Multicast Optimized Link State Routing

Anis Laouiti, Philippe Jacquet, Pascale Minet, Laurent Viennot, Thomas Clausen, Cédric Adjih

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Multicast Optimized Link State Routing

Anis Laouiti — Philippe Jacquet — Pascale Minet — Laurent Viennot

— Thomas Clausen — Cédric Adjih

N°

4721

Février 2003

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--4721--FR+ENG

Multicast Optimized Link State Routing

Anis Laouiti [*] , Philippe Jacquet [†], Pascale Minet [‡] , Laurent Viennot [§] , Thomas Clausen [¶] , Cédric Adjih [‖]

Thème 1 — Réseaux et systèmes
Projet HIPERCOM

Rapport de recherche n°
4721 — Février 2003 — 19 pages

Abstract: This document describes the Multicast extension for the Optimized Link State Routing protocol (MOLSR). MOLSR is in charge of building a multicast structure in order to route multicast traffic in an ad-hoc network. MOLSR is designed for mobile multicast routers, and works in a heterogenous network composed of simple unicast OLSR routers, MOLSR routers and hosts. In the last part of this document we introduce also a Wireless Internet Group Management Protocol (WIGMP). It offers the possibility for OLSR nodes (without multicast capabilities) to join multicast groups and receive multicast data.

Key-words: wireless network, mobile ad hoc networks, MANET, OLSR, multicast, IGMP, multicast routing

[*] Anis.Laouiti@inria.fr
[†] Philippe.Jacquet@inria.fr
[‡] Pascale.Minet@inria.fr
[§] Laurent.Viennot@inria.fr
[¶] Thomas.Clausen@inria.fr
[‖] Cedric.Adjih@inria.fr

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

Multicast Optimized Link State Routing

Résumé : Dans ce document on décrit une extension multicast au protocole de routage OLSR. MOLSR est chargé de la construction de la structure multicast afin de router les trafics point-à-multipoint dans un réseau ad-hoc. MOLSR est conçu pour des routeurs mobiles et supporte aussi un environnement hétérogène composé de simples routeurs point-à-point OLSR, des routeurs MOLSR et des machines sans protocole de routage. Dans la dernière partie de ce document, nous décrivons l'adaptation au monde sans fil du protocole d'abonnement aux groupes multicast (WIGMP). Ce dernier permet aux nœuds OLSR sans capacité de routage point-à-multipoint de joindre des groupes multicast.

Mots-clés : réseaux sans fil, réseaux mobiles ad hoc, MANET, OLSR, multicast, IGMP, routage multipoint

Contents

1 Introduction

OLSR is a unicast routing protocol. If multicast traffics exist in an ad-hoc network implementing OLSR, two possibilities exist. First, multicast traffic is flooded over the network by means of multipoint relays[4][5]. Second, multicast traffic is disseminated only to nodes interested in. If few nodes are interested, the first possibility has a high cost in bandwidth and CPU, even though it is optimized by means of multipoint relays. That is why in this report, we deal with the second possibility.

To deliver multicast data to the group destinations, we can use tree-based or mesh-based structures. Trees are built per group with a core (example: AMRoute[9]), or for each tuple (source, multicast group) (example: MOSPF[8]). Mesh structures are built for each multicast group. The mesh structure includes the multicast sources and destinations (example: ODMRP[11], CAMP[10]).

The Multicast Optimized Link State Routing (MOLSR) protocol takes benefit of the topology knowledge gathered by the OLSR protocol with its Topology Control messages exchange to build multicast trees. MOLSR is developed as an extension to OLSR. It works even when not all nodes are multicast capable provided that multicast nodes offer the minimal connectivity between the sources and the members of the multicast group. A multicast tree is built and maintained for any tuple (source, multicast group) in a distributed manner without any central entity and provides shortest routes from the source to the multicast group members. The trees are updated whenever a topology change is detected.

Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [1]. The terms "connection", "holding time" "multipoint relay", "MPR", "multipoint relay selector", "MPR selector", "node" and "symmetric link" as well as other terms relating to the functionality of OLSR are to be interpreted as described in draft-ietf-olsr-07.txt [3].
Additionally, the following terms are used throughout this document:

Multicast router
A node with multicast capabilities which can be used as router to build multicast trees. This node supports MOLSR extension.

Designated multicast router
A multicast router which has been designated to forward multicast data to a host in its neighborhood.

Multicast source
A host or a router which wants to send data to a multicast group.

Multicast group member
A node which wants to receive data sent to a multicast group.

Participant
A node which belongs to a multicast tree. It may be a member of the group. If this node is not a multicast group member, its presence is needed to forward multicast data to a multicast group member.

Multicast tree
A tree whose root is the source of the multicast data and whose nodes are multicast routers needed to forward data to all multicast group members. There is one multicast tree per tuple (source, multicast group).

Parent
A parent of a node is a multicast router that can relay multicast packets coming from the source to that node. A parent has one or more downstream links (one per son) in the associated multicast tree.

Son
A node which has an upstream link to its parent in the associated multicast tree.

2 Protocol Overview

The multicast optimized link state routing protocol (MOLSR) belongs to the source based family. It maintains one multicast tree per tuple (source, multicast group). Those trees are only composed of multicast capable nodes. Three steps are distinguished: the tree building, the tree maintenance, and the tree detachement. Each step is detailed in a subsection.

2.1 Tree building

Multicast routers declare themselves to the entire network by broadcasting a MC_CLAIM message (using the optimized flooding technique of OLSR). Such nodes will disseminate this information periodically.

Once a source wants to send data to a specific multicast group G, it sends a SOURCE_CLAIM message enabling nodes which are members of this group to detect its presence and to attach themselves to the associated multicast tree.

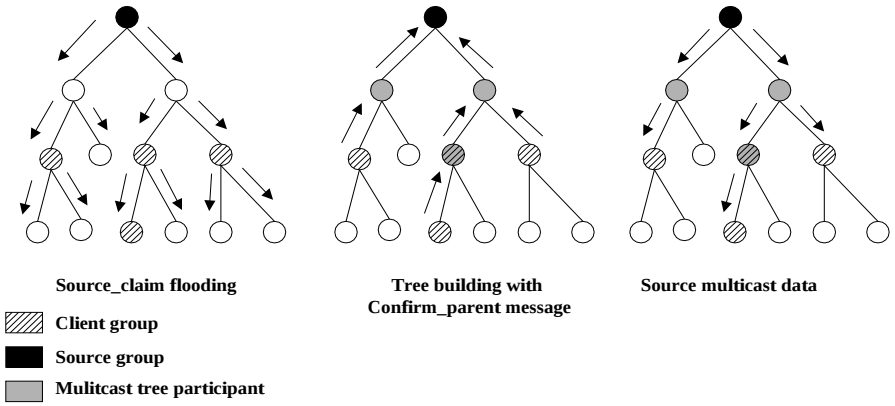


Figure 1: Tree building mechanism

This message is flooded within the ad hoc network using the optimized flooding technique of OLSR. Branches are built in a backward manner: group members which do not know yet about this source try to attach themselves to the corresponding tree.

More specifically, when a group member receives a SOURCE_CLAIM message and it is not already a participant of this (source, multicast group) tree, it attaches itself to the (source, multicast group)tree by proceeding as follows:

- it looks into the multicast routing table for the next hop to reach the source (the multicast routing table provides shortest routes to all the multicast capable nodes). This next hop becomes its parent in the multicast tree.
- Then it sends a CONFIRM_PARENT message to its parent node.
- The parent node receiving this message attaches itself to the (source, multicast group) tree, if it is not already a participant to this tree.

The CONFIRM_PARENT message is handled hop by hop, by intermediate multicast routers which build the corresponding branch.

2.2 Tree maintenance

The trees are periodically refreshed, by means of the SOURCE_CLAIM message and the CONFIRM_PARENT message. Notice that topology changes are still detected by the exchange of topology control messages which is done naturally by OLSR. Thus, trees updates are triggered by the detection of topology changes.

2.3 Tree detachement

If a node wants to leave the multicast tree and it is a leaf, it detaches itself from the tree: it just sends a LEAVE message to its parent in this multicast tree. If its parent becomes a leaf, and this parent is not a group member, it detaches itself from the tree on its turn.

The LEAVE message is processed hop by hop and unused branches are deleted automatically.

3 Tables

MOLSR uses the information stored and handled by OLSR. Additional information are kept in three different tables: MC_router_table, MC_tree_table, and Multicast_routing_table.

3.1 MC_router_table

This table contains all the nodes with multicast capabilities. Each entry contains the MM_addr and the MM_time. MM_addr stands for the node address and MM_time specifies the time at which this record expires.

3.2 MC_tree_table

This table contains the information dealing with different multicast groups that this node is participating in. The table stores one entry per tuple (source, multicast group). Each entry has the following information:

Source_tree_info
consists of the address of the source_tree (MT_source_addr) and the time to live for this entry (MT_source_tree_time).

Multicast_Group_addr
The address of the multicast goup (MT_group_addr).

Parent address
The address of the next node towards the source (MT_parent_addr).

Sons information
List of the sons.
For each son, its address (MT_son_addr) and a time to live for it (MT_son_time) are recorded.

3.3 Multicast_routing_table

The multicast_routing_table provides the shortest routes to all the multicast capable nodes in the network using only the multicast routers. The calculations are done in the same way as they are done for OLSR routing table. Each entry has the following information:

MR_dest
the destination multicast capable node.

MR_next
The next hop multicast router in the route to MC_destination.

MR_dist
The estimated number of hops to reach the MC_destination.

The multicast_routing_table is updated whenever the OLSR routing table is recalculated or when a new multicast node is discovered, or it disappears.

4 Message format

Four new messages are needed to handle the multicast trees:

MC_CLAIM message
used to declare the multicast capabilities of a node (i.e this node supports MOLSR extension).

SOURCE_CLAIM message
used by the sources to declare themselves in a multicast group.

CONFIRM_PARENT message
used to attach a node to its parent in a multicast tree.

LEAVE message
used to leave a multicast tree or to change the parent of a node.

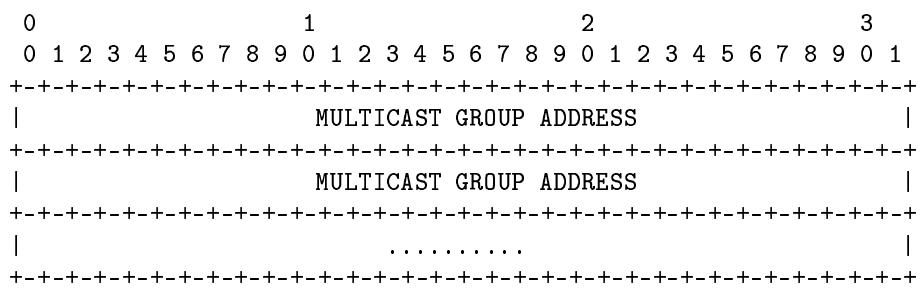
The different messages use the uniform message format for OLSR[3]. Each new message will be assigned a new type of message in the message header (see the section Proposed values for the constants).

4.1 MC_CLAIM message

The MC_CLAIM message does not carry a specific information. It is only needed to flood a message header with the corresponding type of message. It is used by multicast routers to build their MC_router_table.

4.2 SOURCE_CLAIM message

The SOURCE_CLAIM is flooded by the source node, root of the multicast trees. We recall that a multicast tree is established per tuple (source, multicast group). The address of this source is specified in the message header. The source asks for members in the specified multicast groups.



MULTICAST GROUP ADDRESS
The multicast group address to which the source will multicast data.

4.3 CONFIRM_PARENT and LEAVE messages

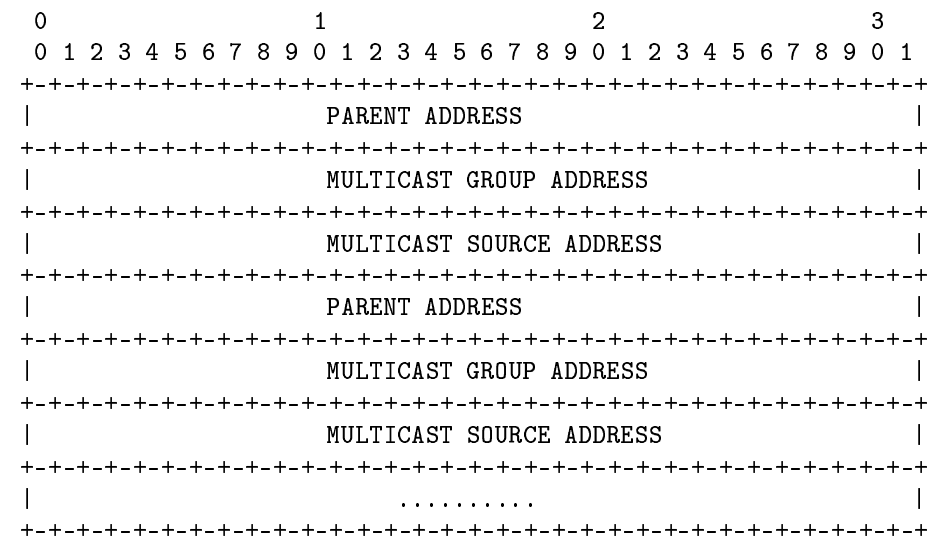
Those two messages have the same message format. To distinguish between them,the message type of the header has to be checked.

This message has several fields:

PARENT ADDRESS
The address of the next node toward the source of the multicast tree.

MULTICAST GROUP ADDRESS
The multicast group address for which the message is destinated.

MULTICAST SOURCE ADDRESS
The address of the multicast source. This address combined with the multicast group address identifies the multicast tree.



5 Detailed operation

We describe here the functioning of the multicast extension and how the different messages are processed.

5.1 Requirements

The calculation of the MPR set as done in OLSR must be modified. The MPR set must be calculated in such a way that, 2-hop multicast capable nodes are covered by 1-hop multicast capable nodes when it is possible. Nodes with multicast capabilities will be chosen preferably to the other nodes. And, of course, this set should be minimal and must cover all the 2-hop nodes.

5.2 Advertisement of multicast routers

Each node with multicast capabilities declares itself by sending a MC_CLAIM to all the other nodes in the network. With this information the other nodes can calculate routes to different sources in the same way as the regular OLSR routing calculation. The routes are composed only with multicast routers. The MC_CLAIM message is flooded each

MC_CLAIM_PERIOD seconds. Since this information does not change in time, the MC_CLAIM_PERIOD should have a value as big as possible. This information is kept in the MC_router_table for MC_HOLD_TIME seconds.

5.3 Advertisement of multicast sources

The source declares itself by sending a SOURCE_CLAIM message. This message is first generated whenever a node wants to send data to multicast group G. This SOURCE_CLAIM message will be flooded in the entire network.

5.4 Tree building

A tree is associated with any tuple (source, multicast group). This tree is built in a backward manner automatically without any central entity from the different multicast group members to the source. Branches are constructed hop by hop. The tree building is started when a multicast router receives a SOURCE_CLAIM message for a group in which it is participating. Branches are attached to the multicast tree by means of the CONFIRM_PARENT message.

5.4.1 SOURCE_CLAIM message processing

When a participant receives a SOURCE_CLAIM it does the following operations:

- **IF** there is no entry corresponding to that group and source **THEN**
  1. create a new one.
  2. set the MT_source_addr to the originator of the message.
  3. set the MT_source_tree_time to the SOURCE_HOLD_TIME.
  4. set the MT_group_addr to MULTICAST GROUP ADDRESS.
  5. set parent and sons information fields to NULL.
  6. Send a CONFIRM_PARENT message to its parent. The parent is the corresponding MR_next in the multicast_routing_table to reach the source.
- **ELSE** update the MT_source_tree_time.

- **IF** the node is a MPR of the sender, **THEN** it relays the SOURCE_CLAIM message.

5.4.2 CONFIRM_PARENT message processing

The CONFIRM_PARENT message is sent by a node to its parent. Upon receiving such a message, a node has to do the following operations:

- **IF** a corresponding entry to that group and source tree does not exist in the MC_tree_table, **THEN**
  1. create a new one.
  2. set the MT_source_addr to the originator of the message.
  3. set the MT_source_tree_time to the SOURCE_HOLD_TIME.
  4. set the MT_group_addr to MULTICAST GROUP ADDRESS.
  5. set parent and sons information fields to NULL.
- **IF** the son address does not exist in the sons list, **THEN**
  1. create a new record for this son.
  2. set MT_son_addr to the originator address of the CONFIRM_PARENT message (found in the header of the message).
  3. set MT_son_time to current time + SON_HOLD_TIME.
- **ELSE** update MT_son_time to current time + SON_HOLD_TIME.
- **IF** the parent address is equal to NULL,**THEN**
  1. set the MT_parent_addr to the next hop (MR_next) in the multicast routing table of the current node to reach the source.
  2. send a CONFIRM_PARENT message to the MR_next.

5.5 Tree maintenance

Each source periodically sends a SOURCE_CLAIM message to the whole network. In this way, new multicast group members can join the corresponding multicast tree. Upon a receipt of this message the nodes in the specific tree will update their time to live field MT_source_tree_time in the MC_tree_table.

The multicast_routing_table is recalculated whenever the neighborhood or topology changes. If the next hop node (MR_next) towards the source changes, the node (the group member or a participant node in the tree) must generate a new CONFIRM_PARENT message to inform the new parent. Then, it MAY generate a LEAVE message to disable the old route (of course if the neighbor is still reachable). No message is sent if the next hop does not change.

Each participant will send periodically a CONFIRM_PARENT message to its parents in order to refresh the corresponding timeout each CONFIRM_PERIOD seconds. Those messages may be grouped in a single one as allowed by the message format.

5.6 Detachement of a multicast tree

A LEAVE message is generated by a leaf node which wants to leave a multicast tree, or by another intermediate node which has received such a message and this participant is not a multicast group member.

A node can leave a group or a specific tree. Leaving a specific tree may occur when the topology changes and a participant finds a new route to the corresponding source. To leave a specific tree a node may send a LEAVE message to its parent. If this parent becomes a leaf, and if it is not a group member it will send on its turn a LEAVE message to its parent.

Leaving a multicast group means that a node is no more interested in that group. Consequently, the node leaves all the trees associated with this group.

A participant will also send a LEAVE message to its parents when it has no more sons (e.g timed out) and it is not itself a multicast group member. This procedure eliminates quickly the unused branches in the network.

6 Sources without multicast capabilities

In this section we deal with the sources that are not multicast routers. As usual such sources send multicast data to multicast groups. In that case, the following MOLSR procedures are modified:

- The SOURCE_CLAIM generation and processing.
- The CONFIRM_PARENT generation for the source neighbors.
- The computation of the next hop to reach the source.

6.1 SOURCE_CLAIM generation and processing

The designated multicast router (see next section) of this source is in charge of generating a SOURCE_CLAIM message on behalf of this source. In that case, the first field of the SOURCE_CLAIM message contains the individual address of the source. The following fields specify the multicast group addresses to which the source sends data. Notice that it is easy to distinguish a multicast address from the individual address of the host.

The SOURCE_CLAIM message is processed as previously described except that the MT_source_addr is set to the first field of the SOURCE_CLAIM message.

The SOURCE_CLAIM message is periodically sent by the designated router to refresh the corresponding entry in the MC_tree_table.

6.2 COMFIRM_PARENT message generation

The source neighbors do not send a CONFIRM_PARENT message, because the source is not able to process them.

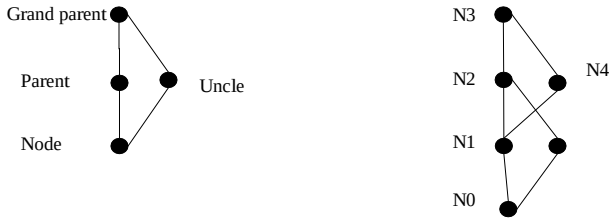


Figure 2: Reliable branches

6.3 The computation of the next hop to reach the source

In order to compute the next hop to the source, the multicast routers that are not source neighbors, select the next hop to one of the MPRs with multicast capabilities of the source giving the shortest distance.

7 Reliable multicast tree

Sometimes a link fails, and data are lost. This situation is very odd for a multicast tree, because a whole part of it, is banned from receiving the data due to a single link failure. To prevent this kind of problems the branches of the multicast tree are doubled when possible in order to have a more robust and reliable multicast tree. This mechanism offers an alternative path and overcomes link failure in the classical MOLSR branches.

Each node N doubles the number of relayers, but those relayers will repeat the message only if needed (i.e. on link failure). Each participant, chooses two nodes, a parent node and an uncle node having a link with the grand parent node (figure 2). The parent and uncle nodes should be neighbors if possible, and if there are multiple candidates for the uncle, the node N should choose the one with the lowest identifier. Thus, the grand parent node has two potentiel relayers to join the node N. The node N accepts the data relayed by its parent or by its uncle node. And it will retransmit the data only once.

In the figure 2, N1 receives a CONFIRM_PARENT message from N0. N1 knows the two hop topology, and will choose for example N2 as a parent and N4 as an uncle to reach the N3 node in order to build the branches. N1 sends CONFIRM_PARENT message to N2, and a CONFIRM_UNCLE to N4. The CONFIRM_UNCLE message has the same format as the CONFIRM_PARENT message.

Mnemonic	Value	Extension
MC_CLAIM	7	Multicast
SOURCE_CLAIM	8	Multicast
CONFIRM_PARENT	9	Multicast
LEAVE	10	Multicast
CONFIRM_UNCLE	11	Multicast

Table 1: Message types

8 Proposed values for the constants

$MC\_CLAIM\_PERIOD = 30sec$
$MC\_HOLD\_TIME = 3 \times MC\_CLAIM\_PERIOD$
$SOURCE\_CLAIM = 15sec$
$SOURCE\_HOLD\_TIME = 3 \times SOURCE\_CLAIM$
$CONFIRM\_PERIOD = 10sec$
$SON\_HOLD\_TIME = 3 \times CONFIRM\_PERIOD$
$CONFIRM\_DELAY = 1sec$

The message types of this extension are given by the table 1.

9 WIGMP

9.1 Motivation

The Internet Group Management Protocol (IGMP) is used by hosts to report their multicast group membership to neighboring multicast routers[2]. It is also used by multicast routers to discover group members. According to IGMPv2[6] and IGMPv3[7], IGMP is required to be implemented by any host wishing to receive IP multicasts. However in a wireless context like MANET, the use of IGMP may lead to some inconsistencies. More precisely, the elimination procedure of queries as described in IGMP is problematic in a wireless context:

- The elimination procedure eliminates too many queriers, as illustrated by this example.

Example :

In the figure 3 a host H has only one neighbor router R2. Router R1 and router R2 are neighbors. Initially both routers R1 and R2 are queriers. When router R2 receives the query of router R1 with a smaller address, router R2 becomes non querier. Router R1 is the only querier. As host H can only be heard by R2, router R1 does not receive the unsolicited report of H for group G. Hence no router will forward IP multicast packets for group G to H.

To cope with this problem we propose to modify the behaviour of IGMP in the routers.

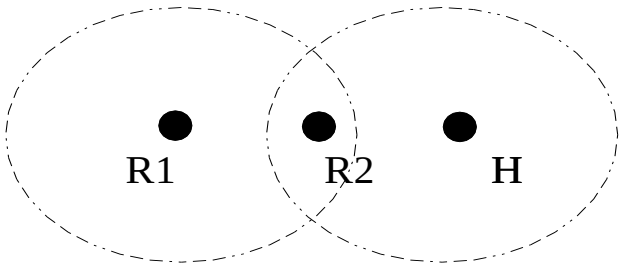


Figure 3: Elimination mechanism in IGMP

9.2 WIGMP overview

WIGMP is a Wireless IGMP. It is only concerned with the exchanges between hosts and routers to determine group membership. The goals of WIGMP are:

- to enable any multicast router to know whether at least one host in its neighborhood is member of a multicast group;
- to enable a multicast router to know whether it must forward multicast data to a host in its neighborhood.

On a multicast router, WIGMP coexists with a multicast routing protocol. The multicast routing protocol is in charge of building and maintaining a multicast structure (e.g. a tree between a source and all the multicast routers participants of a multicast group) enabling any member of a multicast group G to receive IP multicast data destinated to G.

Notice that, in WIGMP there is no new messages added to the IGMP ones, we keep the same messages specified as in IGMPv3[7]. The host IGMP is kept as it is also.

The major change is in the IGMP router. We modify the behaviour of the IGMP router when sending and receiving those messages.
Consequently:

1. The rule for multicast routers elimination is modified.
2. For each host H, a multicast router in H neighborhood is designated to forward multicast data.
3. A multicast router, member of a multicast group, plays the only role of IGMP router.

Whenever a host H wants to join a multicast group G, it sends an IGMP report which is called Unsolicited Report to reach neighbor multicast routers. MOLSR routers do not send this kind of messages. If a MOLSR router wants to join a group it builds simply and immediately a route to the sources of the requested group. A multicast router is designated to forward multicast data to H (see the designation procedure in the next section).

Multicast routers that have not been eliminated (see the elimination procedure in the next section), periodically send general queries to their neighbors.

9.3 WIGMP router

9.3.1 Elimination procedure

The elimination procedure is modified in such a way that IGMP is a particular case of WIGMP. The role of this procedure is to select a multicast router in charge of the host queries in the neighborhood. However, any host must be queried by a multicast router. In IGMPv2 and v3, if two multicast routers hear each other, only the one with the smallest address is elected as a querier. In WIGMP, the following rule is applied:

Let R_i and R_j, be any two multicast routers, and $OneHop(R_i)$ be the one hop neighborhood of R_i
IF $OneHop(R_i) \cup R_i = OneHop(R_j) \cup R_j$
THEN the router R_j with the highest address is eliminated
ELSE IF $OneHop(R_i) \cup R_i$ is included in $OneHop(R_j) \cup R_j$
THEN the router R_i is eliminated.

9.3.2 The router designation procedure

The router designation procedure enables the designation of the multicast router in charge of forwarding multicast data to a host.

When a multicast router receives an Unsolicited Report from a host H its checks its neighbor table to see whether it has the lowest IP address among the multicast routers neighbors of H. In fact, with OLSR, each node has a two hop neighbors table which provides all links in the neighborhood. In this way, the multicast router with the lowest IP address designates itself as a router to this host.

If the host H moves or a link breaks with this router, another multicast router having detected the presence of H, is designated as a multicast router for H. It builds a branch to the multicast group (if it does not exist), and forwards the corresponding multicast data.

9.3.3 Interaction between WIGMP and MOLSR

Each time there is a change of group G membership, WIGMP notifies its local multicast routing protocol when:

- either an unsolicited report has been received from a new member,
- or a previous member has left.

9.4 The host behaviour

The host that is not a multicast router behaves according to IGMP as in the wired network. When it wants to join a multicast group, it starts sending its Unsolicited Report, and, then normal reports upon receiving queries from multicast routers. In case of the presence of multiple multicast routers sending their query messages in the neighborhood, the host bundles multiples multicast group reports as in IGMPv3 and its reports are addressed to all multicast routers in its neighborhood.

References

[1] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. Request for Comments (Best Current Practice) 2119, Internet Engineering Task Force, March 1997.

[2] Fred Baker. Requirements for IP Version 4 Routers. Request for Comments (standard track), Internet Engineering Task Force,June 1995.

[3] Philippe Jacquet, Amir Qayyum, Thomas H. Clausen, Anis Laouiti, Laurent Viennot, Pascale Minet and Paul Muhlethaler. Optimized Link State Routing Protocol. Internet-Draft, draft-ietf-manet-olsr-07.txt, 10 June 2003. Work in progress.

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

[5] A. Qayyum, A. Laouiti, L. Viennot,"Multipoint relaying technique for flooding broadcast messages in mobile wireless networks", HICSS, Hawaii, Jan 2002.

[6] William C. Fenner. Internet Group Management Protocol, Version2. Request for Comments 2236, Internet Engineering Task Force, November 1997.

[7] B. Cain, S. Deering, B. Fenner, I. Kouvelas, A. Thyagarajan, "Internet Group Management Protocol, Version 3", Request for Comments 3376, Internet Engineering Task Force, October 2002.

[8] J. Moy, " Multicast Extensions to OSPF", IETF RFC 1584, March 1994.

[9] M. Liu, R. R. Talpade, A. McAuley, E. Bommaiah, "AMRoute: Adhoc Multicast Routing Protocol", UMD TechReport 99-8.

[10] J.J. Garcia-Luna-Aceves, E.L. Madruga, "The Core Assisted Mesh Protocol", IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks, Vol. 17, No. 8, pp. 1380-1394, August 1999.

[11] S.-J. Lee, M. Gerla, C.-C Chiang, "On Demand Multicast Routing Protocol", In Proceedings of the IEEE Wireless Communicationns and Networking Conference (WCNC), pages 1298-1304, New Orleans, LA, September 1999.

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

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay 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 - 38330 Montbonnot-St-Martin (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