Internet Engineering Task Force (IETF)
Request for Comments: 5820
Category: Experimental
ISSN: 2070-1721
March 2010

A. Roy, Ed.
Cisco Systems
M. Chandra, Ed.

Extensions to OSPF to Support Mobile Ad Hoc Networking

Abstract

This document describes extensions to OSPF to support mobile ad hoc networks (MANETs). The extensions, called OSPF-OR (OSPF-Overlapping Relay), include mechanisms for link-local signaling (LLS), an OSPF-MANET interface, a simple technique to reduce the size of Hello packets by only transmitting incremental state changes, and a method for optimized flooding of routing updates. OSPF-OR also provides a means to reduce unnecessary adjacencies to support larger MANETs.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Mobile ad hoc networks (MANETs) have been an area of study for some time within various working groups and areas within the IETF, various military branches, and various government agencies. Recently, networks with mobile ad hoc requirements have been proposed and are being seriously considered for deployment in the near term, which means the concepts and research now need to be applied to deployed networks. Towards that end, this document applies many of the principles and concepts learned through prior work to [OSPFv3], along with new concepts based on current requirements.

1.1. Problem Statement

MANETs are synonymous with packet radio networks, which have been around since the 1960s in a limited military capacity. With the boom in mobile devices and wireless communications, MANETs are finding scope in commercial and military environments. The aim of these networks is to support robust and efficient communication in a mobile wireless network by incorporating routing functionality into mobile nodes.

A MANET is an autonomous set of nodes distributed over a wide geographical area that communicate over bandwidth-constrained wireless links. Each node may represent a transmitter, receiver, or relay station with varying physical capabilities. Packets may traverse through several intermediate (relay) nodes before reaching their destination. These networks typically lack infrastructure: nodes are mobile, and there is no central hub or controller; thus, there is no fixed network topology. Moreover, MANETs must contend with a difficult and variable communication environment. Packet transmissions are plagued by the usual problems of radio communication, which include propagation path loss, signal multipath and fading, and thermal noise. These effects vary with terminal movement, which also induces Doppler spreading in the frequency of the transmitted signal. Finally, transmissions from neighboring terminals, known as multi-access interference, hostile jammers, and impulsive interference, e.g., ignition systems, generators, and other non-similar in-band communications, may contribute additional interference.

Given this nature of MANETs, the existence of a communication link between a pair of nodes is a function of their variable link quality, including signal strength and bandwidth. Thus, routing paths vary, based on environment and the resulting network topology. In such networks, the topology may be stable for periods of time and then suddenly become unpredictable. Since MANETs are typically decentralized systems, there are no central controllers or specially designated routers to determine the routing paths as the topology changes. All of the routing decisions and forwarding (relaying) of packets must be done by the nodes themselves, and communication is on a peer-to-peer basis.

1.2. Motivation for Extending OSPF to Support MANETs

The motivation to extend a standard protocol, OSPF (described in [OSPF] and [OSPFv3]), to operate on MANETs is twofold. The primary reason is for interoperability -- MANET devices need to be able to work when plugged into a wireline network in as many cases as possible. The junction point between a MANET and wire-line network should also be as fluid as possible, allowing a MANET to "plug in" to just about any location within a wire-line network, and also find connectivity, etc., as needed.

While routes could be redistributed between two routing protocols, one designed just for wire-line networks, and the other just for MANETs, this adds complexity and overhead to the MANET/wireline interface, increases the odds of an error being introduced between the two domains, and decreases flexibility.

The second motivation is that OSPF is a well-understood and widely deployed routing protocol. This provides a strong basis of experience and skills from which to work. A protocol that is known to work can be extended, rather than developing a new protocol that must then be completely troubleshot, tested, and modified over a number of years. Working with a well-known protocol allows development effort to be placed in a narrowly focused area, rather than rebuilding, from scratch, many things that are already known to work.

2. Requirements Notation

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 [KEY].

3. Proposed Enhancements

This document proposes modifications to [OSPFv3] to support mobile ad hoc networks (MANETs). Note that it is possible to use the mechanisms defined in Sections 3.2 and 3.3 independently of one another.

The challenges with deploying standard [OSPFv3] in a MANET environment fit into two categories. First, traditional link-state routing protocols are designed for a statically configured environment. As a result, most of the configuration is done manually when a new router is placed in the network. Thus, OSPF will not function in an environment where routers interconnect and disconnect in somewhat random topologies and combinations. There are modifications that must be made in order for routers running the same protocol to communicate in a heterogeneous and dynamic environment.

Currently there is no defined interface type that describes a wireless network. Wireless links have characteristics of both multi-access and point-to-multipoint links. Treating wireless links as multi-access does not take into account that not all nodes on the same Layer 2 link have bi-directional connectivity. However, any transmission on a link will reach nodes that are within transmission range. In this way, the link is multi-access due to the fact that two simultaneous transmissions may collide. A new interface type needs to be defined in order to accurately describe this behavior.

The second category of challenges involves scalability. A MANET must transmit more state information to maintain reachability. Therefore, OSPF will need scalability enhancements to support MANETs. While some flooding optimizations are present in OSPF, such as designated router (DR) election, many of these were built under the assumption of a true multi-access network. Wireless networks are not true multi-access networks, because it cannot be assumed that there is 2-way connectivity between everyone on the same Layer 2 link. Therefore, optimizations such as DR election will not perform correctly in MANET networks. Without any further optimizations in link-state flooding, current OSPF would not be able to operate in a highly dynamic environment in which links are constantly being formed and broken. The amount of information that would need to be flooded would overload the network.

Another scalability issue is the periodic transmission of Hello messages. Currently, even if there are no changes in a router’s neighbor list, the Hello messages still list all the neighbors on a particular link. For a MANET router, where saving bandwidth and transmission power is a critical issue, the transmission of potentially large Hello messages is particularly wasteful.

Finally, current routing protocols will form a neighbor relationship with any router on a Layer 2 link that is correctly configured. For MANET routers in a wireless network, this may lead to an excessive number of parallel links between two routers if communication is achieved via multiple interfaces. In a statically configured network, this is not a problem, since the physical topology can be built to prevent excessive redundancy. However, in a dynamic network, there must exist additional mechanisms to prevent too many redundant links. (Note that links between two nodes on different radio types, different antennae, different channels, etc., are considered different links and not redundant links.) In scalability tests, it has been demonstrated that the presence of too many redundant links will both increase the size of routing updates and cause extra flooding, resulting in even relatively small networks not converging.

3.1. OSPF-MANET Interface

Interfaces are defined as the connection between a router and one of its attached networks [OSPF]. Four types of interfaces have been defined and supported in [OSPF] and [OSPFv3]: broadcast, Non-Broadcast Multi-Access (NBMA), point-to-point, and point-to-multipoint.

The point-to-multipoint model has been chosen to represent MANET interfaces. (The features designed in this document MAY be included on other interface types as appropriate.) The MANET interface allows the following:

- OSPF treats all router-to-router connections over the MANET interface as if they were point-to-point links.
- Link metric can be set on a per-neighbor basis.
- Broadcast and multicast can be accomplished through Layer 2 broadcast or Layer 2 pseudo-broadcast.
    - The MANET interface supports Layer 2 broadcast if it is able to address a single physical message to all of the attached neighbors. One such example is 802.11.
    - The MANET interface supports Layer 2 pseudo-broadcast if it is able to pick up a packet from the broadcast queue, replicate the packet, and send a copy over each point-to-point link. One such example is Frame Relay.
- An API must be provided for Layer 3 to determine the Layer 2 broadcast capability. Based on the return of the API, OSPF classifies the MANET interfaces into the following three types: MANET broadcast, MANET pseudo-broadcast, and MANET non-broadcast.
- Multicast SHOULD be used for OSPF packets. When the MANET interface supports Layer 2 broadcast or pseudo-broadcast, the multicast process is transparent to OSPF. Otherwise, OSPF MUST replicate multicast packets by itself.

3.1.1. Interface Operation

A MANET node has at least one MANET interface. MANET nodes can communicate with each other through MANET interfaces. MANET nodes can communicate with non-MANET routers only through normal interfaces, such as Ethernet, ATM, etc.

For scalability reasons, it is not required to configure IPv6 global unicast addresses on MANET interfaces. Instead, a management loopback interface with an IPv6 global unicast address MAY be configured on each MANET node.

The link state advertisements (LSAs) associated with a MANET interface SHOULD have the DC-bit set in the OSPFv3 Options Field and the DoNotAge bit set in the LS Age field as described in [OSPFv3]. Demand Circuits are an optional feature; hence, the DC-bit setting recommendation level is SHOULD.

3.1.2. LSA Formats and Examples

LSA formats are specified in [OSPFv3].

In order to display example LSAs, a network map is included below. Router names are prefixed with the letters RT, network names with the letter N, and router interface names with the letter I.

- Four MANET nodes, RT1, RT2, RT3, and RT4, reside in area 2.

- RT1 has one MANET interface, I11. Through the interface, RT1 is full-adjacent to RT2, RT3, and RT4.

- RT2 has two MANET interfaces, I21 and I22, and one Ethernet interface, I23. RT2 is full-adjacent to RT1 and RT4 through the interface I21, and full-adjacent to RT4 through the interface I22. Stub network N1 is attached with RT2 through the interface I23.

- RT3 has one MANET interface, I31, and is full-adjacent to RT1 through the interface.

- RT4 has two MANET interfaces, I41 and I42. It is full-adjacent to RT2 through the interface I41, and full-adjacent to RT1 and RT2 through the interface I42.

- Moreover, each MANET node is configured with a management loopback interface.

```
   +---+I11        I21+---+I23   |
   |RT1|-+----------+-|RT2|------|N1
   +---+ |          | +---+      |
   |                |   VI22
   |                |   +
   |                |   |
   |                |   |
   |                |   |
   |                |   |
   |                |   +
   |                |   ^I41
   +---+ |          +---+
   |RT3|-+        +-|RT4|
   +---+I31      I42+---+
```

The assignment of IPv6 global unicast prefixes to network links is shown below. (Note: No IPv6 global unicast addresses are configured on the MANET interfaces).

RT1	LOOPBACK	2001:DB8:0001::/64
	I11	n/a
RT2	LOOPBACK	2001:DB8:0002::/64
	I21	n/a
	I22	n/a
	I23	2001:DB8:0012::/60
RT3	LOOPBACK	2001:DB8:0003::/64
	I31	n/a
RT4	LOOPBACK	2001:DB8:0004::/64
	I41	n/a
	I42	n/a

The OSPF interface IDs and the link-local addresses for the router interfaces in the network are shown below. EUIxy represents the 64-bit interface identifier of the interface Ixy, in Modified EUI-64 format [IPV6ADD].

Node	Interface	Interface ID	Link-Local address
RT1	LOOPBACK	1	n/a
	I11	2	fe80:0002::EUI11
RT2	LOOPBACK	1	n/a
	I21	2	fe80:0002::EUI21
	I22	3	fe80:0003::EUI22
	I23	4	fe80:0004::EUI23
RT3	LOOPBACK	1	n/a
	I31	2	fe80:0002::EUI31
RT4	LOOPBACK	1	n/a
	I41	2	fe80:0002::EUI41
	I42	3	fe80:0003::EUI42

3.1.2.1. Router-LSAs

As an example, consider the router-LSAs that node RT2 would originate. Two MANET interfaces, consisting of 3 point-to-point links, are presented.

```
RT2’s router-LSA

LS age = DoNotAge+0               ;newly originated
LS type = 0x2001                  ;router-LSA
Link State ID = 0                 ;first fragment
Advertising Router = 192.0.2.2    ;RT2’s Router ID
bit E = 0                         ;not an AS boundary router
bit B = 0                         ;not an area border router
Options = (V6-bit|E-bit|R-bit)
 Type = 1                         ;p2p link to RT1 over I21
 Metric = 10                      ;cost to RT1
 Interface ID = 2                 ;Interface ID of I21
 Neighbor Interface ID = 2        ;Interface ID of I11
 Neighbor Router ID = 192.0.2.1   ;RT1’s Router ID
 Type = 1                         ;p2p link to RT4 over I21
 Metric = 25                      ;cost to RT4
 Interface ID = 2                 ;Interface ID of I21
 Neighbor Interface ID = 3        ;Interface ID of I42
 Neighbor Router ID = 192.0.2.4   ;RT4’s Router ID
 Type = 1                         ;p2p link to RT4 over I22
 Metric = 15                      ;cost to RT4
 Interface ID = 3                 ;Interface ID of I22
 Neighbor Interface ID = 2        ;Interface ID of I41
 Neighbor Router ID = 192.0.2.4   ;RT4’s Router ID
```

3.1.2.2. Link-LSAs

A MANET node originates a separate link-LSA for each attached interface. As an example, consider the link-LSA that RT3 will build for its MANET interface I31.

```
RT3's link-LSA for MANET interface I31

LS age = DoNotAge+0              ;newly originated
LS type = 0x0008                 ;link-LSA
Link State ID = 2                ;Interface ID of I31
Advertising Router = 192.0.2.3   ;RT3's Router ID
Rtr Pri = 1                      ;default priority
Options = (V6-bit|E-bit|R-bit)
Link-local Interface Address = fe80:0002::EUI31
# prefixes = 0                   ;no global unicast address
```

3.1.2.3. Intra-Area-Prefix-LSAs

A MANET node originates an intra-area-prefix-LSA to advertise its own prefixes and those of its attached stub links. As an example, consider the intra-area-prefix-LSA that RT2 will build.

```
RT2's intra-area-prefix-LSA for its own prefixes

LS age = DoNotAge+0              ;newly originated
LS type = 0x2009                 ;intra-area-prefix-LSA
Link State ID = 177              ;or something else
Advertising Router = 192.0.2.2   ;RT2's Router ID
# prefixes = 2
Referenced LS type = 0x2001      ;router-LSA reference
Referenced Link State ID = 0     ;always 0 for router-LSA
                                 ;reference
Referenced Advertising Router = 192.0.2.2
                                 ;RT2's Router ID
 PrefixLength = 64               ;prefix on RT2's LOOPBACK
 PrefixOptions = 0
 Metric = 0                      ;cost of RT2's LOOPBACK
 Address Prefix = 2001:DB8:0002::
 PrefixLength = 60               ;prefix on I23
 PrefixOptions = 0
 Metric = 10                     ;cost of I23
 Address Prefix = 2001:DB8:0012::
```

Note: MANET nodes may originate intra-area-prefix-LSAs for attached transit (broadcast/NBMA) networks. This is normal behavior (defined in [OSPFv3]), which is irrelevant to MANET interfaces. Please consult [OSPFv3] for details.

3.2. Incremental OSPF-MANET Hellos

In MANETs, reducing the size of periodically transmitted packets can be very important in decreasing the total amount of overhead associated with routing. Towards this end, removing the list of neighbors from Hello packets, unless that information changes, can reduce routing protocol overhead. While the reduction for each Hello packet is small, over time it will be significant.

A new option bit is defined in this document to facilitate the operation of incremental Hello packets. A new State Check Sequence TLV (SCS TLV) and Neighbor Drop TLV are also defined, transmitted using LLS [LLS].

3.2.1. The I Option Bit

A new I-bit is defined in the LLS Type 1 Extended Options and Flags field. The bit is defined for Hello packets and indicates that only incremental information is present. See Section 5 for placement of the I-bit.

3.2.2. State Check Sequence TLV (SCS TLV)

A new TLV is defined that indicates the current state, which is represented by a State Check Sequence (SCS) number of the transmitting router.

```
 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                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+--+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         SCS Number            |R|FS|N |   Reserved              |
+-----------------------------------------------------------------+
```

- Type: 6

- Length: Set to 4.

- SCS Number: A circular two-octet unsigned integer indicating the current state of the transmitting device. Note that when the incremental Hello mechanism is invoked (or re-started), an initial SCS value of ’1’ SHOULD be used for the first incremental Hello packet. This sequence number is referred to as InitialSCS. Note that InitialSCS also implies a full state.

- R: Request bit. If set, this is a request for current state. The list of routers that should respond to this request is indicated in the Request From TLV (RF TLV) (defined below). If the RF TLV is not present, it is assumed that the request is meant for all nodes.

- FS: Full State bit. If set, the Hello packet contains full state as far as the neighbor(s) in the Full State For TLV (FSF TLV) (defined below) are concerned. If the FSF TLV is not present, the Hello packet contains full state for all neighbors.

- N: Incomplete bit. If NOT set, the complete state associated with the SCS number is included in the Hello packet. If set, this indicates that the appended TLVs are being sent 'persistently', and that there is more state associated with the SCS number that was sent originally, but is not included in this Hello packet. This bit allows any desired TLVs to be sent 'persistently' for a number of Hellos with the same SCS number without requiring all of the TLVs associated with that SCS number to be transmitted. The first time an SCS number is sent, the entire state associated with that SCS number is transmitted, and the N-bit MUST NOT be set.

- Reserved: Set to 0. Reserved for future use.

A Hello with the SCS TLV appended and with the R-bit set will be referred to as a Hello request.

3.2.3. Neighbor Drop TLV

A new TLV is defined in this document that indicates neighbor(s) that have been removed from the list of known neighbors.

```
 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Dropped Neighbor(s)                                           |
+---------------------------------------------------------------+
| ....
+--------------------
```

- Type: 7
- Length: Set to the number of dropped neighbors included in the TLV multiplied by 4.

- Dropped Neighbor(s) - Router ID of the neighbor being dropped.

3.2.4. Request From TLV (RF TLV)

A new TLV is defined in this document that indicates neighbor(s) from which the latest Hello state is being requested.

```
 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              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Request From Neighbor(s)                  |
+---------------------------------------------------------------+
| ....
+--------------------
```

- o Type: 8
- o Length: Set to the number of neighbors included in the TLV multiplied by 4.
- o Request From Neighbor(s) - Router ID of the neighbor(s) from which Hello state is being requested.

3.2.5. Full State For TLV (FSF TLV)

A new TLV is defined in this document that indicates neighbor(s) to which the transmitting node is responding with full state.

```
 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              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Full State For Neighbor(s)                |
+---------------------------------------------------------------+
| ....
+--------------------
```

- o Type: 9
- o Length: Set to the number of neighbors included in the TLV multiplied by 4.
- o Full State For Neighbor(s) - Router ID of the neighbor(s) should process this packet.

3.2.6. Neighbor Adjacencies

This section describes building neighbor adjacencies and the failure of such adjacencies using the incremental Hello signaling.

3.2.6.1. Building Neighbor Adjacencies

Hello packets are sent periodically in accordance with [OSPF] and [OSPFv3]. An OSPF implementation that supports sending only partial neighbor information in Hello packets SHOULD always set the I-bit in its transmitted Hello packets, except as described elsewhere in this document. Hello packets MAY be suppressed from being transmitted every HelloInterval if other packet transmissions are sent by the router during that time.

On receiving a Hello packet from a new neighbor (in this context, a new neighbor is a neighbor in less than Init state as defined in Section 10.1 [OSPF]), if the Hello has the I-bit set, a router will:

- Place the new neighbor in the neighbor list described in [OSPFv3], Appendix A.3.2.

- Increment the router’s SCS number that it will use in its next Hello (indicated in the SCS TLV).

- Remove the neighbor from the neighbor list described in [OSPFv3], Appendix A.3.2, when the neighbor has reached the Exchange state (as described in [OSPF], Section 10.1).

- Remove the neighbor from the neighbor list described in [OSPFv3], Appendix A.3.2, if the neighbor is not a DR or backup designated router (BDR) on an OSPF broadcast link, and if the neighbor is advertised as connected in the network-LSA advertised by the DR.

3.2.6.2. Adjacency Failure

On discovering an adjacency failure (going to state less than Exchange), a router using I-bit signaling SHOULD:

- Remove the adjacent router from local tables, and take the appropriate actions for a failed adjacency described in [OSPF] and [OSPFv3].

- Add the formerly adjacent router to a Neighbor Drop TLV.

- Increment the router’s SCS number that it will transmit in its next Hello.

- Transmit Hellos with this Neighbor Drop TLV. It may be desirable to send the Neighbor Drop TLV in three consecutive Hellos to increase the probability of reception. In this case, 'persistent' Hello packets would be sent with the same SCS number, the Neighbor Drop TLV, and the N-bit set. Thus, the receiver knows that the Neighbor Drop TLV is being sent persistently, and there is more state associated with the SCS in case it must request missing state presumably transmitted in a previous Hello.

3.2.7. Sending Hellos

When a device is first attached to a network (whether by being brought within range of another device, powering the device on, enabling the device's radio interface, etc.), it will need to obtain complete neighbor state from each of its neighbors before it can utilize the incremental Hello mechanism. Thus, upon initialization, a device MAY send a multicast Hello request (and omit the Request From TLV). Neighbors will receive the request and respond with a Hello with their complete neighbor state.

If a device is in INIT state with a neighbor and receives a Hello from the neighbor without its router ID listed in the neighbor list, the device SHOULD request the current state from the neighbor. Note that this is to avoid a "race" condition, since the received Hello can either mean that the device is NOT SEEN by the neighbor, or that the device is adjacent and not listed in the incremental list. Thus, by receiving a Hello request, the neighbor will respond with its neighbor state for the neighbor.

The first Hello packet with a particular SCS number MUST contain the full state associated with that SCS number, i.e., all state changes since the last SCS number. The N-bit MUST NOT be set in the State Check Sequence TLV.

Incremental Hello packets can be sent persistently (sent in k successive Hello packets), with flexibility in the actual amount of information being sent. The three options include:

- The entire incremental Hello packet is sent persistently. This is accomplished by simply sending the entire state associated with a SCS number for k successive Hellos. Since the SCS number remains the same, the N-bit is not set in these incremental Hello packets.

- Partial information for a particular SCS number is sent persistently. After the first Hello packet with a particular SCS number is sent, only the TLVs that are desired to be sent persistently are sent in subsequent Hellos with the same SCS number and the N-bit set.

- No information is sent persistently. This is simply the default behavior where an incremental Hello packet with a particular SCS number is only sent once.

3.2.8. Receiving Hellos

Each OSPF device supporting incremental Hello signaling, as described in this document, MUST keep the last known SCS number from each neighbor it has received Hellos from as long as the neighbor adjacency structure is maintained.

If a device receives a Hello from an adjacent neighbor with an SCS number less than the last known SCS number from that neighbor, it MUST first check if the SCS number is a wrap around. "Wrap around" is a condition when the last known SCS number is MAX_SCS (65535) and the new SCS number is 1. If it is not a wrap around, then the device MUST send a Hello request to the neighbor.

If it is a wrap around, or if a device receives a Hello from an adjacent neighbor with an SCS number one greater than the last known SCS number from that neighbor, it MUST:

- Examine the neighbor list described in [OSPFv3], Appendix A.3.2. If any neighbors are contained in this list, increment the SCS number contained in the adjacent neighbor’s data structure.

- Examine the Neighbor Drop TLV as described in Section 3.2.6.2. If this list contains a neighbor other than the local router, increment the SCS number contained in the adjacent neighbor’s data structure.

- Examine the Neighbor Drop TLV as described in Section 3.2.6.2. If the local router identifier is contained in this list, destroy the transmitting adjacent neighbor’s data structures.

- Examine any other TLVs incrementally signaled, as described in documents referring to this RFC. If there are other state changes indicated, increment the SCS number contained in the adjacent neighbor’s data structure.

- If no state change information is contained in the received Hello, send a request for current state (by setting the ’R’-bit) in the next Hello.

If a device receives a Hello from an adjacent neighbor with an SCS number greater than the last known SCS number + 1 from that neighbor, it MUST send a Hello request to the neighbor, since it may be missing some neighbor state.

3.2.8.1. Receiving Hellos with the N-bit Set

If a device receives a Hello with the SCS TLV included and the N-bit set in this TLV, it MUST verify that it has already received the SCS number with the N-bit NOT set from the neighbor. If the device determines that this is the first receipt of the SCS number from this neighbor, then it MUST send a Hello request to the neighbor, since it missed the initial Hello packet with the SCS number and thus is missing state.

3.2.8.2. Receiving Hellos with the R-bit Set

If a device receives a Hello with the SCS TLV included and the R-bit set, it looks for the RF TLV. If its router ID is listed in the RF TLV or the TLV is not found, it includes its full state in the next Hello. This MUST include:

- The neighbor ID of the requesting neighbor(s) in the list of neighbors described in [OSPFv3], Appendix A.3.2.

- An SCS TLV with the transmitter’s current SCS number and the FS-bit set. Note that the transmitter’s SCS number is NOT incremented.

- Any other TLVs, defined in other documents referencing this RFC, indicating the current state of the local system.

- The neighbor ID of all the neighbors who have requested current state, in the FSF TLV.

If the full state is being sent to a large number of existing neighbors, an implementation could choose to instead generate a full state for all neighbors and omit the FSF TLV.

3.2.8.3. Receiving Hellos with the FS-bit Set

When a device receives a Hello with the SCS TLV included and the FS-bit set, the Hello packet contains the neighbor’s full state for the device. The packet SHOULD be processed as follows:

- If the received SCS number is equal to the last known SCS number, the packet SHOULD be ignored, since the device already has the latest state information.

- If the received SCS number is different than the last known SCS number, this Hello has new information and MUST be parsed.

- If it is listed in the FSF TLV, or if the FSF TLV is not present, the device MUST save the SCS number, process the Hello as described in Section 3.2.8, and process any other appended TLVs.

3.2.9. Interoperability

On receiving a Hello packet from a new neighbor without the I-bit set, the local router will continue to place that router's identifier in transmitted Hellos on this link as described in [OSPFv3], Appendix A.3.2.

3.2.10. Support for OSPF Graceful Restart

OSPF graceful restart, as described in [OSPFREST] and [OSPFGR], relies on the lack of neighbors in the list of neighbors described in [OSPFv3], Appendix A.3.2, to determine that an adjacent router has restarted, and other signaling to determine that the adjacency should not be torn down. If all Hello packets transmitted by a given router have an empty Hello list, reliance on an empty Hello packet to signal a restart (or to reliably tear down an OSPF adjacency) is no longer possible. Hence, this signaling must be slightly altered. When a router would like to tear down all adjacencies, or signal that it has restarted:

- On initially restarting, during the first RouterDeadInterval after restart, the router will transmit Hello packets with an empty neighbor list and the I-bit cleared. Any normal restart or other signaling may be included in these initial Hello packets.

- As adjacencies are learned, these newly learned adjacent routers are included in the multicast Hellos transmitted on the link.

- After one RouterDeadInterval has passed, the incremental Hello mechanism is invoked. An incremental Hello packet with full state is sent with the I-bit set, the SCS TLV included with the FS-bit set, and the InitialSCS value (e.g., SCS of '1'). Subsequent Hello packets will include only incremental state.

Routers that are neighboring with a restarting router MUST continue sending their Hello packets with the I-bit set.

3.3. Optimized Flooding (Overlapping Relays)

A component that may influence the scalability and convergence characteristics of OSPF ([OSPF], [OSPFv3]) in a MANET environment is how much information needs to be flooded. The ideal solution is that a router will receive a particular routing update only once. However, there must be a trade-off between protocol complexity and ensuring that every speaker in the network receives all of the information. Note that a speaker refers to any node in the network that is running the routing protocol and transmitting routing updates and Hello messages.

Controlling the amount of information on the link has increased importance in a MANET environment due to the potential transmission costs and resource availability in general.

In some environments, a group of speakers that share the same logical segment may not be directly visible to each other; some of the possible causes are the following: low signal strength, long distance separation, environmental disruptions, partial VC (virtual circuit) meshing, etc. In these networks, a logical segment refers to the local flooding domain dynamically determined by transmission radius. In these situations, some speakers (the ones not able to directly reach the sender) may never be able to synchronize their databases. To solve the synchronization issues encountered in these environments, a mechanism is needed through which all the nodes on the same logical segment can receive the routing information, regardless of the state of their adjacency to the source.

3.3.1. Operation Overview

The optimized flooding operation relies on the ability of a speaker to advertise all of its locally connected neighbors. In OSPF, this ability is realized through the use of link state advertisements (LSA)s ([OSPF], [OSPFv3]).

A speaker receives router-LSAs from its adjacent neighbors. A speaker’s router-LSA conveys the list of the adjacent speakers of the originator ("neighbor list"). The local speaker can compare the neighbor list reported by each speaker to its own neighbor list. If the local neighbor list contains adjacent speakers that the originator cannot reach directly (i.e., those speakers that are not in the originator’s neighbor list), then these speakers are locally known as non-overlapping neighbors for the originator.

The local speaker should relay any routing information to non-overlapping neighbors of the sender based on the algorithm outlined in Section 3.3.8. Because more than one such speaker may exist, the mechanism is called "overlapping relays". The algorithm, however, does select the set of overlapping relays that should transmit first. This set is known as the active set of overlapping relays for a speaker.

3.3.2. Determination of Overlapping Relays

The first step in the process is for each speaker to build and propagate their neighbor lists in router-LSA packets. Every speaker is then in a position to determine their 2-hop neighborhood, i.e., those nodes that are neighbors of the speaker's 1-hop neighbors.

A bidirectional neighbor is considered an overlapping relay for a speaker if it can reach a node in the 2-hop neighborhood of the speaker, i.e., if it has 1-hop neighbors (excluding the speaker itself).

The set of Active Overlapping Relays for a speaker is the minimum set of direct neighbors such that every node in the 2-hop neighborhood of the speaker is a neighbor of at least one overlapping relay in the active set.

Each speaker SHOULD select a set of Active Overlapping Relays based on a selection algorithm (one such algorithm is suggested in Section 3.3.4 and is based on the multipoint relay (MPR) selection algorithm described in [OLSR]). The behavior of the overlapping relays MUST follow that specified in Section 3.3.8.

Note that a speaker MUST NOT choose a neighbor to serve as an Active Overlapping Relay if that neighbor set the N-bit in its Active Overlapping Relay TLV as defined in Section 3.3.6, unless the neighbor is the only neighbor to reach a 2-hop neighbor.

Election of Active Overlapping Relays is done across interfaces, and thus, it is node-based and not link-based.

3.3.3. Terminology

The following heuristic and terminology for Active Overlapping Relay selection is largely taken from [OLSR]:

- FULL: Neighbor state FULL as defined in [OSPF] and [OSPFv3]. Note that all neighbor references in this document are assumed to be FULL neighbors.

- N: N is the set of FULL neighbors of the node.

- 2-hop FULL neighbors (N2): The list of 2-hop neighbors of the node that are FULL and that can be reached from direct neighbors, excluding any directly connected neighbors.

- Active Set: A (sub)set of the neighbors selected, such that through these selected nodes, all 2-hop FULL neighbors are reachable.

- D(y): The degree of a 1-hop neighbor node y (where y is a member of N) is defined as the number of FULL neighbors of node y, EXCLUDING all the members of N and EXCLUDING the node performing the computation.

3.3.4. Overlapping Relay Discovery Process

A possible algorithm for discovering overlapping relays is the following:

1. Start with an active set made of all members of N that have set the A-bit in their Active Overlapping Relay TLV (AOR TLV) as defined in Section 3.3.6.

2. Calculate D(y), where y is a member of N, for all nodes in N.

3. Add to the active set those nodes in N, which are the *only* nodes to provide reachability to a node in N2, i.e., if node b in N2 can be reached only through a symmetric link to node a in N, then add node a to the active set. Remove the nodes from N2 that are now covered by a node in the active set.

4. While there exist nodes in N2 that are not covered by at least one node in the active set:

   A. For each node in N, calculate the reachability, i.e., the number of nodes in N2 that are not yet covered by at least one node in the active set and that are reachable through this 1-hop neighbor.

   B. Select as an Active Overlapping Relay the node with the highest Willingness value (Section 3.3.7) among the nodes in N with non-zero reachability. In the case of multiple choices, select the node that provides reachability to the maximum number of nodes in N2. In the case of multiple nodes providing the same amount of reachability, select as active the node whose D(y) is greater. As a final tie breaker, the node with the highest router ID should be chosen. Remove the nodes from N2 that are now covered by a node in the active set.

5. As an optimization, process each node, y, in the active set in increasing order of Willingness value. If all nodes in N2 are still covered by at least one node in the active set, excluding node y, and if Willingness of node y is smaller than MAX_WILLINGNESS, then node y should be removed from the active set.

3.3.5. The F Option Bit

A single new option bit, the F-bit, is defined in the LLS Type 1 Extended Options and Flags field. The F-bit indicates that the node supports the optimized flooding mechanism as specified in this document. See Section 5 for placement of the F-bit.

3.3.6. Active Overlapping Relay TLV (AOR TLV)

A new TLV is defined so that each speaker can convey its set of Active Overlapping Relays in the Hello messages. The TLV is transmitted using LLS [LLS].

```
 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                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Relays Added |A|N|          Reserved                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Router ID(s) of Active Overlapping Relay(s)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Type: 10

- Length - variable. Length of TLV in bytes, NOT including Type and Length.

- Relays Added - variable. Number of Active Overlapping Relays that are being added. Note that the number of Active Overlapping Relays that are being dropped is then given by [(Length - 4)/4 - Relays Added].

- A-bit - If this bit is set, the node is specifying that it will always flood routing updates that it receives, regardless of whether it is selected as an Active Overlapping Relay.

- N-bit - If this bit is set, the node is specifying that it most likely will not flood routing updates. The node SHOULD NOT be chosen to be an Active Overlapping Relay unless it is the *only* neighbor that can reach 2-hop neighbor(s). Note that if the node is selected as an Active Overlapping Relay and the node cannot perform the required duties, network behavior is not compromised, since it results in the same behavior as if the node was not chosen as an Active Overlapping Relay.

- Reserved - Reserved for future use. MUST be set to zero by the sender, and MUST be ignored upon receipt.

- Router ID(s) of Active Overlapping Relay(s) - The router ID(s) of neighbor(s) that are either chosen to serve as an Active Overlapping Relay or removed from serving as an Active Overlapping Relay. The Active Overlapping Relays that are being added MUST be listed first, and the number of such relays MUST equal Add Length. The remaining listed relays are being dropped as Active Overlapping Relays, and the number of such relays MUST equal [(Length - 4)/4 - Relays Added].

Note that the A-bit and N-bit are independent of any particular selection algorithm to determine the set of Active Overlapping Relays. However, the bits SHOULD be considered as input into the selection algorithm.

If a node is selected as an Active Overlapping Relay and it does not support the Incremental Hello mechanism defined in Section 3.2, then it SHOULD always be included as an Active Overlapping Relay in the TLV. Note that while a node needs to know whether it is an Active Overlapping Relay, it does not necessarily have to know the identities of the other Active Overlapping Relays.

3.3.7. Willingness TLV

A new TLV is defined so that each speaker can convey its willingness to serve as an Active Overlapping Relay in the Hello message. The TLV is transmitted using the LLS [LLS].

```
 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                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Willingness |                       Reserved                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Type: 11

- Length - 4 bytes. It does not include the Type and Length fields.

- Willingness - 1 byte to indicate the willingness of the node to serve as an Active Overlapping Relay for its neighbors.
  - 0: MIN_WILLINGNESS
  - 128: DEFAULT_WILLINGNESS
  - 255: MAX_WILLINGNESS

The TLV is optional and MUST be silently ignored if not understood. If the Willingness TLV is not included in the Hello packet, the Willingness value SHOULD be taken as DEFAULT_WILLINGNESS.

3.3.8. Flooding and Relay Decisions

The decision whether to relay any received LSAs, and when to relay such information, is made depending on the topology and whether the node is part of the set of Active Overlapping Relays.

Upon receiving an LSA from a bi-directional neighbor, a node makes flooding decisions based on the following algorithm:

1. If the node is an Active Overlapping Relay for the adjacent speaker, then the router SHOULD immediately relay any information received from the adjacent speaker.

2. If the node is a non-Active Overlapping Relay for the adjacent speaker, then the router SHOULD wait a specified amount of time (PushbackInterval plus jitter (see Section 3.3.10)) to decide whether to transmit. [Jitter is used to try to avoid several non-Active Overlapping Relays from propagating redundant information.] Note that a node with the N-bit set in the 'Active Overlapping Relays' extension will not be chosen as an Active Overlapping Relay unless it is the only node to provide reachability to a 2-hop neighbor. However, it MUST perform the duties of a non-Active Overlapping Relay as required. Non-Active Overlapping Relays MUST follow the acknowledgment mechanism outlined in Section 3.3.9.

   A. During this time, if the node determines that flooding the LSA will only result in a redundant transmission, the node MUST suppress its transmission. Otherwise, it MUST transmit upon expiration of PushbackInterval plus jitter.

   B. If a non-Active Overlapping Relay hears a re-flood from another node that covers its non-overlapping neighbors before its timer to transmit expires, it SHOULD reset its PushbackInterval plus jitter timer. (Note that the implementation should take care to avoid resetting the PushbackInterval timer based on transmissions from Active Overlapping Relays.) During this time, if the node determines that flooding the update will only result in a redundant transmission, the node MUST suppress its transmission. Otherwise, it MUST transmit upon expiration of PushbackInterval plus jitter.

C. If a non-Active Overlapping Relay hears an old instance of the LSA during this time, it SHOULD ignore the LSA, and it SHOULD NOT send a unicast packet to the neighbor with the most recent LSA as specified in [OSPFv3].

3. For LSAs that are received unicast because of retransmission by the originator, the node MUST determine whether it has already received the LSA from another speaker. If it already has the current instance of the LSA in its database, it MUST do nothing further in terms of flooding the LSA (since it would have taken appropriate action when it initially received the LSA). However, if it does not have the current instance of the LSA in its database, it MUST take action according to the rules above, just as if it received the multicast LSA. The acknowledgment mechanism outlined in Section 3.3.9 MUST be followed, and any timeout mechanism for unicast LSAs MAY be followed.

Note that a node can determine whether further flooding an LSA will only result in a redundant transmission by already having heard link state acknowledgments (ACKs) or floods for the LSA from all of its neighbors.

Due to the dynamic nature of a network, the set of Active Overlapping Relays may not be up to date at the time the relay decision is made or may not be able to perform the flooding duties, e.g., due to poor link quality. The non-Active Overlapping Relays prevent this situation from causing database synchronization issues and, thus, packet loss.

Since the originator of the information, the relay, and the receiver are all in the same dynamically determined local flooding domain, the relay MUST NOT change the routing update information. In general, LSAs SHOULD be sent to a well-known multicast address. In some cases, routing updates MAY be sent using unicast packets.

3.3.9. Intelligent Transmission of Link State Acknowledgments

In order to optimize the bandwidth utilization on the link, a speaker MUST follow these recommendations related to ACK transmissions:

1. All ACKs MUST be sent via multicast.

2. Typically, LSAs are acknowledged by all of the adjacent speakers. In the case of relayed information, the relay MUST only expect

either explicit or implicit acknowledgments from neighbors that have not previously acknowledged this LSA.

3. Because routing updates are sent via multicast, the set of overlapping speakers will usually receive the same update more than once. A speaker SHOULD only acknowledge the first update received on the link.

4. An Active Overlapping Relay SHOULD NOT explicitly acknowledge information that it is relaying. The relayed information will serve as an acknowledgment to the sender. If no information is being relayed, then an explicit ACK MUST be sent.

5. Several ACKs MAY be bundled into a single packet. The wait (AckInterval) before sending one such packet reduces the number of packet transmissions required in acknowledging multiple LSAs.

6. All ACK packets SHOULD reset the RouterDeadInterval at the receiver. If there is no state waiting to be transmitted in a Hello packet at the sender, then the HelloInterval at the sender SHOULD be reset. Note that an ACK serves as a Hello packet with no state change.

7. Any LSA received via unicast MUST be acknowledged. (Note that acknowledgment is via multicast as specified in rule (1) above.)

An ACK received from a non-overlapping neighbor should prevent redundant transmission of the information to it by another overlapping relay.

3.3.10. Important Timers

This section details the timers that were introduced in Sections 3.3.8 and 3.3.9.

- PushbackInterval: The length of time in seconds that a non-Active Overlapping Relay SHOULD wait before further flooding an LSA if needed. This timer MUST be less than 1/2 of the RxmtInterval ([OSPF], [OSPFv3]) minus propagation delays, i.e., (PushbackInterval + propagation delay) < RxmtInterval/2. The PushbackInterval is set by a non-Active Overlapping Relay upon receipt of an LSA.

- AckInterval: After a node determines that it must transmit an ACK and the AckInterval timer is not already set, the node SHOULD set the AckInterval timer. The AckInterval is the length of time in seconds that a node should wait in order to transmit many ACKs in the acknowledgment packet. This wait reduces the number of packet

transmissions required in acknowledging multiple LSAs. The AckInterval MUST be less than the PushbackInterval minus propagation delays, i.e.,
(AckInterval + propagation delay) < PushbackInterval.

3.3.11. Miscellaneous Protocol Considerations

The mechanism described refers to the operation of relays on a common media segment. In other words, an LSA is only relayed out the same interface through which it was received. However, the concept of information relay may be extended to the flooding of all link state advertisements received on any interface (and forwarded on any other interface). OSPF works on the premise that all of the nodes in a flooding domain will receive all of the routing information. Note that one of the important properties is that the routing information is not altered when relayed.

If each speaker advertised all of its adjacent neighbors on all interfaces, then the overlap check would result in the determination of which speakers are adjacent to both speakers. As a result, link state information should only be flooded to non-overlapping neighbors (taking all of the interfaces into account).

The flooding mechanism in OSPF relies on a designated router to guarantee that any new LSA received by one router attached to the broadcast network will be re-flooded properly to all the other routers attached to the broadcast network. Such designated routers must be able to reach all of the other speakers on the same subnet. A designated router SHOULD NOT be elected if overlapping relays are used.

If such designated routers already exist, then the relays MUST be capable of differentiating them and then making the relaying decisions based on the OSPF’s normal operation. As a result, there may be groups of neighbors to which some information should not be relayed. This mode of operation is NOT RECOMMENDED, as it adds to the complexity of the system.

The intent of the overlapping relay mechanism is to optimize flooding of routing control information. However, other information (such as data) may also be relayed in some networks using the same mechanism.

3.3.12. Interoperability

On receiving a Hello packet from a new neighbor without the F-bit set, the local router will assume that the new neighbor will flood normally as described in [OSPFv3]. Thus, the local router SHOULD include the neighbor in its overlapping relay set since the neighbor will flood by default. This will allow the local router to more optimally select its entire overlapping relay set.

If the F-bit is set and the I-bit as defined in Section 3.2 is not set in the neighbor's Hello, and the neighbor is selected as an overlapping relay by the local router, the local router will continue to include the neighbor's identifier in its active relay set.

3.4. New Bits in LLS Type 1 Extended Options and Flags

Two new option bits are defined in the "LLS Type 1 Extended Options and Flags" Field [LLS] as follows:

- I-bit - defined in Section 3.2.1: The I-bit is only defined for Hello packets and indicates that only incremental information is present.

- F-bit - defined in Section 3.3.5: The F-bit indicates that the node supports the optimized flooding mechanism as specified in this document.

3.5. Smart Peering

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 document proposes a technique to reduce the number of adjacencies based on shortest path tree (SPT) reachability information.

3.5.1. Rationale for Smart Peering

In OSPF ([OSPF], [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 (Shortest Path First) 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 trade-off 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.

3.5.2. Previous Related Work

The formation of a FULL adjacency requires discovery (2-way relationship) and database synchronization. To prevent 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 backward compatible.

3.5.3. Smart Peering Solution

Two routers are defined as synchronized when they have identical link state databases. 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 as when normal adjacency formation processes are used. We should always peer with a neighbor if it provides our only path to currently unreachable destinations.

3.5.3.1. 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 that 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, then 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.

3.5.3.2. State Machine

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

```
                        ......................
                        |Receive a Hello     |
                    (1) |from a new potential|
                        |neighbor            |
                        ''''''''''''''''''''''
                                  |
                                  |
                                  |
                        ,''''''''''''''''''''''|
                        |Check to see if there |
                    (2) |is a router-LSA from  |----no--(4)form a
                        |the new potential     |          new
                        |neighbor in the link  |          neighbor
                        |state database, which |
                        |is reachable in SPT   |
                        ''''''''''''''''''''''''
                                  |
                                  |yes
         (3)                      |
      ,'''''''''''''''''''''''''''''''''''''''''''''''''''''''''|
      |                            (3b)........................ |
      |(3a),______________________     |Determine if the      | |
      |    |Determine if the new |     |number of redundant   | |
      |    |link cost is better  |     |paths to the potential| |
      |    |than the current path|     |neighbor is < the     | |
      |    |cost by a configured |     |maximum configured    | |
      |    |amount               |     |value                 | |
      |    '''''''''''''''''''''''     '''''''''''''''''''''''' |
      |                       \             /                   |
      |                   .....\.........../....                |
      |                   |User configurable   |                |
      |                   |selection algorithm |                |
      |                   '''''/'''''''\''''''''                |
      |                       /         \                       |
      '''''''''''''''''''''''/'''''''''''\'''''''''''''''''''''''
                            /             \
                     requirements     requirements
                        met              not met
                        /                    \
                       /                      \
           (4) form a new neighbor      (5) do not become
                                            neighbors
```

3.5.4. Advertising 2-Way Links in Router-LSAs

The technique described in Section 3.5.3 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.

3.5.4.1. 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 3.5.4.2.

- After a 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 the U-bit (see below).

- Database synchronization is postponed:

    - By a configured amount of time -OR-

    - 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 that 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 allows 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 that 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. We would like to be able to detect that the neighbor is reachable via a path that 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, but 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, which 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 that has SPT dependence for its viability. This approach has scalability concerns and is not considered further here.

We can achieve the same results with just ONE additional SPF that 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) Defining LD Options and using a bit in it, as shown below:

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |   LD Options  |          Metric               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Interface ID                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Neighbor Interface ID                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Neighbor Router ID                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Link Description in a Router-LSA

LD Options

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

```
+-+-+-+-+-+-+-+-+
| | | | | | | |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 the OSPF standard [OSPF]. Hence, this solution isn’t considered further.

3.5.4.2. Unsynchronized SPT

Whenever link state changes happen, we need to run ONE additional SPF by ignoring all links with the 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 state to decide if we should form the adjacency immediately or defer it for later.

3.5.4.3. 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.

3.5.4.4. Overlapping Relay (OR) Election Impact

The overlapping relay election algorithm uses the 2-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.

4. Security Considerations

In a MANET, security is both more difficult and important, due to the wireless nature of the medium. Controlling the ability of devices to connect to a MANET at Layer 2 will be relegated to Layer 2 security mechanisms, such as 802.1x, and others. Controlling the ability of attached devices to transmit traffic will require some type of security system (outside the scope of this document) that can authenticate, and provide authorization for, individual members of the routing domain.

Additional security considerations are similar to any MANET protocol extension. The following text is from [MDR]:

As with OSPFv3 [OSPFv3], OSPF-OR can use the IPv6 Authentication Header (AH) [AH] and/or the IPv6 Encapsulation Security Payload (ESP) [ESP] to provide authentication, integrity, and/or confidentiality. The use of AH and ESP for OSPFv3 is described in [OSPFv3-SEC].

Generic threats to routing protocols are described and categorized in [THREATS]. The mechanisms described in [OSPFv3-SEC] provide protection against many of these threats, but not all of them. In particular, as mentioned in [OSPFv3], these mechanisms do not provide protection against compromised, malfunctioning, or misconfigured routers (also called Byzantine routers); this is true for both OSPFv3 and OSPF-OR.

The extension of OSPFv3 to include MANET routers does not introduce any new security threats. However, the use of a wireless medium and lack of infrastructure, inherent with MANET routers, may render some of the attacks described in [THREATS] easier to mount. Depending on the network context, these increased vulnerabilities may increase the need to provide authentication, integrity, and/or confidentiality, as well as anti-replay service.

For example, sniffing of routing information and traffic analysis are easier tasks with wireless routers than with wired routers, since the attacker only needs to be within the radio range of a router. The use of confidentiality (encryption) provides protection against sniffing but not traffic analysis.

Similarly, interference attacks are also easier to mount against MANET routers due to their wireless nature. Such attacks can be mounted even if OSPF packets are protected by authentication and confidentiality, e.g., by transmitting noise or replaying outdated OSPF packets. As discussed below, an anti-replay service (provided by both ESP and AH) can be used to protect against the latter attack.

The following threat actions are also easier with MANET routers: spoofing (assuming the identity of a legitimate router), falsification (sending false routing information), and overloading (sending or triggering an excessive amount of routing updates). These attacks are only possible if authentication is not used, or the attacker takes control of a router or is able to forge legitimacy (e.g., by discovering the cryptographic key).

[OSPFv3-SEC] mandates the use of manual keying when current IPsec protocols are used with OSPFv3. Routers are required to use manually configured keys with the same security association (SA) parameters for both inbound and outbound traffic. For MANET routers, this implies that all routers attached to the same MANET must use the same key for multicasting packets. This is required in order to achieve scalability and feasibility, as explained in [OSPFv3-SEC]. Future specifications can explore the use of automated key management protocols that may be suitable for MANETs.

As discussed in [OSPFv3-SEC], the use of manual keys can increase vulnerability. For example, manual keys are usually long lived, thus giving an attacker more time to discover the keys. In addition, the use of the same key on all routers attached to the same MANET leaves all routers insecure against impersonation attacks if any one of the routers is compromised.

Although [AH] and [ESP] state that implementations of AH and ESP SHOULD NOT provide anti-replay service in conjunction with SAs that are manually keyed, it is important to note that such service is allowed if the sequence number counter at the sender is correctly maintained across local reboots until the key is replaced. Therefore, it may be possible for MANET routers to make use of the anti-replay service provided by AH and ESP.

When an OSPF routing domain includes both MANETs and fixed networks, the frequency of OSPF updates either due to actual topology changes or malfeasance could result in instability in the fixed networks. In situations where this is a concern, it is recommended that the border routers segregate the MANETs from the fixed networks with either separate OSPF areas or, in cases where legacy routers are very sensitive to OSPF update frequency, separate OSPF instances. With separate OSPF areas, the 5-second MinLSInterval will dampen the frequency of changes originated in the MANETs. Additionally, OSPF ranges can be configured to aggregate prefixes for the areas supporting MANETs. With separate OSPF instances, more conservative local policies can be employed to limit the volume of updates emanating from the MANETs.

5. IANA Considerations

IANA has made the assignments as explained below using the policies outlined in [IANA].

- I-bit and F-bit from "LLS Type 1 Extended Options and Flags" registry as defined below:

```
+---+---+---+---+---+---+---+- -+---+---+---+---+---+---+---+---+
| * | * | * | * | * | * | * |...| * | * | * | * | F | I | RS| LR|
+---+---+---+---+---+---+---+- -+---+---+---+---+---+---+---+---+
```

Bits in Extended Options and Flags TLV

- New TLV types from the "Link Local Signalling TLV Identifiers (LLS Types)" registry as defined below:

TLV Name	TLV Type
State Check Sequence TLV	6
Neighbor Drop TLV	7
Request From TLV	8
Full State For TLV	9
Active Overlapping Relay TLV	10
Willingness TLV	11

- A new registry has been defined for LD Options as defined in Section 3.5.4.1. The U-bit is allocated by this document.

  All future additions to LD Options are subject to OSPF WG review and require IETF Review.

6. Contributors

The following persons are contributing authors to the document:

Fred Baker
Cisco Systems
1121 Via Del Rey
Santa Barbara, CA 93117
USA
EMail: fred@cisco.com

Dave Cook
Cisco Systems
7025-4 Kit Creek Road
Research Triangle Park, NC 27709
USA
EMail: dacook@cisco.com

Alvaro Retana
Cisco Systems
7025-4 Kit Creek Road
Research Triangle Park, NC 27709
USA
EMail: aretana@cisco.com

Yi Yang
Cisco Systems
7025-4 Kit Creek Road
Research Triangle Park, NC 27709
USA
EMail: yiya@cisco.com

Russ White
Cisco Systems
7025-4 Kit Creek Road
Research Triangle Park, NC 27709
USA
EMail: riw@cisco.com

7. Acknowledgments

The authors and contributors would like to thank Pratap Pellakuru and Stan Ratliff for their feedback and implementation of the document. Thanks to Richard Ogier and John Avery for doing a final review.

8. References

8.1. Normative References

[OSPF] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[OSPFv3] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

[LLS] Zinin, A., Roy, A., Nguyen, L., Friedman, B., and D. Yeung, "OSPF Link-Local Signaling", RFC 5613, August 2009.

[IANA]	Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
[KEY]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

[IPV6ADD]	Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
[OSPFGR]	Moy, J., Pillay-Esnault, P., and A. Lindem, "Graceful OSPF Restart", RFC 3623, November 2003.
[OSPFREST]	Nguyen, L., Roy, A., and A. Zinin, "OSPF Restart Signaling", RFC 4812, March 2007.
[OOB]	Nguyen, L., Roy, A., and A. Zinin, "OSPF Out-of-Band Link State Database (LSDB) Resynchronization", RFC 4811, March 2007.
[OLSR]	Clausen, T., Ed., and P. Jacquet, Ed., "Optimized Link State Routing Protocol (OLSR)", RFC 3626, October 2003.
[WINTF]	Ahrenholz J., et al., "OSPFv2 Wireless Interface Type", Work in Progress, May 2004.
[MDR]	Ogier, R. and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding", RFC 5614, August 2009.
[AH]	Kent, S., "IP Authentication Header", RFC 4302, December 2005.
[ESP]	Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
[OSPFv3-SEC]	Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, June 2006.
[THREATS]	Barbir, A., Murphy, S., and Y. Yang, "Generic Threats to Routing Protocols", RFC 4593, October 2006.

Authors' Addresses

Abhay Roy (Editor)
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA
EMail: akr@cisco.com

Madhavi W. Chandra (Editor)
113 Holmhurst Court
Cary, NC 27519

EMail: mw.chandra@gmail.com