to a router. If a router can ensure that reachability to other nodes is available without bringing up a new adjacency, it can choose not to bring up the new adjacency.

We propose an algorithm which uses the existing information about a new neighbor’s reachability in the SPT. If the two routers can already reach each other in the SPT, it is not necessary to form an adjacency between them.

The decision to peer or not, is made when a hello is received. When a hello is received from a new neighbor or a neighbor in a state lower than Exchange:

- A check is made in the link state database to see if the peer is already reachable in the SPT.
  - If the peer is either not known in the SPT or is not reachable, we start the Exchange process.
  - If the peer is reachable than bringing up adjacency with this neighbor does not provide reachability to any new destinations.

Let’s take an example of a single OSPF area. This check would look for the neighbor’s Router LSA. If the LSA is present in the database and is reachable in the SPT, we have a chance to suppress adjacency formation.

It’s worth noting that as the number of links and redundancy in the network is reduced, the likelihood of suboptimal routing increases.

3.2. State Machine

The state machine of a basic implementation of this algorithm is provided below: An implementation MAY use some heuristics below (step
(3)), beyond the SPT reachability to decide whether or not it considers a new adjacency to be of value.

```
(1)
|Receive a hello |
|from a new potential |
|neighbor |

(2)
|Check to see if there |
|is a router LSA from |
|the new potential |
|neighbor in the link |
|state database, which |
|is reachable in SPT |

(3)
|yes |

(3a),
|Determine if the new |
|link cost is better |
|than the current path |
|cost by a configured |
|amount |

(3b)
|Determine if the |
|number of redundant |
|paths to the potential |
|neighbor is < the |
|maximum configured |
|value |

(4) form a new neighbor |
(5) do not become |
neighbors 
```
4. Advertise the 2-way link in Router-LSA

The technique described in Section 3 minimizes the number of adjacencies in highly meshed environments. This is especially useful when the network is in motion and the average adjacency lifetime is small.

However, it suffers from an undesirable side effect of limiting the number of transit links available to forward traffic.

An implementation may choose to allow some (or even all) of these 2-way state neighbors to be announced in the Router-LSA. Since the state remains 2-way, we don’t incur control plane (database sync and flooding) overhead. However, advertising the link in the Router-LSA makes the link available to the data plane.

This can be safely done if the neighbor is reachable in a special SPT constructed by ignoring any other 2-way links in the network. This optional optimization is described below.

4.1. Unsynchronized Adjacencies

If the new neighbor is already reachable in the SPT, there is no urgency in doing a full database sync with it. These are the steps we need to perform when a neighbor has reached 2-way state.

Note that when we say SPT in this section, we mean the special SPT constructed based on rules in Section 4.2.

- After 2-WayReceived event, check if the neighbor is reachable in the SPT. If yes, mark the neighbor as FULL with respect to link advertisement.

- This means that the router-LSA or network-LSA link corresponding to the neighbor is advertised as if the neighbor is FULL.

- The adjacency information is constructed with U-bit (see below).

- Database synchronization is postponed:
  - By a configured amount of time
  - Until the time it’s absolutely "necessary"

In either case, if a database sync is currently pending, it is started as soon as we detect the neighbor is no longer reachable in the SPT. The database sync can be done by Out-of-band Sync [OOB], which maintains the current adjacency and does the sync in the
background. A normal Resync can alternately be done with the drawback of adjacency flap.

In standard OSPF we first bring up adjacency and then announce a transit link. The approach described above will allow the link to be used as a forwarding path very quickly and still allows the database to be synchronized in a timely fashion when the alternate flooding path has recently been broken.

There is a circular dependency issue which also needs to be resolved. Once you start announcing the link, the shortest path will likely be via this very link. So it’s non-trivial to detect when the alternate dependent path is gone. What we would like to be able to detect is that the neighbor is reachable via a path which doesn’t traverse an unsynchronized path.

We have generally solved this class of problems by running an SPF and pretending that the link in question doesn’t exist. It doesn’t require a full SPF, just enough to see if ANY other path is available to reach the neighbor. The worst case is when the alternate path is really gone and we find that out by building a full SPT. This needs to be done every time the link state database changes, and for EACH link which has SPT dependence for it’s viability. This approach has scalability concerns and is not considered further here.

We can achieve the same results with just ONE additional SPF which is capable of ignoring these Unsynchronized links. The result from this SPT can be used to satisfy the reachability condition for ANY number of Unsynchronized Adjacencies. This basically requires that we can actually tell the difference between a normal FULL adjacency and this new Unsynchronized Adjacency. We can do this in one of two ways:

[A] Define LD Options and use a bit in it, as shown below:
Link Description in a Router-LSA

LD Options
Link Description options. Used to specify some special capability or state of a link.

-----+
| 0   | U |
-----+

LD Options

U-bit
The "Unsynchronized" bit. This is set if the adjacency is being announced before databases are fully synchronized.

This approach is backward compatible because the only routers looking at this bit are those that support the mechanisms specified in this document.

[B] Introducing a new link type in Router-LSA.

This is a much more complex solution with backward compatibility concerns due to the fact that unknown link type handling is not defined in OSPF standard [OSPFV2]. Hence this solution isn’t considered further.

4.2. Unsynchronized SPT

Whenever link state changes happen, we need to run ONE additional SPF by ignoring all links with U-bit set. This SPT is then consulted to see if any of our Unsynchronized Adjacencies need to start database sync. This SPT is also consulted when a new neighbor goes into 2-way to decide if we should form the adjacency immediately or defer it for later.
4.3. Flooding Considerations

One of the main goals in trying to delay the database synchronization is to be able to reduce unnecessary OSPF packets traversing these links. Since the unsynchronized Adjacencies remain in 2-way state, OSPF updates will not be flooded over the corresponding interfaces resulting in additional savings.

An option is provided to enable or disable flooding over these Unsynchronized Adjacencies. The advantage of allowing flooding is being able to use more links for control plane purposes. We will still have the savings of not having to form the adjacency.

4.4. Overlapping Relay [OR] election impact

The overlapping relay election algorithm uses the two hop neighborhood it gleans from our neighbor’s Router-LSAs. The introduction of Unsynchronized Adjacencies needs to be considered in the relay election algorithm.

If flooding is enabled on unsynchronized Adjacencies, no change is needed in the relay election algorithm. If flooding is disabled, then the relay election algorithm needs to prune neighbors that are connected via an Unsynchronized Adjacency from our 1-hop and 2-hop neighbor lists.
5. Security Considerations

The function described in this document does not create any new security issues for the OSPF protocol.

Security considerations for the base OSPF protocol are covered in [OSPFV2], [OSPFV3] and [OSPFV3SEC].
6. IANA Considerations

A new registry is defined for LD Options.

The values are defined in Section 4.1. All additions to LD Options are subject to OSPF WG review and require IETF standards action.
7. References

7.1. Normative References


7.2. Informative References


Appendix A. Contributors

These people made significant contributions to the document:

Dave Cook                         Russ White
Cisco Systems                     Cisco Systems
7025-4 Kit Creek Road             7025-4 Kit Creek Road
RTP, NC 27709                     RTP, NC 27709
USA                               USA
Email: dacook@cisco.com           Email: riw@cisco.com

Alvaro Retana                      Yi Yang
Cisco Systems                     Cisco Systems
7025-4 Kit Creek Road             7025-4 Kit Creek Road
RTP, NC 27709                     RTP, NC 27709
USA                               USA
Email: aretana@cisco.com           Email: yiya@cisco.com

Madhavi W. Chandra                 Acee Lindem
Cisco Systems                     Cisco Systems
7025-4 Kit Creek Road             7025-4 Kit Creek Road
RTP, NC 27709                     RTP, NC 27709
USA                               USA
Email: mchandra@cisco.com         Email: acee@cisco.com
Appendix B. Acknowledgments

The RFC text was produced using Marshall Rose’s xml2rfc tool.
Author’s Address

Abhay Roy (editor)
Cisco Systems
170 W. Tasman Dr
San Jose, CA  95134
USA

Email: akr@cisco.com
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.