UNIVERSITY OF OSLO
Department of Informatics

Multicast in a Heterogeneous Cellular and Ad Hoc Network

Specification of an Ad Hoc Routing Protocol with Cellular Assistance

Research report 334

Mariann Hauge

ISBN 82-7368-289-7
ISSN 0806-3036

March 2006

UNIVERSITAS OSLOENSIS • MDCCCXI •

1 Introduction

The Ad Hoc Routing Protocol with Cellular Assistance is intended to be used for high-bandwidth streaming on a heterogeneous 3G-cellular [5] and Ad Hoc IEEE 802.11 [6] network architecture. The architecture and a preliminary specification of the protocol are presented in [1]. This report describes an updated version of the protocol in details.

The multicast protocol is build from an optimized set of standard ad hoc multicast protocol mechanisms like the ones used in e.g., MAODV [2], ODMRP [3] and ADMR [4]. The ad hoc mechanisms are complemented with some routing support from the administered 3G-cellular network.

The protocol describes the IP-encapsulated signalling packets and routines needed to execute the multicast protocol in the 3G-cellular and ad hoc networks. This document **does not** describe internal signalling and routines in the 3G-cellular network needed to establish and maintain the 3G-cellular channels and to support the multicast protocol in the cellular network. The multicast protocol design is adequate for simulation purposes, however before the protocol is implemented for a real testbed, many aspects of the 3G-cellular part of the protocol should be optimized and modified to suit the relevant 3G release.

2 Abbreviations and Terminology

MGTW	Multicast gateway between the 3G-cellular multicast distribution and the ad hoc multihop multicast distribution.
MZONE	Represents the fraction of a 3G cell/sector covered by a cellular multicast/broadcast channel.
NOH	The number of hops on the multihop path between a source and a destination.
TTL	Time to live (in hops) for a flooded packet.
3G-cellular	Refers to a 3G infrastructure-based architecture (e.g., UMTS).
On-tree terminal	A mobile terminal that is part of a multicast forwarding tree.
RELAY	Represents an on-tree terminal that is not a member of the multicast group

3 Overview

3.1 The Heterogeneous Architecture

The Multicast protocol is designed for the heterogeneous 3G-cellular and ad hoc network architecture. The purpose of the architecture is to reduce the radio resources required to provide high-bandwidth group services in the 3G-cellular network by reducing the range of a 3G multicast channel, and use local ad hoc networks to forward the data onto users located outside the 3G multicast range.

In this architecture the 3G network administers Authentication, Authorization and Accounting (AAA) mechanisms, IP address assignments and multicast group management. Local ad hoc multihop networks are established on demand by the 3G-cellular network. Several ad hoc forwarding networks may coexist and cover adjacent areas.

All multicast terminals within coverage of the 3G multicast channel are potential multicast gateways between the 3G-cellular distribution (MZONE) and an ad hoc multicast forwarding network (see Fig. 1).

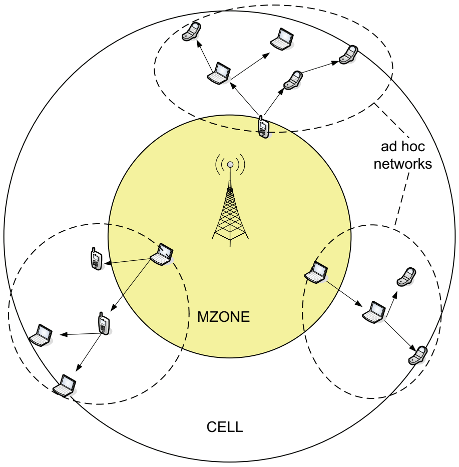


Fig. 1. The figure pictures the heterogeneous architecture. MZONE represents the range of the 3G multicast channel.

Terminals that want to join a multicast service must register with the 3G-cellular network. The cellular network executes a *Resource validation algorithm* to decide how the multicast data shall be distributed to the terminals (e.g., unicast, multicast or not available). The cellular network continuously monitors the multicast terminals and dynamically changes the data distribution method based on e.g., cellular channel quality, location of terminals, and available cellular radio resources.

3.2 The Multicast Routing Protocol

The protocol consist of two parts; one part that executes in the ad hoc network domain and one part that runs in the 3G-cellular network domain. An overview of the protocol is given as follows:

Establish a multihop network: A standard query (flooded) - response mechanism is used to establish a local ad hoc forwarding network. When the cellular network finds that an assisting ad hoc network is needed for multicast transport, potential gateways are instructed to transmit a limited query and wait for response from the multicast terminals.

The base station is informed whenever a multicast terminal is not able to connect to the multicast tree, or it experiences a very long path to the multicast gateway.

Add members to an existing network: A three-way handshake is used to connect terminals to an existing network. When the cellular network receives a join from a terminal in the vicinity of an existing ad hoc forwarding network, it instructs the joining terminal to broadcast a limited connect query. On-tree members respond to this message. The new member finally validates the best path.

Link breaks in the multihop network: A broken link is detected when a terminal has not received data for a defined period of time. The downlink terminal attempts a local repair with a three-way handshake similar to the way new members are added to a tree. In this case however, the query message holds the identification of the latest received multicast packet and the number of hops towards the gateway to avoid routing loops. This message also serves as a notification to downlink nodes that an uplink node is attempting to repair a broken link (i.e. downlink nodes should refrain from doing so for a defined time.)

Mobility in the multihop network: All terminals continuously monitor the ad hoc channel for neighbour data traffic. If a multicast terminal detects a parallel multicast flow, it will join the adjacent branch if the new path offers a better route to a multicast gateway. The transferring terminal is attached to the new branch immediately after it successfully unicast a response (validate) on the new uplink. The previous uplink node is notified with a multicast-tree-leave message. If this message for some reason can not be delivered, a softstate mechanism will eventually prune the leaving terminal from the old branch.

Refresh of the multihop multicast trees: The routing scheme does not include any periodic refresh of the multicast distribution tree. Maintenance of the routing tree depends on passive acknowledgments, detection of link breaks and local link repairs. A multicast receiver will also switch to another tree/branch connection if a better path is overheard.

After some time with many local link repairs, the distribution tree will be less optimal and it might be beneficial to refresh the tree structure. A new multicast routing tree is established when the multicast gateways initiates a new query (flooded) - response message sequence. The 3G-cellular network orders the gateways to refresh the multicast trees for the following reasons:

- The cellular network wants to add or remove a gateway.
- Some terminals have reported “no connection” or a “long path” to the multicast gateway.

4 Message Formats

Below follows a description of the normal use of most of the multicast protocol message fields. If a field is used differently for a specific message format, the use is described next to the message format.

Type: The message type.

NOH: The number of hops this message has traversed from its origin.

Seq. no: A unique identification of the packet.

Multicast Group: IP address of the multicast group.

Original Source: IP address of the originator of the message. This field is needed since the source address in the IP header for the routing messages contains the address of the previous hop router.

Multicast Gateway: IP address of the multicast gateway (MGTW) which is the root of the current multicast tree.

TTL: The maximum Time To Live to be used with the instructed message type.

4.1 Cellular Messages

4.1.1 Instr_MQuery (unicast message)

The Instr_MQuery message is used by the cellular network to instruct a set of potential MGTWs to initiate a query-response sequence to build a multicast distribution tree.

Instr_MQuery

<table>
<tr><td>Type</td><td>R</td><td>L</td><td>TTL</td><td>Seq. no.</td></tr>
<tr><td colspan="5">Multicast group IP address</td></tr>
</table>

R: Refresh flag, this flag is used to inform when the message is used for a refresh of an existing multicast tree.

L: Leave flag, this flag is used to inform an MGTW that it is no longer needed as a gateway.

4.1.2 Instr_MConn (unicast message)

The Instr_MConn message is used by the cellular network to instruct a joining multicast member to initiate a three-way-handshake to be attached to a multicast distribution tree.

Instr_MConn

Type	TTL	Seq. no.
Multicast group IP address		

4.1.3 Inf_No-Route (unicast message)

Multicast members that are not able to connect to a multicast distribution tree inform the cellular network with the Inf_No-Route message.

Inf_No-Route

Type	Reserved	Seq. no.
Multicast group IP address		

4.1.4 Inf_NOH (unicast message)

Multicast members that experience a very long path (in number of hops) towards the MGTW inform the cellular network of this problem with the Inf_NOH message. This message is repeated periodically if the situation persist

Inf_NOH

Type	NOH	Seq. no.
Multicast group IP address		

NOH: Give the current number of hops between the MGTW and the reporting multicast receiver. NOH=0 informs that a disconnected terminal has been able to reconnect.

4.2 Ad Hoc Messages

4.2.1 MQuery (flooded, broadcast message)

The MQuery message is used by MGTWs to initiate the query-response sequence to build a multicast distribution tree.

MQuery

Type	R	NOH	Seq. no.
Multicast group IP address			
Multicast Gateway (MGTW) IP address			

R: Refresh flag, this flag is used to inform when the message is used for a refresh of an existing multicast tree.

4.2.2 MConn (flooded, broadcast message)

The MConn message is used by new multicast members to find a path to an existing multicast distribution tree.

MConn

Type	NOH	Seq. no.
Multicast group IP address		
Original source IP address		

4.2.3 MReply (unicast message)

The MReply message is the response to MQuery, MConn and MRepair. MReply with NOH=0 is used to validate the chosen path, while an MReply message with a given NOH represents one of possibly several paths available as a result of an MConn or MRepair message.

MRreplay

Type	NOH	Seq. no.
Multicast group IP address		
Original source IP address		
Multicast Gateway (MGTW) IP address		

NOH: NOH=0 means that the message is an MReply validate message.

4.2.4 MRepair (flooded, broadcast message)

The MRepair message is used to initiate a repair of a broken link in the multicast tree. This message also serves as a notification to downlink nodes that an uplink node is attempting to repair a broken link (i.e. downlink nodes should refrain from doing so for a defined time).

MRrepair

<table>
<tr><td>Type</td><td>NOH</td><td colspan="4">Seq. no.</td></tr>
<tr><td colspan="6">Multicast group IP address</td></tr>
<tr><td colspan="6">Original source IP address</td></tr>
<tr><td colspan="6">Multicast Gateway (MGTW) IP address</td></tr>
<tr><td colspan="2">Identification of last received pkt</td><td></td><td></td><td></td><td>Fragment offset of last pkt</td></tr>
</table>

Identification: Is a copy of the IPv4 header Identification field of the last received multicast packet [7]. For IPv6 compatibility this field must be increased to 4 octets.

Fragment offset: Is a copy of the Fragment offset of the respective IPv4 packet Identification [7].

4.2.5 MAck (1-hop broadcast message)

The MAck message is sent by a leaf node to inform its uplink node of its continued existence.

MAck

Type	Reserved	Seq. no.
Multicast group IP address		

4.2.6 MTL (1-hop unicast message)

A multicast member joins an adjacent multicast tree branches if this branch represents a better path to the MGTW. The Multicast Tree Leave (MTL) message is used to inform the previous uplink of the branch change.

MTL

Type	Reserved	Seq. no.
Multicast group IP address		

5 Protocol Operation

This protocol requires bidirectional links. The 802.11 MAC [6] layer provides reliable unicast. Some of the unicast messages in the protocol logic depend on this link layer acknowledgment for the protocol to detect asymmetric links.

The protocol description given below is supplemented with message sequence charts (MSC) of some message sequences in Appendix A.

5.1 Establish a Multihop Multicast Tree.

5.1.1 Multicast tree establishment: 3G-cellular channel

Mobile terminals join a multicast session by sending an IGMP [8] type membership report (join) to the cellular network. The cellular network runs the *Resource validation algorithm* to decide how the multicast group should be served. The algorithm might suggest that assistance from local ad hoc networks is required.

Thus, the 3G-cellular network decides when it needs assistance from local ad hoc networks. Next the cellular network sets the size of the MZONE. The multicast terminals inside MZONE are identified as potential MGTWs between the cellular multicast distribution and the ad hoc forwarding trees and are stored in the mGTWs list for the current multicast group. Based on approximate information of the position of all multicast terminals, a set of MGTWs are chosen to establish ad hoc multicast trees in the area outside the range of MZONE. These terminals are marked as active in the mGTWs list. An Instr_MQuery message holding the query range (MQUERY_TTL) is unicast to the chosen MGTWs.

Next the chosen MGTWs initiate a query-response sequence to build the multicast trees as described in the next section. If a multicast terminal is not able become part of the multicast tree, a Inf_No-Route message is returned to the cellular network, and the member is marked as not connected.

5.1.2 Multicast tree establishment: Ad hoc channel

Receiving Instr_MQuery and sending MQuery

A set of multicast members receive the Instr_MQuery message on the cellular channel. These terminals set their status to MGTW and initiate a limited (MQUERY_TTL) flooding of the MQuery message and wait to receive MReplys. If no MReplys are received after MQUERY_TIMEOUT, the MQuery message is repeated MQUERY_RETRANSMIT times. The MGTWs silently discards MQuery messages received from other MGTWs.

Receiving MQuery and sending MReply validate

When a multicast member receive a new MQuery it ads an entry in its routing table to the originating MGTW, and increments the NOH element in the message. As long as the IP-TTL element is not 0, the message is reforwarded. The multicast member then waits BEST_ROUTE_DELAY for possible MQuery messages that advertise a better path to an MGTW. If a better MQuery is received, the routing entry is updated, and the message is reforwarded as above. When the BEST_ROUTE_DELAY timer times out, an MReply validate message is created and unicast to the originating MGTW.

When a terminal that is not a multicast member receive an MQuery message, the terminal ads a temporary entry to the routing table that is valid REV_ROUTE_LIFE time. The terminal wait NO_MEMBER_DELAY before the MQuery is reforwarded. (This timer is used to prioritize multicast members in the multicast distribution tree.) If the terminal receive subsequent MQuery message that advertise a better path to the MGTW, the routing entry is updated and the message is forwarded after a NO_MEMBER_DELAY.

If any terminal receives an MQuery message it has already seen, or a message with a longer path to the MGTW, the message is silently discarded.

If a multicast member does not receive an MQuery message within NO_DATA_TIMEOUT time it attempts a local repair (ref. Section 5.3).

Receiving MReply validate

When a non member that is not already part of the multicast tree receives an MReply validate it sets its status to RELAY. The temporary entry in the routing table is made permanent. The terminal creates a new downlink interface in its downlink interface list. The associated interface timer is set to PASSIVE_ACK_EXPIRE. The MReply is reforwarded towards the MGTW.

When a multicast member or a terminal that is already a RELAY, receives an MReply validate, it adds a new downlink interface in its downlink interface list. The associated interface timer is set. The MReply message is not forwarded any further.

Maintaining the downlink interface list

The PASSIVE_ACK_EXPIRE timer associated with each downlink interface is reset whenever the terminal hears that the downlink node forwards a known multicast data packet, thus the terminal concludes that the downlink node is alive and the branch should be kept. Leaf nodes on the multicast tree does not forward any multicast traffic thus these terminals must send periodic 1-hop MACK messages to inform its uplink that it is present and alive. The MACK message is sent every MACK_MAX_DELAY.

If the PASSIVE_ACK_EXPIRE times out, the downlink interface is removed from the list. If no more downlink interfaces exits, then the associated multicast entry in the routing table is deleted. If the terminal is RELAY, then this terminal can be pruned from the network in the same way as RELAYs with no downlink interface is removed when a terminal leaves the multicast session (ref. Section 5.4.2 -Remove Member).

5.2 Add New Members to a Multicast Tree

5.2.1 New multicast tree members: 3G-cellular channel

When the cellular network receives a join from a terminal in the vicinity of an existing ad hoc multicast tree, it sends an Inf_MConn message holding the query range (MCONN_TTL) to the new member.

If it turns out the multicast terminal is not able to connect to the multicast tree, or it experience a very long path to the MGTW, the corresponding Inf_No-Route or Inf_NOH message is returned. The cellular network marks the member status accordingly. If the long path to the MGTW persist, the Inf_NOH message is repeated every NOH_REPORT_FREQ.

5.2.2 New multicast tree members: Ad hoc channel

Receiving Instr_MConn and sending MConn

When the new multicast member receive the Instr_MConn message on the cellular channel it initiate a limited (MCONN_TTL) flooding of the MConn message and wait to receive MReplys. (The IP address of the new member is put in the original source field of the MConn message.) If no MReplys are received after MCONN_TIMEOUT, the MConn message is repeated MCONN_RETRANSMIT times. If there is still no MReply response, an Inf_No-Route message is sent to the cellular network.

Receiving MConn and sending MReply

When an on-tree terminal receive a new MConn it ads a temporary entry (valid for REV_ROUTE_LIFE time) in its routing table. The MConn message is not forwarded any further. The terminal waits BEST_ROUTE_DELAY for possible MConn message that has traversed a better path. An on-tree RELAY waits an additional NO_MEMBER_DELAY to prioritize connections to multicast members. If a better message arrives, the routing entry is updated. After timeout, an MReply message is created. The number of hops from the current terminal to the MGTW is placed in the NOH field, and the IP address of the current node is placed in the original source field. The message is then unicast to the terminal that initiated the MConn message.

When a terminal that is not on-tree receives a new MConn it adds the same temporary routing entry as above, increments the NOH element and reforwards the message immediately assuming TTL is not 0. If a later MConn with a better route arrives, the routing table is updated, and this message is also reforwarded.

If any terminal receives an MConn message it has already seen, or a message with a longer route than the ones already seen, the message is silently discarded.

Receiving MReply and sending MReply validate

When a terminal receives the MReply message, it adds a temporary routing entry (valid for REV_ROUTE_LIFE time), increments the NOH field and reforwards the message towards its destination.

When the new multicast member receive the MReply message it adds the corresponding routing entry and wait BEST_ROUTE_DELAY time to see if an MReply with a better route arrives. After timeout, a new MReply message with NOH=0 to specify that this is a validate message is created. This message is unicast to the original source of the best MReply message. If the path from the new member to the MGTW is longer than MAX_NOH, an Inf_NOH message is sent to the cellular network. If the long path to the MGTW persist, the Inf_NOH message is repeated every NOH_REPORT_FREQ.

Receiving MReply validate

When a non member that is not already part of the multicast tree receives the MReply validate, it sets its status to RELAY. The temporary entry in the routing table is made permanent. The terminal creates a new downlink interface in its downlink interface list. The associated interface timer is set to PASSIVE_ACK_EXPIRE. The MReply is forwarded towards the on-tree destination.

When the destination (on-tree member or RELAY) receives the MReply validate, it adds a new downlink interface in its downlink interface list. The associated interface timer is set. The temporary entry in the routing table is made permanent.

5.3 Local Link Repair

5.3.1 Local link repair: 3G-cellular channel

The cellular network does not participate in the local link repair except for the situation when the link can not be repaired. In this case the Inf_No-Route message is transmitted. Additionally, the Inf_NOH message is returned if the repair results in a very long path to the MGTW. The cellular network marks the member status accordingly. If the long path to the MGTW persist, the Inf_NOH message is repeated every NOH_REPORT_FREQ.

5.3.2 Local link repair: Ad hoc channel

A broken link is detected when a terminal has not received data for a NO_DATA_TIMEOUT period of time. The terminal downlink of the break, initiates a local repair.

Sending MRepair

When an on-tree terminal detects a link break it creates an MRepair message. The unique identification (the Identification and Fragment offset fields of the IPv4 header [7]) of the last received multicast packet is included, as is the current MGTW IP address and the NOH to the MGTW. This information is used to avoid routing loops. The repairing node's IP address is put in the original source field of the packet. The Mrepair packet is flooded maximum MREPAIR_TTL hops.

If no MReplys are received after MREPAIR_TIMEOUT, the MRepair message is repeated MREPAIR_RETRANSMIT times. If there is still no MReply response, an Inf_No-Route message is sent to the cellular network.

Receiving MRepair and sending MReply

When an on-tree terminal receive a new MRepair it ads a temporary entry (valid for REV_ROUTE_LIFE time) in its routing table. If the terminal has received newer multicast packets than the packet identified in the MRepair message then the terminal is allowed to respond to the MRepair and the MRepair message is not forwarded any further. An on-tree RELAY waits NO_MEMBER_DELAY to prioritize connections to multicast members. An MReply message is created. The number of hops from the current terminal to the MGTW is placed in the NOH field, and the IP address of the current node is placed in the original source field. The message is then unicast to the terminal that initiated the MConn message.

If the on-tree terminal has not received a newer packet, it reforwards the message immediately assuming TTL is not 0. If the node's NOH to the MGTW is larger than the NOH given in the MRepair message, the terminal is downlink of the path break. The state is set to REPAIR, and the node wait UPLINK_REPAIR_TIMEOUT for the uplink to fix the break. If no data is received in this period, the node starts its own link repair.

When a terminal that is not on-tree receives a new MRepair it adds the same temporary routing entry as above, and reforwards the message immediately assuming TTL is not 0.

If any terminal receives an MRepair message it has already seen, the message is silently discarded.

Receiving MReply and sending MReply validate

When a terminal receives the MReply message, it adds a temporary routing entry (valid for REV_ROUTE_LIFE time), increments the NOH field and reforwards the message towards its destination.

When the repairing terminal receive the MReply message it adds the corresponding routing entry and wait BEST_ROUTE_DELAY time to see if an MReply with a better route arrives. After timeout, a new MReply message with NOH=0 to specify that this is a validate message is created. This message is unicast to the original source of the best MReply message. If the path from the new member to the MGTW is longer than MAX_NOH, an Inf_NOH message is sent to the cellular network. If the long path to the MGTW persist, the Inf_NOH message is repeated every NOH_REPORT_FREQ.

Receiving MReply validate

When a non member that is not already part of the multicast tree receives the MReply validate, it sets its status to RELAY. The temporary entry in the routing table is made permanent. The terminal creates a new downlink interface in its downlink interface list. The associated interface timer is set to PASSIVE_ACK_EXPIRE. The MReply is forwarded towards the on-tree destination.

When the destination (on-tree member or RELAY) receives the MReply validate, it adds a new downlink interface in its downlink interface list. The associated interface timer is set. The temporary entry in the routing table is made permanent.

5.4 Remove a Member from the Routing Tree

5.4.1 Remove multicast tree members: 3G-cellular channel

A multicast terminal sends an IGMP type leave [8] to the cellular network to inform that it wants to leave the multicast session. If the terminal was a normal member, the only action taken by the cellular network (assuming there are more members left in the multicast session) is to remove the multicast terminal from its member list and from the potential mGTWs list if it was present there.

If on the other hand, the multicast member was an active MGTW, then the cellular network removes the terminal from the member list and mGTWs list, and sends an Instr_MQuery message with the leave (L) flag set to the leaving member. Finally it initiates a refresh of the multicast trees (ref. Section 5.6 for the refresh routine).

5.4.2 Remove multicast tree members: Ad hoc channel

Remove MGTW

When a multicast member that act as MGTW want to leave the multicast session it must wait for the Instr_MQuery message with the leave flag set, from the cellular network. It changes its status to RELAY. After REFRESH_GTW_DELAY it removes all state information associated with the multicast group if it has not become RELAY for the new refreshed multicast trees.

Remove Member

When a normal multicast member wants to leave the multicast session, it changes its status to RELAY. If downlinks exist in the downlink interface list, no further action is taken. If no downlinks exist, the link to the terminal is pruned from the multicast tree with an MTL message to the uplink. Finally the terminal deletes all state information, and the routing table entry associated with the multicast group.

Receive MTL

When a terminal receives an MTL message it removes the downlink interface corresponding to the source of the MTL from its downlink interface list. The terminal checks to see if it has more interfaces left in its interface list. If interfaces exist, no further action is taken. If no more interfaces exist then a multicast member becomes a leaf node and removes the multicast entry in the routing table. If the terminal is RELAY, an MTL message is sent to the new uplink to prune the current branch from the tree, and so on.

5.5 Handle Mobility in the Network

5.5.1 Mobility mechanism: 3G-cellular channel

The cellular network periodically checks the position of the multicast terminals and identifies the multicast members that are positioned inside MZONE. If an active MGTW has been outside MZONE more than MGTW_HYSTERESIS time, this terminal must be removed from the MGTW list, and the multicast trees must be refreshed to incorporate the members of the multicast tree rooted by the parted MGTW in other adjacent trees. The cellular network unicast a Instr_MQuery message with the leave (L) flag set to the MGTW and initiates a refresh of the multicast trees (ref. Section 5.6 for the refresh routine).

On the other hand, if no MGTWs has left MZONE, but new terminals has arrived and been inside MZONE for MGTW_HYSTERESIS time, the cellular network check if any multicast members have status as not connected or if several members have reported a long path to the MGTW. In these cases a refresh of the multicast trees are also initiated. Otherwise the available mGTWs list is updated with the new passive mGTWs, and no other action taken.

Whenever the cellular network wants to increase or reduce the MZONE size, the mGTWs list will be updated accordingly, and the multicast trees refreshed.

In the cases when no refresh is performed, the cellular network will send an Instr_MConn message to terminals that has status as not connected when the current terminal position indicates that a new connection might be successful.

Whenever a not connected terminal is able to connect to the network, an Inf_NOH message with NOH=0 is sent to the cellular network to announce a successful connection.

5.5.2 Mobility mechanism: Ad hoc channel

Connect to another branch or tree

All terminals continuously monitor the ad hoc channel for neighbour data traffic. If a multicast terminal detects a parallel multicast flow, it will join the adjacent tree or branch if the new path offers a better route (in NOH) to an MGTW and has been available for more than HYSTERESIS_COUNT time. The terminal identifies the NOH by subtracting the multicast packet TTL from the preconfigured NETWORK_DIAMETER variable. The new connection is established by sending a 1-hop MReply validate message to the new uplink. The uplink subsequently ads the new downlink to its downlink interface list. The old uplink is notified of this change with a 1-hop MTL message (see Fig. 2). The MTL message is treated in the same manner as describe in Section 5.4.2-Receive MTL.

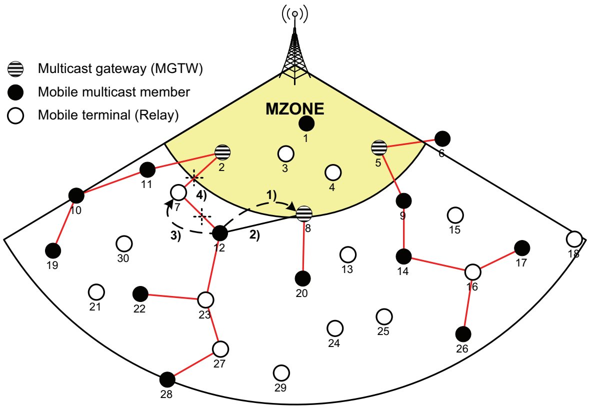

▤ Multicast gateway (MGTW)
● Mobile multicast member
○ Mobile terminal (Relay)
MZONE
1
2
3
4
5
6
11
10
7
4)
1)
8
9
15
3)
12
2)
19
30
13
14
17
18
16
21
22
23
20
24
25
26
27
28
29

Fig. 2. The figure depicts a handover to an adjacent tree. Terminal no. 12 detects a better rout to an MGTW. 1) 12 sends an MReply validate to 8. 2) 8 includes 12 as downlink interface. 3) 12 sends MTL to former up-link (Terminal no. 7) to inform that he has left the tree. 4) The links between terminals 7 and 12, and 7 and 2 are pruned from the tree.

Terminals with no connection

A multicast member that has not been able to connect to the multicast tree monitor the ad hoc channel for any data packets or signalling messages associated with the multicast group. If any message is overheard, the terminal initiates an MConn sequence to attempt a new connection to the tree (ref. Section 5.2 for the MConn sequence).

If the connection was successful and Inf_NOH message with NOH=0 is sent to the cellular network to announce a successful reconnection.

5.6 Refresh of the Multicast Trees

5.6.1 Multicast tree refresh: 3G-cellular channel

Whenever the cellular network sees the need for a refresh of the ad hoc multicast trees, it identifies a new set of active MGTWs from the terminals in the mGTWs list. The MGTWs are chosen based on their location and marked as active in the table. The previously active MGTWs that are not part of the new active MGTW set is informed with the Instr_MQuery with the leave (L) flag set. An Instr_MQuery message with the refresh (R) flag set, holding the query range (MQUERY_TTL) is unicast to the new active MGTWs.

Next the chosen MGTWs initiate query-response sequence to build the multicast trees as the described in the next section. If a multicast terminal is not able become part of the multicast tree, a Inf_No-Route message is returned to the cellular network, and the member is marked as not connected.

5.6.2 Multicast tree refresh: Ad hoc channel

Receive Instr_MQuery w/L

When a multicast member that act as MGTW receive the Instr_MQuery with the leave flag set, it changes its status from MGTW to MEMBER and stops its gateway functionality after REFRESH_GTW_DELAY time. The multicast member will be connected to a multicast tree in the subsequent multicast-tree-refresh instructed by the cellular network. If it is not able to connect to the refreshed tree, the cellular network is notified with an Inf_No-Route message

Receiving Instr_MQuery w/R and sending MQuery w/R

A set of multicast members receive the Instr_MQuery message with the refresh flag on the cellular channel. These terminals set their status to MGTW and initiate a limited (MQUERY_TTL) flooding of the MQuery message with the refresh flag set, and wait to receive MReplys. A refresh flag is set on each of the existing downlink interfaces in order to identify the interfaces that will not be used in the refreshed multicast tree. The status of the interfaces are checked after MQUERY_REFRESH time. The PASSIVE_ACK_EXPIRE timer associated with each interface is reset.

If no MReplys are received after MQUERY_TIMEOUT, the MQuery message is repeated MQUERY_RETRANSMIT times. The MGTWs silently discards MQuery messages received from other MGTWs.

Receiving MQuery w/R and sending MReply validate

When a multicast member receive a new MQuery w/R it ads an entry in its routing table to the originating MGTW, and increments the NOH element in the message. As long as the IP-TTL element is not 0, the message is reforwarded. A refresh flag is set on each of the existing downlink interfaces in order to identify the interfaces that will not be needed in the refreshed multicast tree. The status of the interfaces are checked after MQUERY_REFRESH time. The PASSIVE_ACK_EXPIRE timer associated with each interface is reset. The multicast member then waits BEST_ROUTE_DELAY for possible MQuery messages that advertise a better path to the MGTW. If a better MQuery is received, the routing entry is updated, and the message is reforwarded as above. When the BEST_ROUTE_DELAY timer times out, an MReply validate message is created and unicast to the originating MGTW.

When an non member receive an MQuery w/R message, the terminal ads a temporary entry to the routing table that is valid REV_ROUTE_LIFE time. The terminal wait NO_MEMBER_DELAY before the MQuery is reforwarded. If the terminal receive subsequent MQuery message that advertise a better path to the MGTW, the routing entry is updated and the message is forwarded after a NO_MEMBER_DELAY. If the non member was an on-tree relay, a refresh flag is set on each of the existing downlink interfaces in order to identify the interfaces that will not be needed in the refreshed multicast tree. The status of the interfaces are checked after MQUERY_REFRESH time. The PASSIVE_ACK_EXPIRE timer associated with each interface is reset. An mReply flag is also set to indicate that the RELAY must forward the first MReply it receives to refresh the connection to uplink.

If any terminal receives an MQuery message it has already seen, or a message with a longer path to the MGTW, the message is silently discarded.

Receiving MReply validate

When a non member receives an MReply validate it sets its status to RELAY. The temporary entry in the routing table is made permanent. The terminal creates a new downlink interface in its downlink interface list, or resets the refresh flag of the associated interface if the terminal was RELAY previous of the refresh. The associated interface timer is set to PASSIVE_ACK_EXPIRE. If the terminal was not RELAY or the mReply flag of an RELAY is set, the MReply is reforwarded towards the MGTW. The mReply flag is reset.

When a multicast member or a terminal that is already a RELAY, receives an MReply validate, it resets the refresh flag of the associated downlink interface or adds a new downlink interface in its downlink interface list. The associated interface timer is set. The MReply message is not forwarded any further.

When the MQUERY_REFRESH timer expires, the terminals inspects the downlink interface list. Downlinks where the refresh flag is till set, are deleted. If the downlink list becomes empty the terminal proceeds as described for reception of an MTL message in Section 5.4.2-Receive MTL.

5.7 Timers and Constants

Many of the timers and constants below should be tuned for different environments. Some of the values (e.g., MQuery_TTL, MConn_TTL and MRepair_TTL) might also be changed dynamically as supporting routines for the multicast operation sees fit.

5.7.1 Timers

MQUERY_TIMEOUT Time to wait for a response to an MQuery (set to 2s).

MCONN_TIMEOUT Time to wait for a response to an MConn (set to 2s).

MREPAIR_TIMEOUT Time to wait for a response to MRepair (set to 1s).

BEST_ROUTE_DELAY Time to wait for possible messages that have traversed a better path before an MReply is returned (set to 20ms).

NO_MEMBER_DELAY An additional time to wait for a terminal that is not member of the multicast group before the signalling message is reforwarded, or a RELAY creates an MReply (set to 10ms).

REV_ROUTE_LFE The time a temporary routing entry is valid (set to 5s).

MACK_MAX_DELAY This timers sets the appropriate times for a leaf node to send MACK. The timer depends on the constant HEART_BEAT_FREQ and is set to the time of 30*HEART_BEAT_FREQ.

PASSIVE_ACK_EXPIRE Timer that maintains the softstate of the downlink interfaces. When this timer expires, the associated downlink interface is removed. The timer is set to 3*MACK_MAX_DELAY.

NO_DATA_TIMEOUT When this timer expires, the terminal defines the uplink as broken. This timer depends on the constants and variable HEART_BEAT_FREQ, LINK_TRAVERSAL_TIME and NOH to the MGTW and is set to 6*HEART_BEAT_FREQ + NOH*LINK_TRAVERSAL_TIME.

UPLINK_REPAIR This is the time a node wait for an uplink to attempt a link repair before it try an own repair. The timer is set to

MREPAIR_TIMEOUT+1+MRepair-NOH*2*LINK_TRAVERSAL_TIME.

MQUERY_REFRESH
This timer defines how long a node should wait after a refresh MQuery for a refresh of its downlink interfaces. When the timer expires, the downlink interfaces that have not been refreshed, are deleted. The timer is inverse proportional to its distance to the MGTW and is set to REFRESH_DELAY * (the MQuery message's TTL).

REFRESH_GTW_DELAY
When this timer expires, a leaving MGTW can stop its gateway functionality (set to 0.5s).

NOH_REPORT_FREQ
A terminal reports a Inf_NOH message to the cellular network when the path to the MGTW is longer than MAX_NOH. If the long path persist, the message is repeated every NOH_REPORT_FREQ (set to 5s).

5.7.2 Constants

HEART_BEAT_FREQ
This constant reflects the packet intensity for the multicast flow and is set to the time between packets for a constant bit rate (CBR) flow that represents an approximately equivalent packet flow.

LINK_TRAVERSAL_TIME
This constant represents the time needed to process a packet in one node + the transmission time for one link. The constant is set to 10ms

REFRESH_DELAY
This constant is set to 0.5s.

MQUERY_TTL
This constant represents the range in NOH for the flooding of an MQuery packet (set to 8).

MREPAIR_TTL
This constant represents the range in NOH for the flooding of an MRepair packet (set to 3).

MCONN_TTL
This constant represents the range in NOH for the flooding of an MConn packet (set to 3).

MQUERY_RETRANSMIT
This constant sets the upper limit for retransmit of the MQuery message (set to 1).

MREPAIR_RETRANSMIT
This constant sets the upper limit for retransmit of the MRepair message (set to 2).

MCONN_RETRANSMIT
This constant sets the upper limit for retransmit of the MConn message (set to 2).

MAX_NOH
When the path length to the MGTW has exceeded MAX_NOH, an Inf_NOH message is sent to the cellular network (set to 7).

HYSTERESIS_COUNT
When a terminal has received a sequence of HYSTERESIS_COUNT packets from a new uplink with a better path to the MGTW than the current path, the terminal make an attachment to the new uplink and prunes itself from the old uplink. The constant is set to 6.

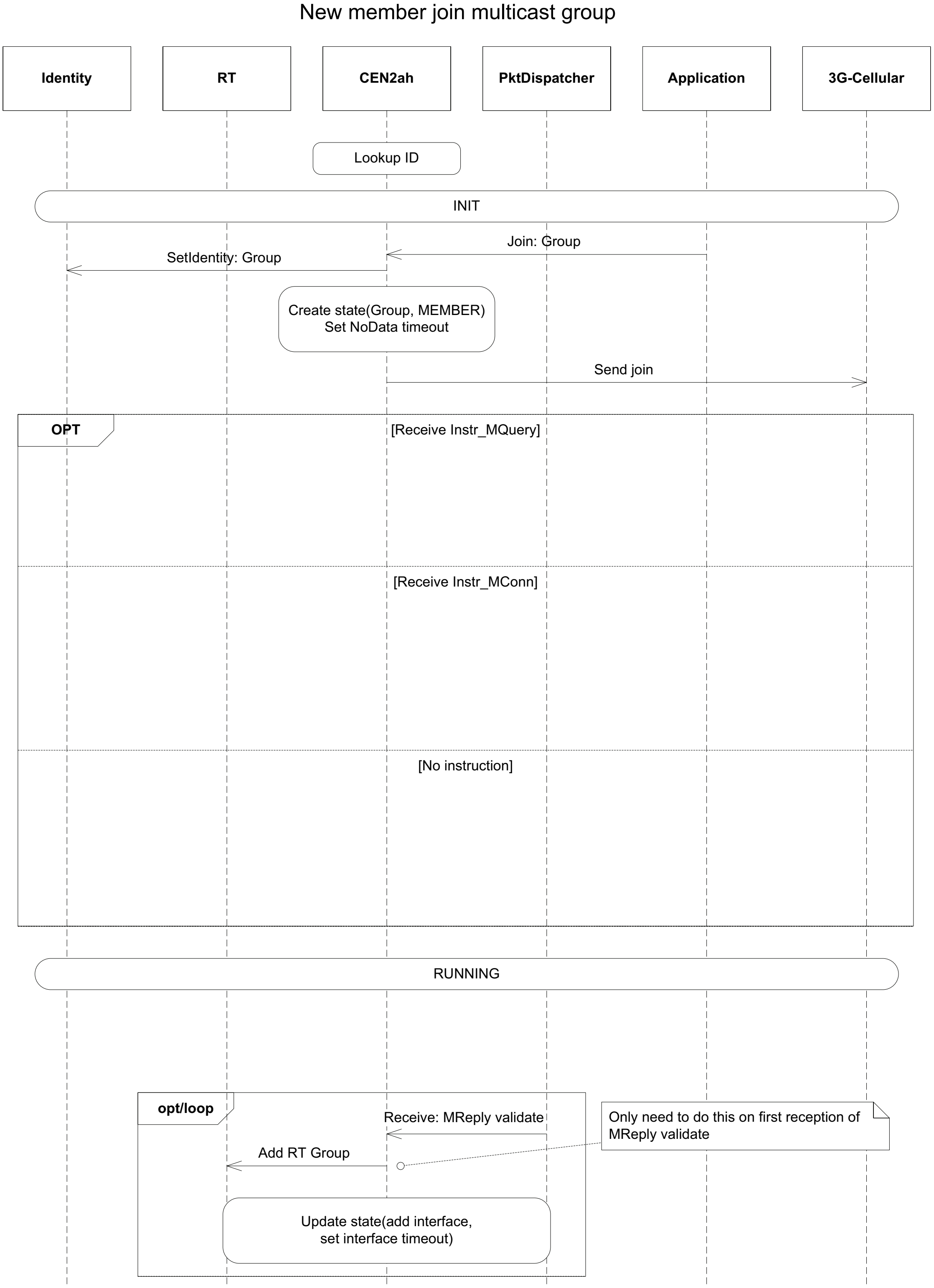
MGTW_HYSTERESIS This is the time to wait before an MGTW that has left the MZONE is defined as outside MZONE, and a multicast terminal that is entering the MZONE is put in the potential mGTWs list. The time is set to 15s.

NETWORK_DIAMETER This is the TTL used by the multicast packets. The constant is set to 35.

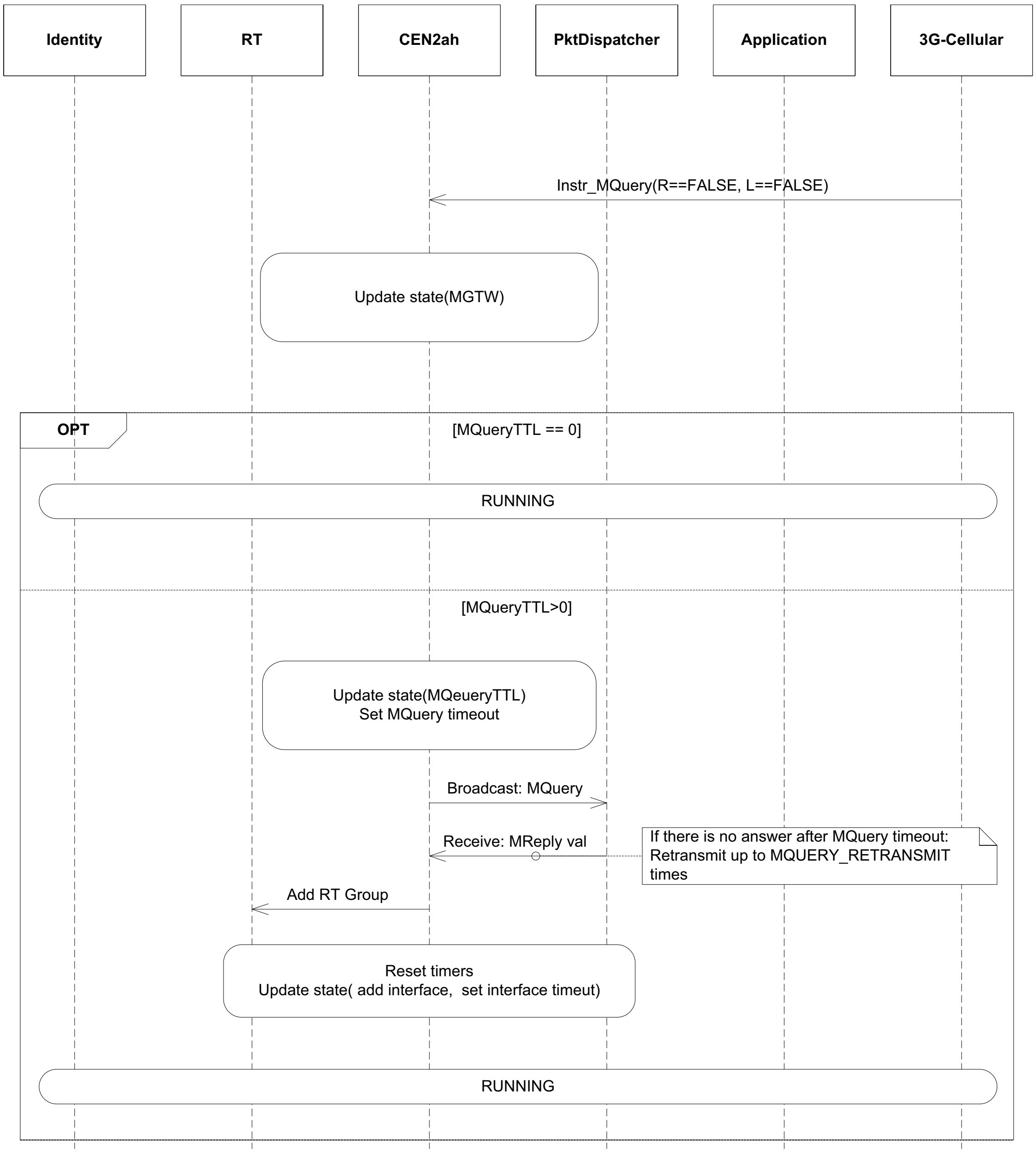
References

1. M. Hauge, et al., "Multicast-service Distribution on a Cellular Network Assisted by Local Ad Hoc Networks," *in Proc. Med-Hoc-Net*, Bodrum, Turkey, 2004, pp. 68-80
2. E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," *in Proc. MobiCom*, Seattle, WA, USA, 1999, pp. 207-218.
3. Y. Yi, et al., "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," draft-yi-manet-odmrp-00.txt, March 2003, http://www.ietf.org
4. J.G. Jetcheva and D.B. Johnson, "Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks," *in Proc. MobiHoc*, Long Beach, CA, USA, 2001, pp. 33-44.
5. 3GPP, UMTS Specifications at http://www.3gpp.org.
6. IEEE, "802 part 11 standard amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band," IEEE Std 802.11g™-2003.
7. IETF, "INTERNET PROTOCOL (IPv4)," *RFC 791*, http://www.ietf.org
8. IETF, "Internet Group Management Protocol, Version 3," *RFC 3376*, October 2002, http://www.ietf.org

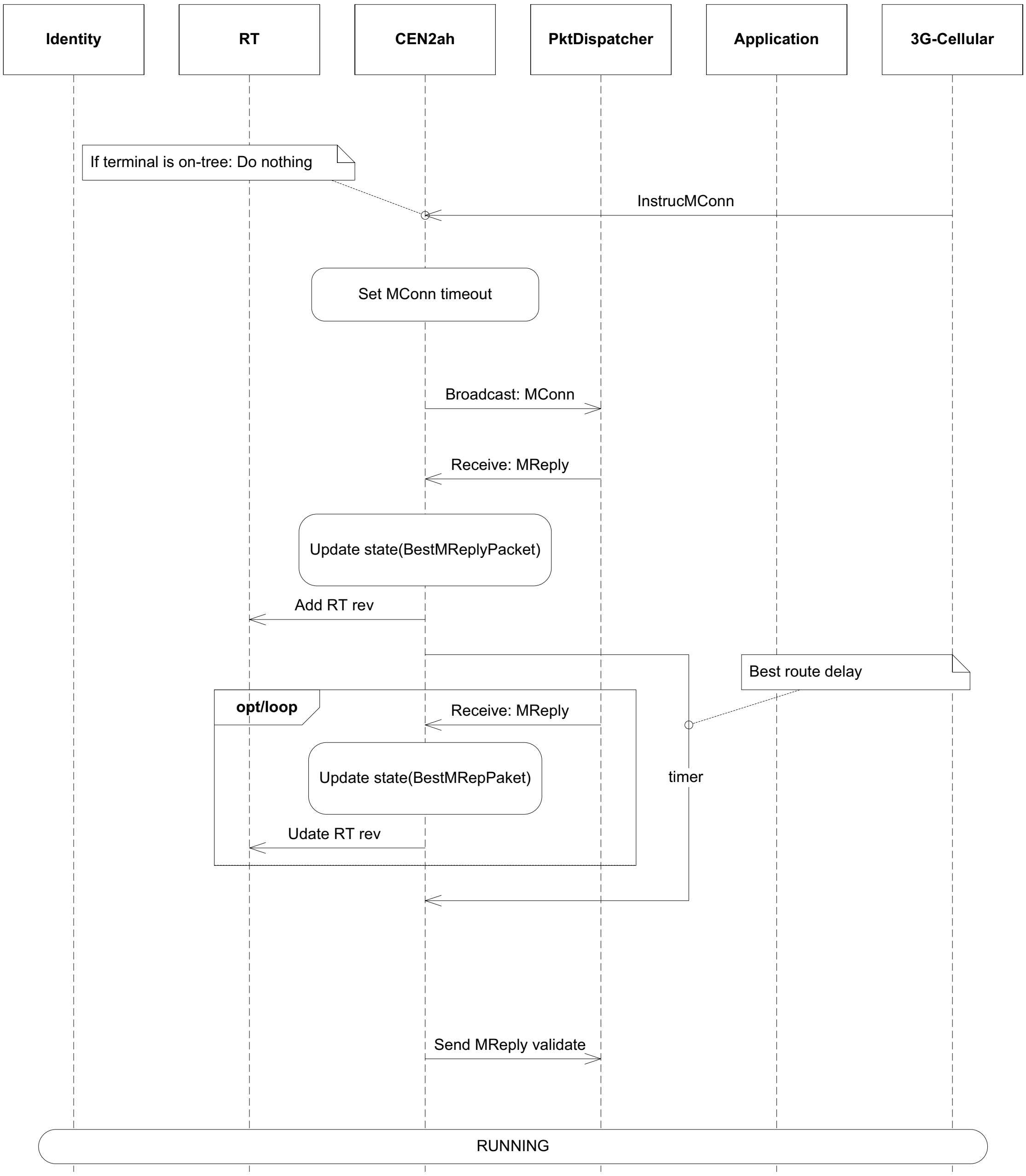
Appendix A: MSC of selected message sequences



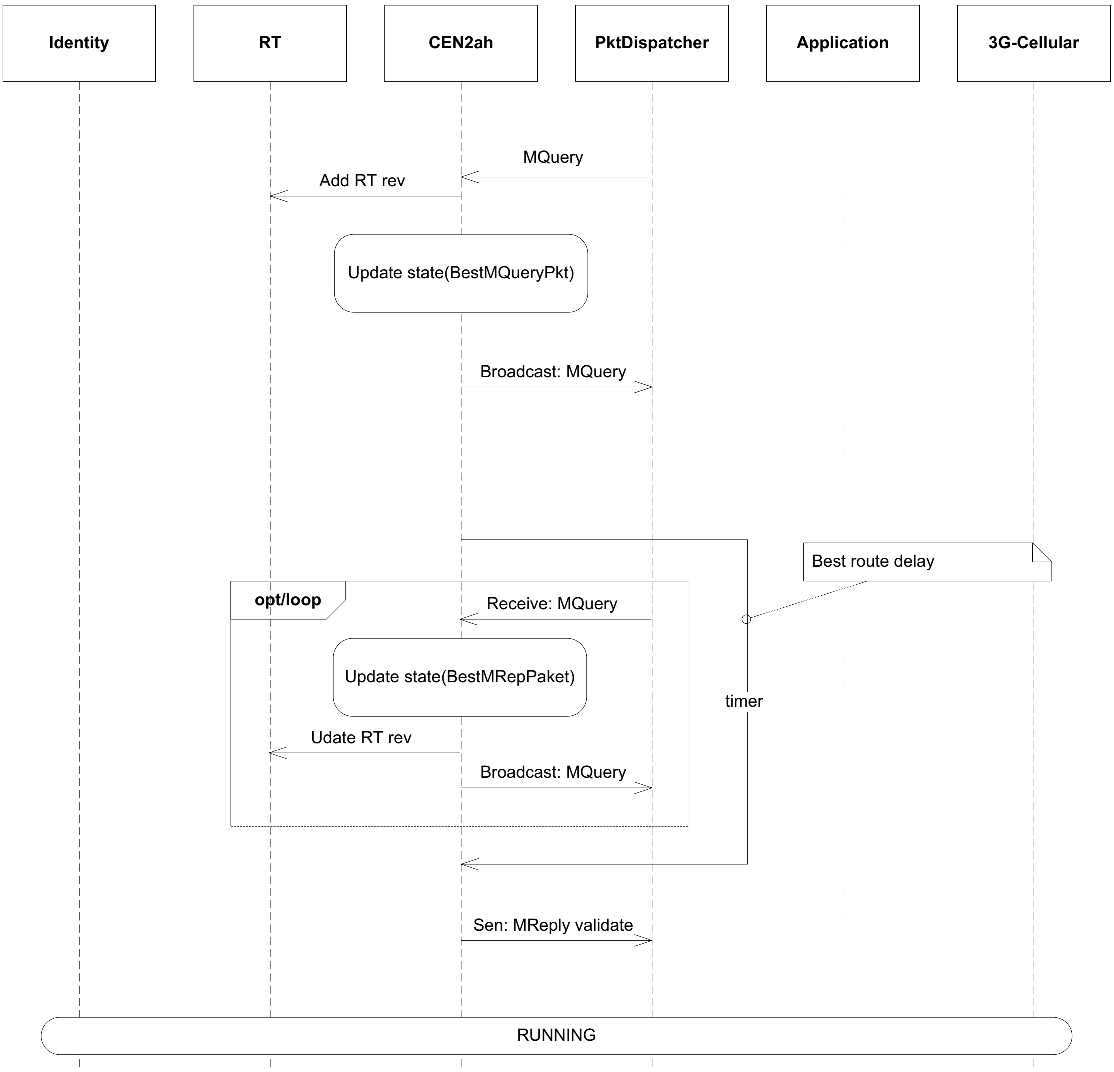
Mew member receive Instr_MQuery

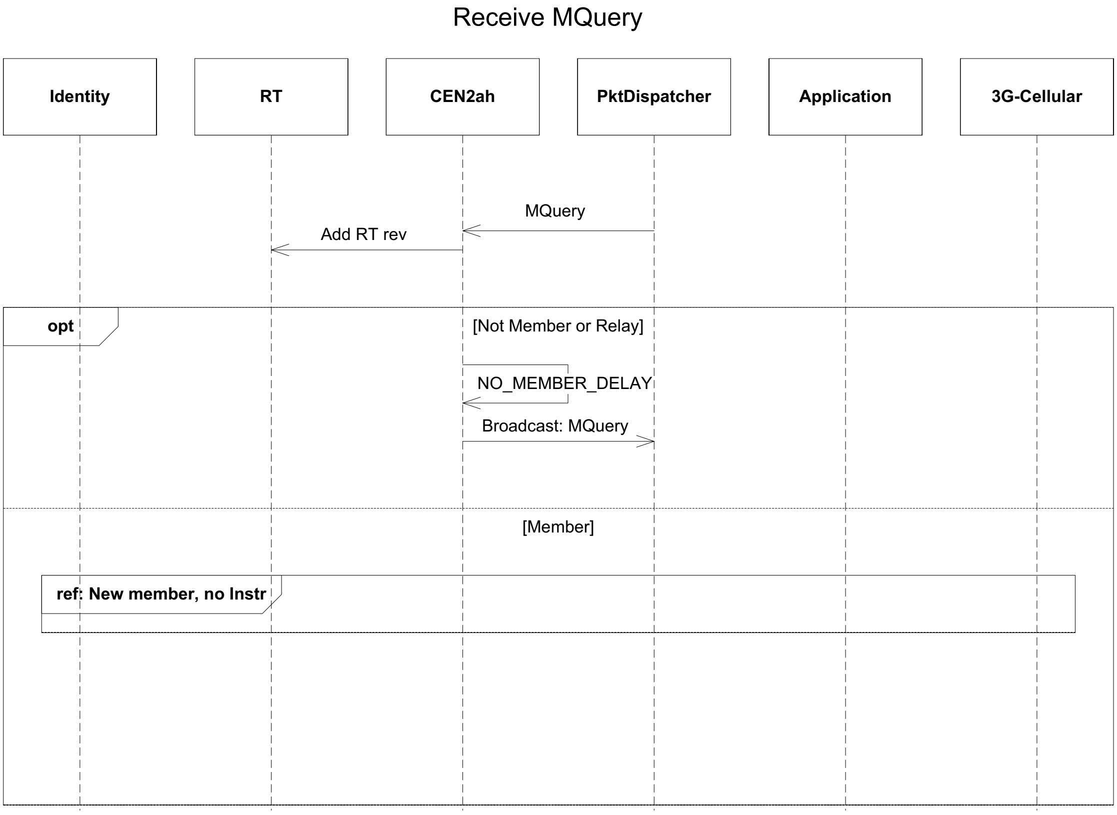


New member receive Instr_MConn

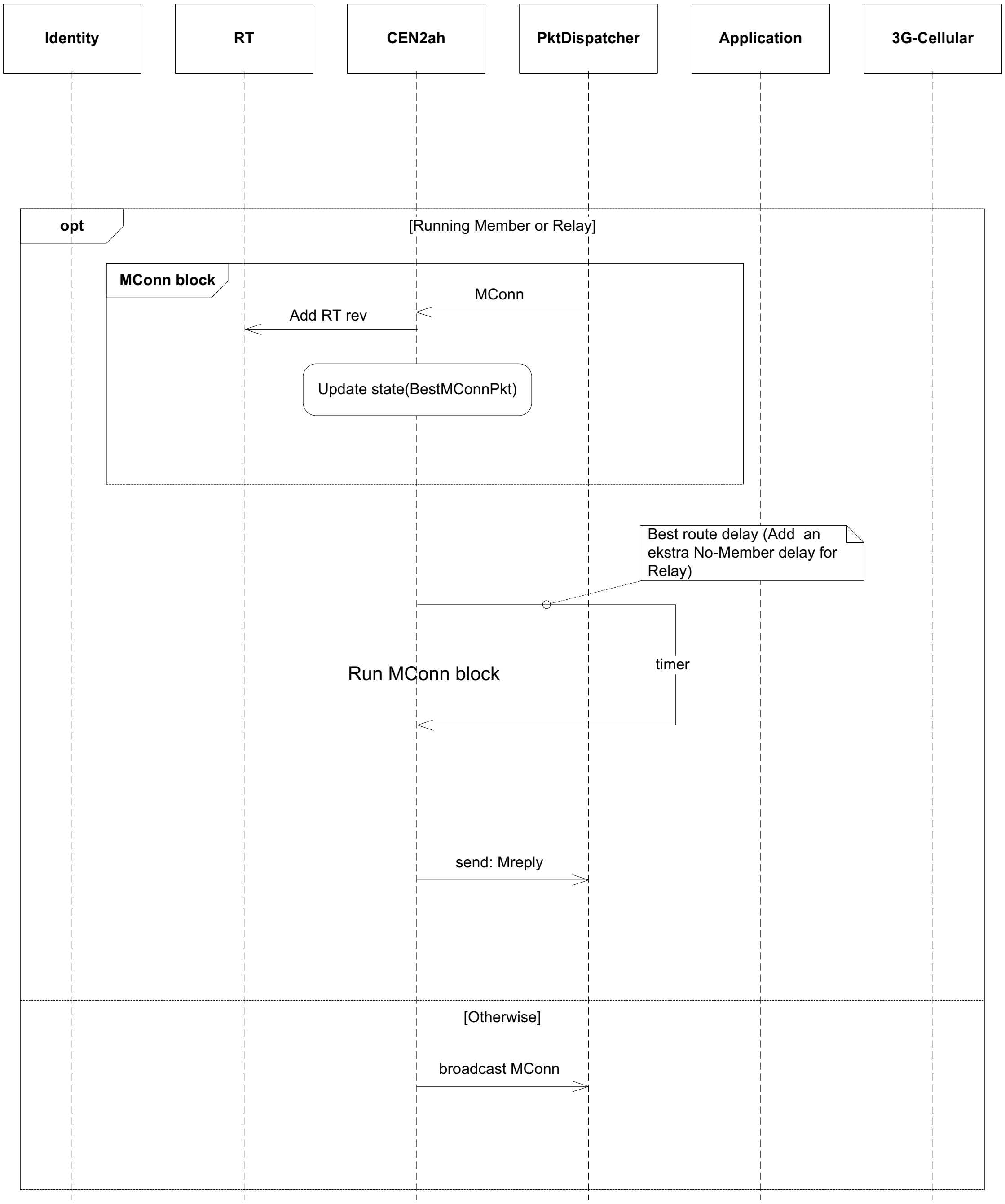


New member receive no instruction: Wait for MQuery

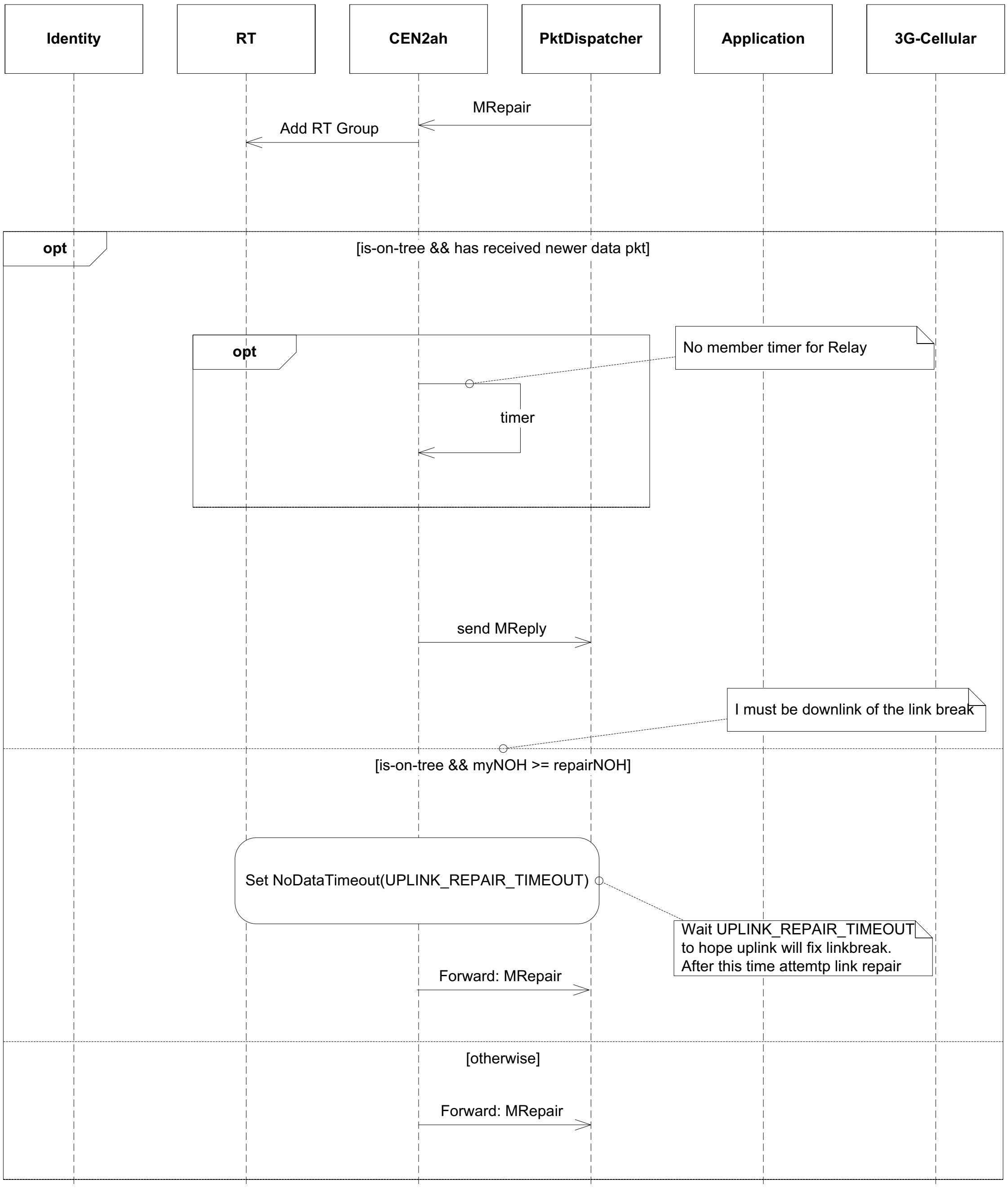



Receive MQuery
Identity
RT
CEN2ah
PktDispatcher
Application
3G-Cellular
MQuery
Add RT rev
opt
[Not Member or Relay]
NO_MEMBER_DELAY
Broadcast: MQuery
[Member]
ref: New member, no Instr

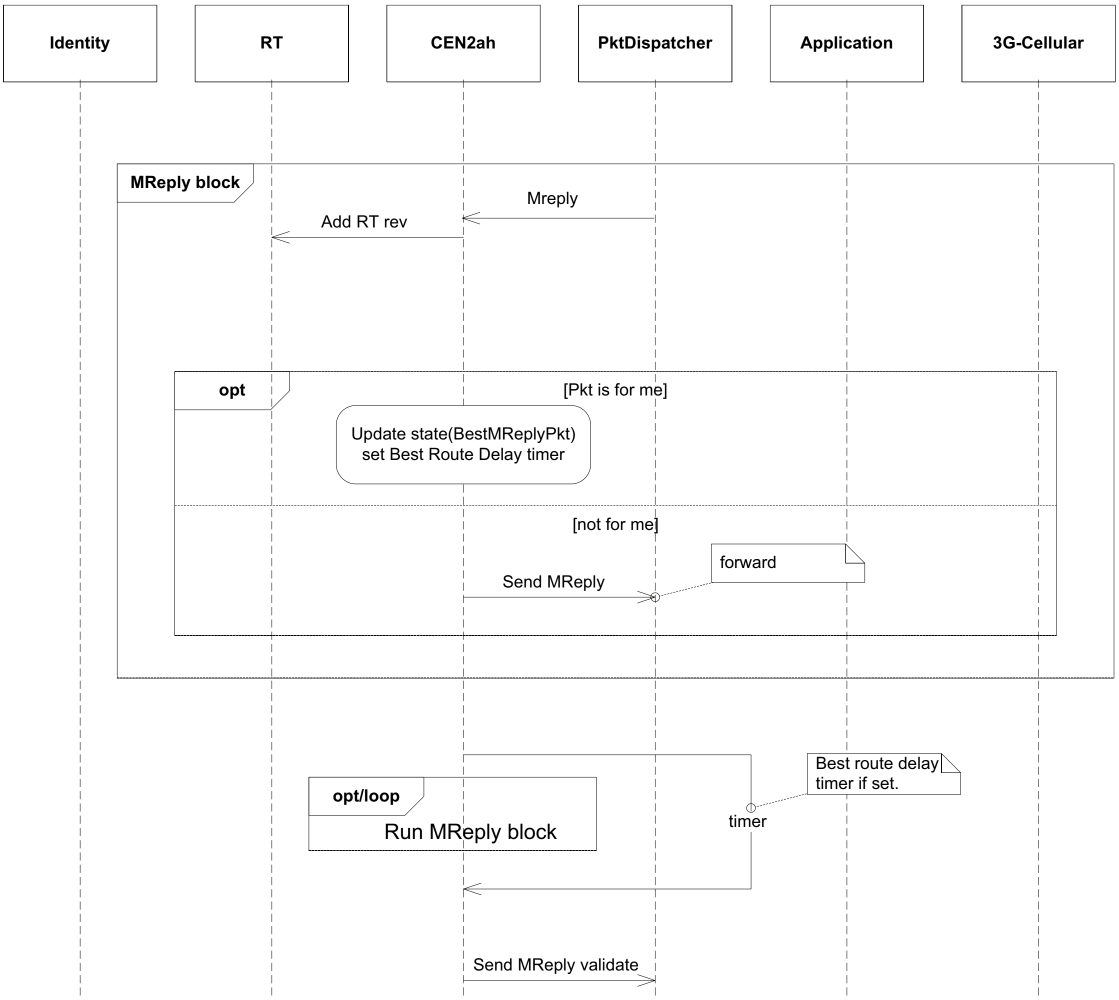
Receive MConn



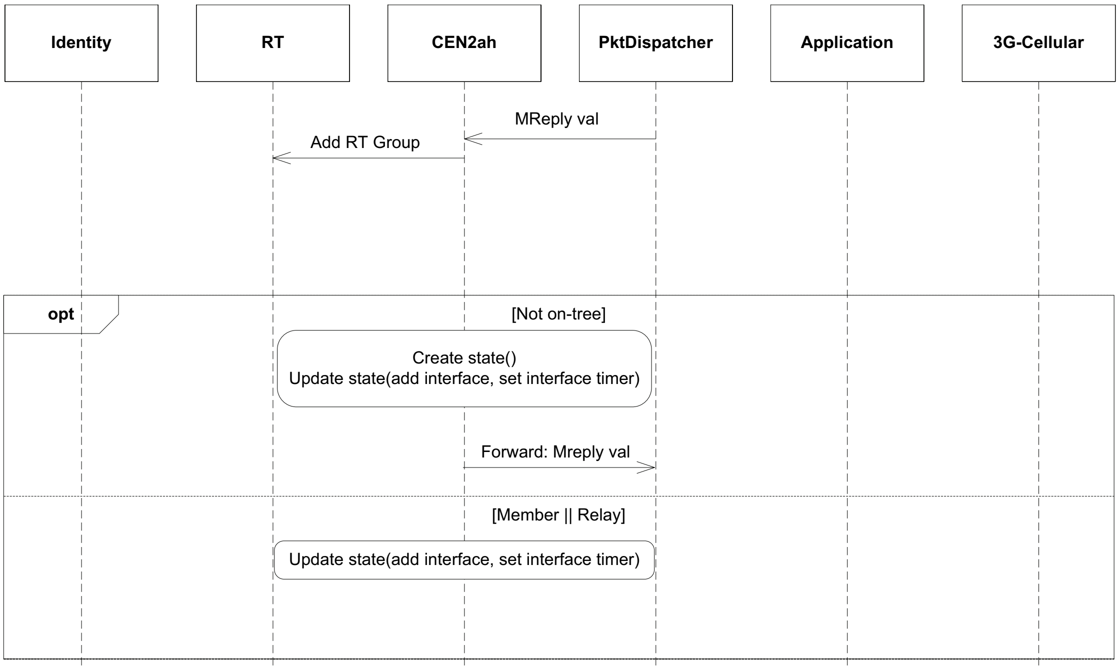
Recv MRepair


Identity
RT
CEN2ah
PktDispatcher
Application
3G-Cellular
MRepair
Add RT Group
opt
[is-on-tree && has received newer data pkt]
opt
No member timer for Relay
timer
send MReply
I must be downlink of the link break
[is-on-tree && myNOH >= repairNOH]
Set NoDataTimeout(UPLINK_REPAIR_TIMEOUT)
Wait UPLINK_REPAIR_TIMEOUT
to hope uplink will fix linkbreak.
After this time attemtp link repair
Forward: MRepair
[otherwise]
Forward: MRepair

Recv MReply


Identity
RT
CEN2ah
PktDispatcher
Application
3G-Cellular
MReply block
Mreply
Add RT rev
opt
[Pkt is for me]
Update state(BestMReplyPkt)
set Best Route Delay timer
[not for me]
forward
Send MReply
Best route delay
timer if set.
opt/loop
Run MReply block
timer
Send MReply validate

Recv MReply Validate


Identity
RT
CEN2ah
PktDispatcher
Application
3G-Cellular
MReply val
Add RT Group
opt
[Not on-tree]
Create state()
Update state(add interface, set interface timer)
Forward: Mreply val
[Member || Relay]
Update state(add interface, set interface timer)