# Toward IPv6 OLSR

Anis Laouiti, Pascale Minet, Cédric Adjih

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Toward IPv6 OLSR*

Anis Laouiti — Pascale Minet — Cédric Adjih

**N° 4997**

Novembre 2003

THÈME 1


ISSN 0249-6399 ISRN INRIA/RR--4997--FR+ENG

# Toward IPv6 OLSR

Anis Laouiti [*] , Pascale Minet [†] , Cédric Adjih [‡] [§]

Thème 1 — Réseaux et systèmes
Projet HIPERCOM

Rapport de recherche n° 4997 — Novembre 2003 — 17 pages

**Abstract:** The first part of this document describes the features of IPv6; such as the different address formats, the neighbor discovery protocol, and the autoconfiguration procedure. The second part is dedicated to the changes required for OLSR to work, and to benefit from IPv6 mechanisms.

**Key-words:** wireless network, mobile ad hoc networks, MANET, OLSR, IPv6, neighbor discovery, autoconfiguration

[*] Anis.Laouiti@inria.fr
[†] Pascale.Minet@inria.fr
[‡] Cedric.Adjih@inria.fr
[§] This study has been funded by CELAR/MoD in the framework of its Upstream Study Plan MANET.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 —Télécopie : +33 1 39 63 53 30

# Toward IPv6 OLSR

**Résumé :**

Dans ce document, nous décrivons dans un premier temps les fonctionnalités offertes par IPv6 telles que les différents formats d'adressage, le protocole de découverte des voisins, et les mécanismes d'autoconfiguration. La seconde partie de ce document, décrit, les adaptations proposées pour OLSR, afin de profiter des fonctionnalités d'IPv6.

**Mots-clés :** réseaux sans fil, réseaux mobiles ad hoc, MANET, OLSR, IPv6, découverte des voisins, autoconfiguration

# Contents

# 1 Introduction

IPv6 stands for "Internet Protocol Version 6". IPv6 is the new Internet Protocol proposed by the IETF to replace the current Internet Protocol version, commonly known as IPv4.

The lack of addresses was one of the reasons that led to develop IPv6. But IPv6 fixes also a number of problems in IPv4 and improves other functionalities such as routing and network configuration. Hereby some characteristics of IPv6:

- Simplified header
- Fixed size header
- New extension headers
- No more fragmentation in intermediate routers

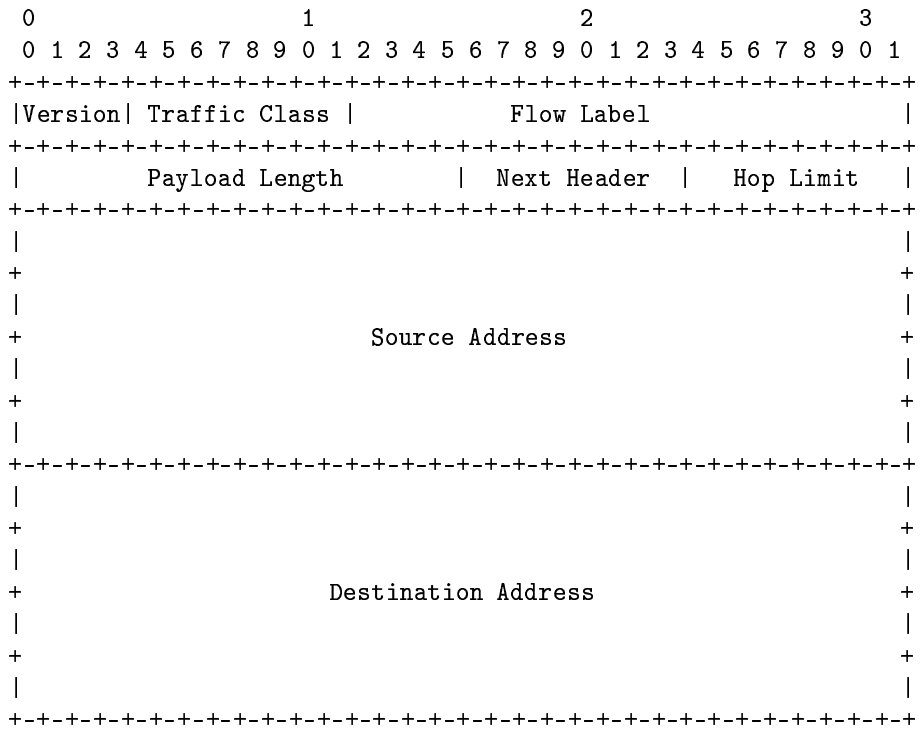
In the following we present a rapid survey over the important principles and features of IPv6 in order to understand how they can affect OLSR. The second part of this document is dedicated to adapting OLSR to IPv6.

# 2 IPv6: Internet Protocol Version 6

In this section we first present the IPv6 packet and address formats. Then, we introduce the neighbor discovery and the IPv6 autoconfiguration mechanisms.

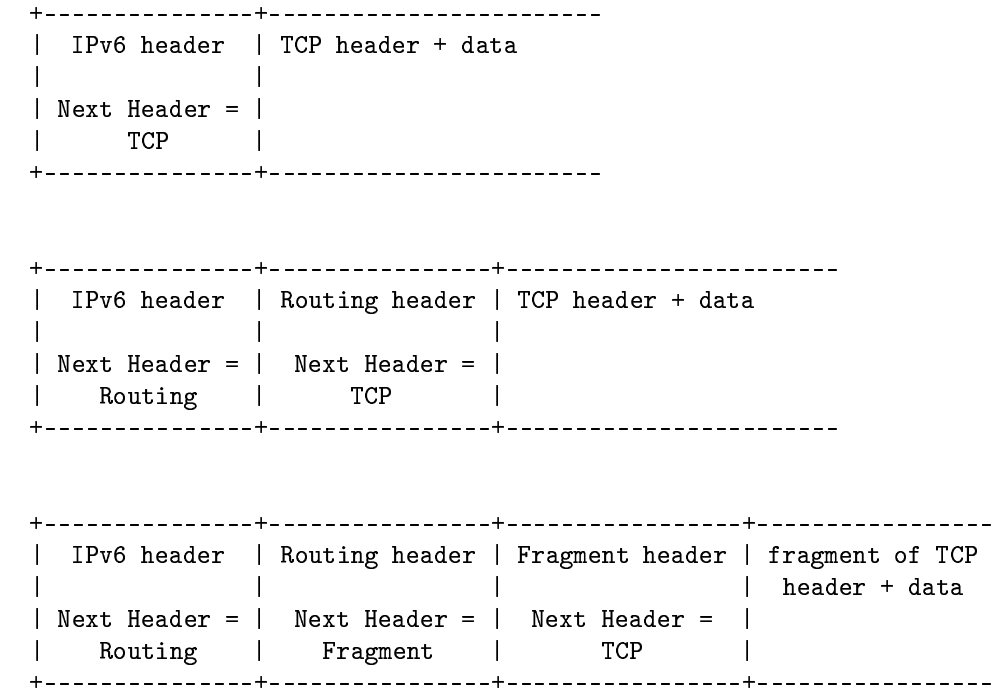
## 2.1 IPv6 Packet format

IPv6 [2] packet is composed of **IPv6 header + set of extensions headers + data**: the IPv6 header has a fixed size, followed by a number of optional headers, and finally the useful data for the upper layers.
We now present the IPv6 header format:




- **Version** 4-bit Internet Protocol version number = 6.
- **Traffic Class** 8-bit traffic class field.
- **Flow Label** 20-bit flow label.
- **Payload Length** 16-bit unsigned integer. Length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets (Note that any extension headers present are considered part of the payload, i.e., included in the length count).
- **Next Header** 8-bit selector. Identifies the type of header immediately following the IPv6 header. Uses the same values as the IPv4 Protocol field.
- **Hop Limit** 8-bit unsigned integer. Decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.
- **Source Address** 128-bit address of the originator of the packet.
- **Destination Address** 128-bit address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing header is present).

In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet. There are a small number of such extension headers, each identified by a distinct Next Header value. As illustrated in these examples, an IPv6 packet may carry zero, one, or more extension headers, each identified by the Next Header field of the preceding header:




A full implementation of IPv6 includes implementation of the following extension headers:

- Hop-by-Hop Options
- Routing
- Fragment
- Destination Options
- Authentication
- Encapsulating Security Payload

The header "hop-by-hop options" is the only header that must be processed by all the intermediate nodes. Note that the headers must be processed strictly in the order.

## 2.2 IPv6 addresses

IPv4 address is 32 bit long, whereas an IPv6 address is 128 bit long. This means, we have many more addresses than IPv4. Note that the IP address is still attached to an interface and not to a machine; therefore, a single interface may have several addresses at the same time.

**Examples:**
Hereby some examples of address presentation

1. IPv4 address 128.93.17.52,
2. IPv6 address FEDC:BA98:7654:3210:FEDC:BA98:7654:3210,
3. IPv6 address FEDC:BA98:7654:3210:FEDC:BA98:7654:3210/64 (64 bits for the network prefix).

## 2.3 IPv6 address types

There are three types of addresses:

- Unicast address refers to a unique interface.
- Anycast address refers to a group of interfaces. A packet sent to an anycast address is delivered to one of the interfaces identified by that address.
- Multicast address refers to a group of interfaces which join a multicast group identified by this address. A packet sent to a multicast address is delivered to all interfaces identified by that address.

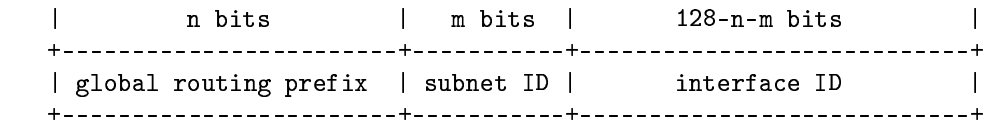
Note that with IPv6 there is no broadcast address like in IPv4 world. Broadcast is achieved by the use of multicast addresses.

## 2.4 Unicast addressing

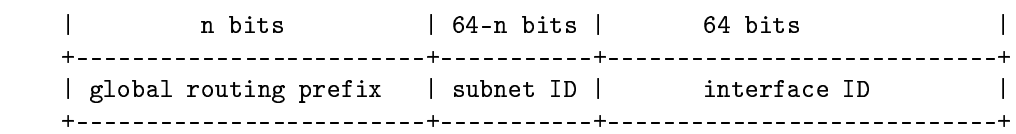
IPv6 unicast addresses [7] are aggregable with prefixes of arbitrary bit-length similar to IPv4 addresses under Classless Interdomain Routing. There are several types of unicast addresses in IPv6, in particular global unicast, site local unicast, and link local unicast. In the following, we first define the global unicast address format, then we present different scopes of a local unicast addresses: link local and site local address. Finally we give some predefined addresses and the way to map IPv4-IPv6 addresses.

### 2.4.1 Global Unicast Addresses

The global unicast address format [8] is as follows:




where the global routing prefix is a (typically hierarchically-structured) value assigned to a site (a cluster of subnets/links), the subnet ID is an identifier of a subnet within the site, and the interface ID is as defined in section 2.5.1 of [7]. The global routing prefix and subnet field are designed to be structured hierarchically.
The rfc [7] also requires that all unicast addresses, except those that start with binary value 000, have Interface IDs that are 64 bits long and to be constructed in Modified EUI-64 format[11].
The format of global unicast address in this case is:



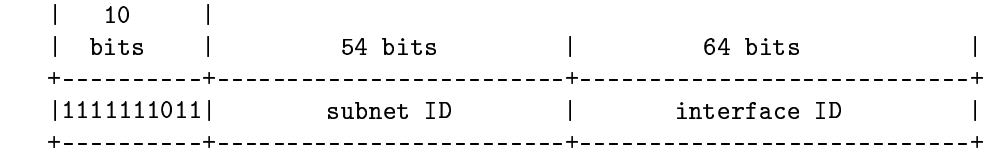

where the routing prefix is a value assigned to identify a site (a cluster of subnets/links), the subnet ID is an identifier of a subnet within the site, and the interface ID.

### 2.4.2 Link local address

Link local addresses [7] are designed to be used for addressing on a single link for purposes such as auto-address configuration, neighbor discovery, or when no routers are present. Link local addresses are obtained by concatenating the prefix FE80::/64 with the interface identifier. This allows any interface to have an IPv6 address easily at start time. Routers must not forward any packets with link local source or destination addresses to other links.

### 2.4.3 Site local address

Site local addresses [7] are designed to be used for addressing inside of a site without the need for a global prefix. Routers must not forward any packets with site-local source or destination addresses outside of the site. Site local addresses are obtained by concatenating the prefix FEC0::/10, a network identifier (up to 54 bits long), and the interface identifier. It is expected that globally-connected sites will use the same subnet IDs for site-local and global prefixes. The site local address format is:




This kind of address is similar to the private addresses in IPv4 (like 10.x.y.z).

### 2.4.4 Other unicast addresses

- unspecified address: it is the address 0:0:0:0:0:0:0:0, that indicates the absence of an address. It is used for the initialization procedure of the node before getting its own address.
- loopback address: it is the address 0:0:0:0:0:0:0:1 (or ::1). Like in IPv4, it is used by a node to send packets to itself. Those packets are never sent outside the node and never forwarded.
- IPv4-compatible IPv6 address: this kind of address has the following format ::a.b.c.d where a.b.c.d is the IPv4 address, and is used for communication between two IPv6 nodes by the mean of a IPv6/IPv4 tunnel. Messages sent to ::a.b.c.d are encapsulated in IPv4 packets, and decapsulated at the destination.
- IPv4-mapped IPv6 address: this kind of address has the following format ::FFFF:a.b.c.d where a.b.c.d is the IPv4 address, and is used to represent the addresses of IPv4-only nodes (those that do not support IPv6) as IPv6 addresses. An IPv6 application server may handle incomming requests from IPv4 machines by mapping their addresses to IPv6 ones. This means that the server uses two IP stacks (v4 and v6) for communication.

## 2.5 Multicast addressing

Multicast addressing [7] assigns an identifer for a group of nodes. A message sent to that group must be delivered to all its nodes. A multicast address can not be assigned to an interface. In this section we introduce the multicast address format, then we give some predefined multicast addresses, and finally we present the solicited multicast addresses.

### 2.5.1 Multicast address format

Multicast addresses have FF00::/8 as a prefix and the following format:

```
|   8    |  4 |  4 |                  112 bits                   |
+------ -+----+----+---------------------------------------------+
|11111111|flgs|scop|                  group ID                   |
+--------+----+----+---------------------------------------------+
```

- flgs: is a set of 4 flags:000T. The three high bits are reserved and must be set to value 0. T indicate whether the address group is permanently-assigned (T=0) or it is a transient one (T=1).
- scop: is a 4-bit multicast scope value used to limit the scope of the multicast group. The values are:
  - 0 reserved

  - 1 node-local scope
  - 2 link-local scope
  - 5 site-local scope
  - 8 organization-local scope
  - E global scope
  - F reserved
- group ID: identifies the multicast group, either permanent or transient, within the given scope.

Multicast addresses must not be used as source addresses in IPv6 packets or appear in any routing header.

### 2.5.2 Some pre-defined multicast addresses

We present some pre-defined multicast addresses [1]:

- the group addresses FF0X:: (where X is in [0..F]) are reserved, and must not be used.
- All IPv6 nodes addresses within:
  - scope 1 (node-local) FF01:0:0:0:0:0:0:1
  - scope 2 (link local) FF02:0:0:0:0:0:0:1
- All IPv6 routers addresses within:
  - scope 1 (node-local) FF01:0:0:0:0:0:0:2
  - scope 2 (link local) FF02:0:0:0:0:0:0:2
  - scope 5 (site-local) FF05:0:0:0:0:0:0:2

### 2.5.3 Solicited multicast addresses

This address is obtained by the concatenation of the prefix FF02:0:0:0:0:1:FF00::/104 and the low-order 24 bits of the IPv6 address (unicast or anycast). This kind of address is used by ICMPv6 [5]. A node which knows the IPv6 address of a local destination (same link), and not the MAC address, may use the solicited multicast address, to ask the destination for. It is similar to the ARP procedure, but, with this technique, only few machines process the request, and not all of them.

## 2.6 Neighbor discovery

Nodes (hosts and routers) use Neighbor Discovery [3] to determine the MAC addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid. Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf. Finally, nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed MAC addresses. When a router or the path to a router fails, a host actively searches for functioning alternates. We note that neighbor discovery does not mean learning information about all other nodes connected to the same link but only those with which the node is willing to communicate.

Neighbor discovery protocol uses five ICMPv6 messages:

- two for router - node communication: Router Solicitation and Router Advertissement messages.
- two for node - node communication: Neighbor Solicitation and Neighbor Advertissement messages.
- A Redirect message to inform hosts of a better first hop for a destination.

Those messages are used by the neighbor discovery protocol to perform different functionalities:

- Address resolution: this function has the same principle as the classical ARP in IPv4. Each node builds a corresponding table between IPv6 and MAC addresses.
- Neighbor unreachability detection : this function is used to delete the stale entries in the neighbor cache table. Neighbor Unreachability detection function is performed only for neighbors to which unicast packets are sent.
- Autoconfiguration: see next section.
- Message redirection : this kind of messages is sent by the router to notify of a better route (next hop) a host from which it is receiving data .

## 2.7 IPv6 Autoconfiguration

This is a new function for IPv6. Traditionally, interface configuration is a matter of "network expert", and done manually. Autoconfiguration eliminates this tedious task. A new machine connected to a local network learns the network parameters from the routers declaration, and configures its interfaces without a human interference.
Autoconfiguration includes the following functionalities:

- Router discovery: with this function, nodes locate routers in the neighborhood (same link).

- Prefix discovery: nodes learn the network prefix by the Router Advertisements. Using this prefix, the node is able to build its address, by adding the interface identifier to the prefix.
- Duplicate Address Detection (DAD): we may have duplicate addresses in the same network, since they are built automatically. This function checks the uniqueness of an IPv6 address in the same link before assigning it to an interface.
- Link parameters discovery: nodes learn different physical link parameters with this functionality. Those parameters may be: the size of the link MTU (Maximum Transmission Unit), maximum number of hops, ....

Address autoconfiguration is done in three steps:

1. creation of link local address.
2. after verification of the uniqueness of the address, this latter is assigned to the interface, otherwise the process is interrupted.
3. determination of the unicast global address.

IPv6 specifies two ways to get the unicast global address:

- stateless autoconfiguration [4]: the configuration is done based on the information advertised by the local routers.
- stateful autoconfiguration [6]: it is based on the Dynamic Host Configuration Protocol for IPv6(DHCPV6). It is recommended when we need strict control of address attribution in the network.

We note that the router specifies in its Router Advertisement which method the node will use for the autoconfiguration.

# 3 IPv6 OLSR

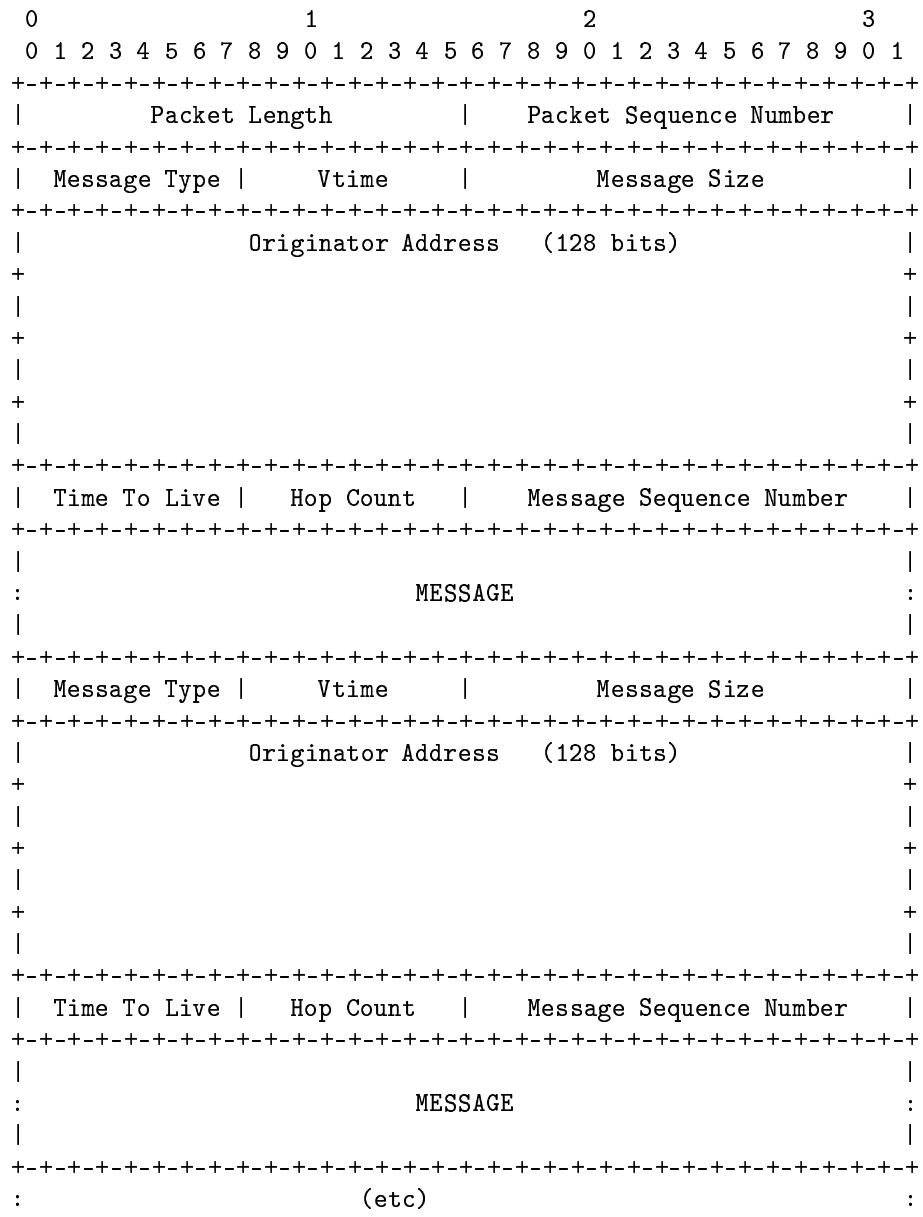
In this second part we present the changes needed to make OLSR work in IPv6 world. First, we describe the changes for OLSR packet format needed for IPv6, and how to make OLSR communicate in IPv6. The second section gives a method to flood non OLSR packets in the network. In the last section we present an autoconfiguration algorithm adapted to OLSR.

## 3.1 Changes to OLSR routing protocol

OLSR packets and algorithms description can be found in [10] [9].

### 3.1.1 OLSR packet format contents

The only change needed for OLSR packets is to replace the IPv4 addresses with the IPv6 addresses in all messages. The OLSR packet header for example will look like:



### 3.1.2 IPv6 ad hoc addressing issues

In the following we first present the addresses that should be used by OLSR to diffuse its control packets. Then we define the solicited multicast group address that each node has to join. Finally we introduce the routable addresses to use and some implementations issues.

**Diffusing OLSR packets**
Basically, OLSR diffuses its control packets to its direct neighbors (nodes within radio reach). Those packets are processed locally, and then, retransmitted if they are destined to the entire network by a subset of neighbors called MPRs (we note, that each node choses its own MPRs nodes, from the set of its neighbors, which cover its two hops neighbors). With this process packets will reach all the nodes in the network.
With IPv4, sending packets to one hop neighbors means that we send them to the broadcast address. In IPv6, we must use a multicast address ALL_LINK_NODES (FF02:0:0:0:0:0:0:1) to reach all the nodes present in the link (here means within radio reach) to have the same result as in IPv4, since, there is no defined broadcast address.
Actually, multicasting packets to the ALL_LINK_NODES is not sufficient to reach all the neighbors, if those nodes did not join this multicast group. Consequently, all the nodes must join this group to receive the flooded packets.

**Solicited multicast address**
All participating nodes must join their corresponding solicited multicast group, in order to be able to reply to the Neighbor Solicitation, and resolve the correspondance between IPv6 and MAC layer addresses.

**Using routable addresses**
As we said in the IPv6 description, link local addresses are not routable. Hence, nodes using such addresses, will not be able to communicate with each other if they are not within radio reach. This kind of network can not be defined as a MANET network.

For MANET networks we should attribute site local addresses for local use, and global unicast addresses when we need to be connected/reached to/from the internet for example.

## 3.2 Diffusing non OLSR packets

MANET are multihop routing networks. In order to flood packets to all the nodes, we usally need retransmissions. With OLSR, packets are retransmitted hop by hop to the direct neighborhood using the MPRs. In the other hand, most of IPv6 messages for neighbor discovery and autoconfiguration for example are multicast only on the local link and they are never routed. This supposes that if we are in the same network, we are on the same link. In other terms, in a multihop network, this kind of messages will not be delivered to all the nodes.
We propose two solutions to diffuse non OLSR packets to all nodes:

1. encapsulate the packets in specific OLSR messages, and use the MPR flooding.

2. use of a new multicast address which we call ALL_MANET_NODES, instead of the ALL_LINK_NODES. All MANET nodes must join this group address to receive the flooding messages. ALL_MANET_NODES address should have a site scope to allow the routing by intermediate nodes. The flooding mechanism in the ad hoc network is based on retransmission of the received packet at most once. This implies also that we need a mechanism to control the process of repetition, in order to avoid useless transmissions. This, can be done, by adding a sequence number in each packet and a duplicate table in each node. A node maintains a duplicate set to prevent transmitting the same packet twice. Unfortunately, IPv6 packets does not contain a sequence number. The idea here is to create a new option for the IPv6 hop_by_hop extension header, to add a packet sequence number and may be the IP address of the intermediate transmitter. We remind here that the IPv6 hop_by_hop header is examined by all the intermediate routers.

## 3.3 Neighbor discovery and autoconfiguration

Routing table in OLSR indicates the next hop for each reachable destination in the network. This next hop is one of the direct neighbors. This means that the neighbor solicitation for address resolution will work without any modification. The node uses link local broadcast, and the destination will reply. With coherent routing tables, the address resolution works correctly.

As we noticed in the section 2.7, the autoconfiguration is based on three steps. After, the creation of a link local address, we must check if the address is already in use by another interface in the network. In wired network, this means that we check all the attached interfaces in the same local link; in MANET network this includes only the interfaces within radio reach of the transmitter and not all the participating nodes. In the following, we propose a new algorithm to perform autoconfiguration in OLSR network. In our algorithm router advertisments are sent to the whole network by using one of the methods described in section 3.2.

1. create a link local address.

2. perform the DAD (Duplicate Addres Detection) procedure in the neighborhood.

3. if the address is unique in the neighborhood, assign it to the interface. The node, considered as a pending node, starts running OLSR by exchanging Hellos. In this intermediate step, the local link address node must not be declared by its neighbors to the entire network, and the node must not be chosen as a MPR.

4. the node designates a neighbor router which is already configured.

5. in this step the designated router performs DAD in the entire network on behalf of the pending node using one of the above methods for diffusing non OLSR packets.

6. if the address is not a duplicate one, the pending node can either:
   (a) wait for Router Advertisement to create its global address. This information is relayed by its designated router.
   (b) or ask its designated router for a network prefix to build its own address.
7. the node informs its designated router of the end of this configuration process,
8. now, it can integrate the network.

If the neighborhood changes before the end of the process, the node has to check that none of the new neighbors has a duplicate address, and designates another router if the previous one has disappeared.

# 4 Conclusions

In this document we presented the important charateristics of IPv6, like the addressing architecture and the neighbor discovery protocol.
In the second part of the document, we described the addresses that must be used to broadcast and receive control packets with OLSR.
OLSR nodes must have routable addresses in order to communicate with each other if they are not within radio range. Those addresses can attributed manually. In such case, the network can work without any problem, but we loose the benefit of the IPv6 autoconfiguration features.
We have shown that the autoconfiguration is not adapated to MANET networks, specifically when we need to flood information to the entire network. That is why we described two methods to flood IPv6 packets to the entire MANET networks. And, finally, we proposed a new algorithm to perform autoconfiguration in OLSR network.

# References

[1] R. Hinden, S. Deering, "IPv6 Multicast Address Assignments", IETF RFC 2375, July 1998.

[2] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6)Specification", IETF RFC 2460, December 1998.

[3] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", IETF RFC 2461, December 1998.

[4] S. Thomson, T. Narten, "IPv6 Stateless Address Autoconfiguration", IETF RFC 2462, December 1998.

[5] A. Conta, S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", IETF RFC 2463, December 1998.

[6] R. Droms, Ed.,J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", IETF RFC 3315, July 2003.

[7] R. Hinden, S. Deering, "IP Version 6 Addressing Architecture", IETF RFC 3513, April 2003.

[8] R. Hinden, S. Deering,E. Nordmark, "IPv6 Global Unicast Address Format", IETF RFC 3587, August 2003.

[9] Philippe Jacquet, Amir Qayyum, Thomas H. Clausen, Anis Laouiti, Laurent Viennot, Pascale Minet and Paul Muhlethaler. "Optimized Link State Routing Protocol". IETF RFC 3626, October 2003.

[10] P. Jacquet, P. Muhletaler, A. Qayyum, A. Laouiti, T. Clausen, L. Viennot, "Optimized Link State Routing Protocol" , IEEE INMIC Pakistan, Dec 2001.

[11] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", http://standards.ieee.org/db/oui/tutorials/EUI64.html, March 1997.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399