DMM Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 2, 2019

CJ. Bernardos
A. de la Oliva
UC3M
F. Giust
Athonet
JC. Zuniga
SIGFOX
A. Mourad
InterDigital
August 29, 2018

Proxy Mobile IPv6 extensions for Distributed Mobility Management
draft-ietf-dmm-pmipv6-dlif-02

Abstract

Distributed Mobility Management solutions allow for setting up networks so that traffic is distributed in an optimal way and does not rely on centrally deployed anchors to provide IP mobility support.

There are many different approaches to address Distributed Mobility Management, as for example extending network-based mobility protocols (like Proxy Mobile IPv6), or client-based mobility protocols (like Mobile IPv6), among others. This document follows the former approach and proposes a solution based on Proxy Mobile IPv6 in which mobility sessions are anchored at the last IP hop router (called mobility anchor and access router). The mobility anchor and access router is an enhanced access router which is also able to operate as a local mobility anchor or mobility access gateway, on a per prefix basis. The document focuses on the required extensions to effectively support simultaneously anchoring several flows at different distributed gateways.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Table of Contents

1. Introduction

The Distributed Mobility Management (DMM) paradigm aims at minimizing the impact of currently standardized mobility management solutions which are centralized (at least to a considerable extent).

Current IP mobility solutions, standardized with the names of Mobile IPv6 [RFC6275], or Proxy Mobile IPv6 (PMIPv6) [RFC5213], just to cite the two most relevant examples, offer mobility support at the cost of handling operations at a cardinal point, the mobility anchor (i.e., the home agent for Mobile IPv6, and the local mobility anchor for Proxy Mobile IPv6), and burdening it with data forwarding and control mechanisms for a great amount of users. As stated in [RFC7333], centralized mobility solutions are prone to several problems and limitations: longer (sub-optimal) routing paths, scalability problems, signaling overhead (and most likely a longer associated handover latency), more complex network deployment, higher vulnerability due to the existence of a potential single point of failure, and lack of granularity of the mobility management service (i.e., mobility is offered on a per-node basis, not being possible to define finer granularity policies, as for example per-application).

The purpose of Distributed Mobility Management is to overcome the limitations of the traditional centralized mobility management [RFC7333] [RFC7429]; the main concept behind DMM solutions is indeed bringing the mobility anchor closer to the Mobile Node (MN). Following this idea, in our proposal, the central anchor is moved to the edge of the network, being deployed in the default gateway of the mobile node. That is, the first elements that provide IP connectivity to a set of MNs are also the mobility managers for those

MNs. In the following, we will call these entities Mobility Anchor and Access Routers (MAARs).

This document focuses on network-based DMM, hence the starting point is making PMIPv6 working in a distributed manner [RFC7429]. Mobility is handled by the network without the MNs involvement, but, differently from PMIPv6, when the MN moves from one access network to another, it also changes anchor router, hence requiring signaling between the anchors to retrieve the MN's previous location(s). Also, a key-aspect of network-based DMM, is that a prefix pool belongs exclusively to each MAAR, in the sense that those prefixes are assigned by the MAAR to the MNs attached to it, and they are routable at that MAAR.

We consider partially distributed schemes, where the data plane only is distributed among access routers similar to MAGs, whereas the control plane is kept centralized towards a cardinal node used as information store, but relieved from any route management and MN's data forwarding task.

2. Terminology

The following terms used in this document are defined in the Proxy Mobile IPv6 specification [RFC5213]:

Local Mobility Anchor (LMA)

Mobile Access Gateway (MAG)

Mobile Node (MN)

Binding Cache Entry (BCE)

Proxy Care-of Address (P-CoA)

Proxy Binding Update (PBU)

Proxy Binding Acknowledgement (PBA)

The following terms used in this document are defined in the DMM Deployment Models and Architectural Considerations document [I-D.ietf-dmm-deployment-models]:

Home Control-Plane Anchor (Home-CPA)

Home Data Plane Anchor (Home-DPA)

Access Control Plane Node (Access-CPN)

Access Data Plane Node (Access-DPN)

The following terms are defined and used in this document:

MAAR (Mobility Anchor and Access Router). First hop router where the mobile nodes attach to. It also plays the role of mobility manager for the IPv6 prefixes it anchors, running the functionalities of PMIP's MAG and LMA. Depending on the prefix, it plays the role of Access-DPN, Home-DPA and Access-CPN.

CMD (Central Mobility Database). Node that stores the BCEs allocated for the MNs in the mobility domain. It plays the role of Home-CPA.

P-MAAR (Previous MAAR). MAAR which was previously visited by the MN and is still involved in an active flow using an IPv6 prefix it has advertised to the MN (i.e., MAAR where that IPv6 prefix is anchored). It plays the role of Home-DPA for the flows it is still serving for the MN's mobility session. There might be multiple P-MAARs for an MN's mobility session.

S-MAAR (Serving MAAR). MAAR which the MN is currently attached to. Depending on the prefix, it plays the role of Access-DPN, Home-DPA and Access-CPN.

DLIF (Distributed Logical Interface). It is a logical interface at the IP stack of the MAAR. For each active prefix used by the mobile node, the S-MAAR has a DLIF configured (associated to each MAAR still anchoring flows). In this way, a S-MAAR exposes itself towards each MN as multiple routers, one as itself and one per P-MAAR.

3. PMIPv6 DMM extensions

The solution consists of de-coupling the entities that participate in the data and the control planes: the data plane becomes distributed and managed by the MAARs near the edge of the network, while the control plane, besides those on the MAARs, relies on a central entity called Central Mobility Database (CMD). In the proposed architecture, the hierarchy present in PMIPv6 between LMA and MAG is preserved, but with the following substantial variations:

o The LMA is relieved from the data forwarding role, only the Binding Cache and its management operations are maintained. Hence the LMA is renamed into Central Mobility Database (CMD), which is therefore an Home-CPA. Also, the CMD is able to send and parse both PBU and PBA messages.

- The MAG is enriched with the LMA functionalities, hence the name Mobility Anchor and Access Router (MAAR). It maintains a local Binding Cache for the MNs that are attached to it and it is able to send and parse PBU and PBA messages.

- The binding cache will be extended to include information regarding P-MAARs where the mobile node was anchored and still retains active data sessions, see Appendix B for further details.

- Each MAAR has a unique set of global prefixes (which are configurable), that can be allocated by the MAAR to the MNs, but must be exclusive to that MAAR, i.e. no other MAAR can allocate the same prefixes.

The MAARs leverage the Central Mobility Database (CMD) to access and update information related to the MNs, stored as mobility sessions; hence, a centralized node maintains a global view of the network status. The CMD is queried whenever a MN is detected to join/leave the mobility domain. It might be a fresh attachment, a detachment or a handover, but as MAARs are not aware of past information related to a mobility session, they contact the CMD to retrieve the data of interest and eventually take the appropriate action. The procedure adopted for the query and the messages exchange sequence might vary to optimize the update latency and/or the signaling overhead. Here is presented one method for the initial registration, and three different approaches to update the mobility sessions using PBUs and PBAs. Each approach assigns a different role to the CMD:

- The CMD is a PBU/PBA relay;

- The CMD is only a MAAR locator;

- The CMD is a PBU/PBA proxy.

This solution can be categorized under Model-1: Split Home Anchor Mode in [I-D.ietf-dmm-deployment-models]. As another note, the solution described in this document allows performing per-prefix anchoring decisions, to support e.g., some flows to be anchored at a central Home-DPA (like a traditional LMA) or to enable an application to switch to the locally anchored prefix to gain route optimization, as indicated in [I-D.ietf-dmm-ondemand-mobility].

Note that a MN MAY move across different MAARs, which might result in several P-MAARs existing at a given moment of time, each of them anchoring a prefix used by the MN.

3.1. Initial registration

Upon the MN's attachment to a MAAR, say MAAR1 as shown in Figure 1, if the MN is authorized for the service, an IPv6 global prefix belonging to the MAAR1's prefix pool is reserved for it (Pref1) into a temporary Binding Cache Entry (BCE) allocated locally. The prefix is sent in a [RFC5213] PBU with the MN's Identifier (MN-ID) to the CMD, which, since the session is new, stores a permanent BCE containing as primary fields the MN-ID, the MN's prefix and MAAR1's address as Proxy-CoA. The CMD replies to MAAR1 with a PBA including the usual options defined in PMIPv6 [RFC5213], meaning that the MN's registration is fresh and no past status is available. MAAR1 stores the temporary BCE previously allocated and unicasts a Router Advertisement (RA) to the MN including the prefix reserved before, that can be used by the MN to configure an IPv6 address (e.g., with stateless auto-configuration, SLAAC). Alternative IPv6 auto-configuration mechanisms can also be used, though this document describes the SLAAC-based one. The address is routable at the MAAR, in the sense that it is on the path of packets addressed to the MN. Moreover, the MAAR acts as plain router for those packets, as no encapsulation nor special handling takes place.

```
     +-----+      +---+                +--+
     |MAAR1|      |CMD|                |CN|
     +-----+      +---+                +*-+
        |           |                   *
       MN           |                   *     +---+
     attach.        |               *****    _|CMD|_
   detection        |         flow1 *       / +-+-+ \
        |           |               *      /    |    \
    local BCE       |               *     /     |     \
    allocation      |               *    /      |      \
        |--- PBU -->|           +---*-+-'    +--+--+    `+-----+
        |          BCE          |   * |      |     |     |     |
        |        creation       |MAAR1+------+MAAR2+-----+MAAR3|
        |<-- PBA ---|           |   * |      |     |     |     |
    local BCE       |           +---*-+      +-----+     +-----+
    finalized       |               *
        |           |         Pref1 *
        |           |              +*-+
        |           |              |MN|
        |           |              +--+

     Operations sequence                  Packets flow
```

Figure 1: First attachment to the network

Note that the registration process does not change regardless of the CMD's modes (relay, locator or proxy) described next.

3.2. The CMD as PBU/PBA relay

When the MN moves from its current access and attaches to MAAR2 (now the S-MAAR), MAAR2 reserves another IPv6 prefix (Pref2), it stores a temporary BCE, and it sends a plain PBU to the CMD for registration. Upon PBU reception and BC lookup, the CMD retrieves an already existing entry for the MN, binding the MN-ID to its former location; thus, the CMD forwards the PBU to the MAAR indicated as Proxy CoA (MAAR1), including a new mobility option to communicate the S-MAAR's global address to MAAR1, defined as Serving MAAR Option in Section 4.6. The CMD updates the P-CoA field in the BCE related to the MN with the S-MAAR's address.

Upon PBU reception, MAAR1 can install a tunnel on its side towards MAAR2 and the related routes for Pref1. Then MAAR1 replies to the CMD with a PBA (including the option mentioned before) to ensure that the new location has successfully changed, containing the prefix anchored at MAAR1 in the Home Network Prefix option. The CMD, after receiving the PBA, updates the BCE populating an instance of the P-MAAR list. The P-MAAR list is an additional field on the BCE that contains an element for each P-MAAR involved in the MN's mobility session. The list element contains the P-MAAR's global address and the prefix it has delegated (see Appendix B for further details). Also, the CMD sends a PBA to the new S-MAAR, containing the previous Proxy-CoA and the prefix anchored to it embedded into a new mobility option called Previous MAAR Option (defined in Section 4.5), so that, upon PBA arrival, a bi-directional tunnel can be established between the two MAARs and new routes are set appropriately to recover the IP flow(s) carrying Pref1.

Now packets destined to Pref1 are first received by MAAR1, encapsulated into the tunnel and forwarded to MAAR2, which finally delivers them to their destination. In uplink, when the MN transmits packets using Pref1 as source address, they are sent to MAAR2, as it is MN's new default gateway, then tunneled to MAAR1 which routes them towards the next hop to destination. Conversely, packets carrying Pref2 are routed by MAAR2 without any special packet handling both for uplink and downlink. The procedure is depicted in Figure 2.

```
   +-----+      +---+      +-----+          +--+            +--+
   |MAAR1|      |CMD|      |MAAR2|          |CN|            |CN|
   +-----+      +---+      +-----+          +*-+            +*-+
      |           |           |               *               *
      |           |          MN               *     +---+     *
      |           |        attach.        *****    _|CMD|_    *
      |           |          det.   flow1 *       / +-+-+ \   *flow2
      |           |<-- PBU ---|           *      /    |    \  *
      |          BCE          |           *     /     | *******
      |        check+         |           *    /      | *    \
      |        update         |       +---*-+-'    +--+-*+    `+-----+
      |<-- PBU*---|           |       |   * |      |    *|     |     |
   route          |           |       |MAAR1|______|MAAR2+-----+MAAR3|
   update         |           |       |   **(______)**  *|     |     |
      |--- PBA*-->|           |       +-----+      +-*--*+     +-----+
      |         BCE           |                     *  *
      |        update         |                Pref1 *  *Pref2
      |           |--- PBA*-->|                    +*--*+
      |           |         route        ---move-->|*MN*|
      |           |         update                 +----+

         Operations sequence                 Data Packets flow
   PBU/PBA Messages with * contain
        a new mobility option
```

Figure 2: Scenario after a handover, CMD as relay

For MN's next movements the process is repeated except the number of P-MAARs involved increases, that rises accordingly to the number of prefixes that the MN wishes to maintain. Indeed, once the CMD receives the first PBU from the new S-MAAR, it forwards copies of the PBU to all the P-MAARs indicated in the BCE as current P-CoA (i.e., the MAAR prior to handover) and in the P-MAARs list. They reply with a PBA to the CMD, which aggregates them into a single one to notify the S-MAAR, that finally can establish the tunnels with the P-MAARs.

It should be noted that this design separates the mobility management at the prefix granularity, and it can be tuned in order to erase old mobility sessions when not required, while the MN is reachable through the latest prefix acquired. Moreover, the latency associated to the mobility update is bound to the PBA sent by the furthest P-MAAR, in terms of RTT, that takes the longest time to reach the CMD. The drawback can be mitigated introducing a timeout at the CMD, by which, after its expiration, all the PBAs so far collected are transmitted, and the remaining are sent later upon their arrival.

3.3. The CMD as MAAR locator

The handover latency experienced in the approach shown before can be reduced if the P-MAARs are allowed to signal directly their information to the new S-MAAR. This procedure reflects what was described in Section 3.2 up to the moment the P-MAAR receives the PBU with the P-MAAR option. At that point a P-MAAR is aware of the new MN's location (because of the S-MAAR's address in the S-MAAR option), and, besides sending a PBA to the CMD, it also sends a PBA to the S-MAAR including the prefix it is anchoring. This latter PBA does not need to include new options, as the prefix is embedded in the HNP option and the P-MAAR's address is taken from the message's source address. The CMD is relieved from forwarding the PBA to the S-MAAR, as the latter receives a copy directly from the P-MAAR with the necessary information to build the tunnels and set the appropriate routes. Figure 3 illustrates the new message sequence, while the data forwarding is unaltered.

```
   +-----+      +---+      +-----+           +--+            +--+
   |MAAR1|      |CMD|      |MAAR2|           |CN|            |CN|
   +-----+      +---+      +-----+           +*-+            +*-+
      |           |           |               *               *
      |           |          MN               *     +---+     *
      |           |        attach.        *****    _|CMD|_    *
      |           |          det.   flow1 *       / +-+-+ \   *flow2
      |           |<-- PBU ---|           *      /    |    \  *
      |          BCE          |           *     /     | *******
      |        check+         |           *    /      | *    \
      |        update         |       +---*-+-'    +--+-*+    `+-----+
      |<-- PBU*---|           |       |   * |      |    *|     |     |
   route          |           |       |MAAR1|______|MAAR2+-----+MAAR3|
   update         |           |       |   **(______)**  *|     |     |
      |--------- PBA -------->|       +-----+      +-*--*+     +-----+
      |--- PBA*-->|         route              *  *
      |          BCE        update            Pref1 *  *Pref2
      |         update        |                    +*--*+
      |           |           |           ---move-->|*MN*|
      |           |           |                    +----+

          Operations sequence                 Data Packets flow
   PBU/PBA Messages with * contain
        a new mobility option
```

Figure 3: Scenario after a handover, CMD as locator

3.4. The CMD as MAAR proxy

A further enhancement of previous solutions can be achieved when the CMD sends the PBA to the new S-MAAR before notifying the P-MAARs of the location change. Indeed, when the CMD receives the PBU for the new registration, it is already in possession of all the information that the new S-MAAR requires to set up the tunnels and the routes. Thus the PBA is sent to the S-MAAR immediately after a PBU is received, including also in this case the P-MAAR option. In parallel, a PBU is sent by the CMD to the P-MAARs containing the S-MAAR option, to notify them about the new MN's location, so they receive the information to establish the tunnels and routes on their side. When P-MAARs complete the update, they send a PBA to the CMD to indicate that the operation is concluded and the information are updated in all network nodes. This procedure is obtained from the first one re-arranging the order of the messages, but the parameters communicated are the same. This scheme is depicted in Figure 4, where, again, the data forwarding is kept untouched.

```
   +-----+      +---+      +-----+           +--+            +--+
   |MAAR1|      |CMD|      |MAAR2|           |CN|            |CN|
   +-----+      +---+      +-----+           +*-+            +*-+
      |           |           |               *               *
      |           |          MN               *     +---+     *
      |           |        attach.        *****    _|CMD|_    *
      |           |          det.   flow1 *       / +-+-+ \   *flow2
      |           |<-- PBU ---|           *      /    |    \  *
      |          BCE          |           *     /     | *******
      |        check+         |           *    /      | *    \
      |        update         |       +---*-+-'    +--+-*+    `+-----+
      |<-- PBU*---x--- PBA*-->|       |   * |      |    *|     |     |
   route          |         route     |MAAR1|______|MAAR2+-----+MAAR3|
   update         |         update    |   **(______)**  *|     |     |
      |--- PBA*-->|           |       +-----+      +-*--*+     +-----+
      |          BCE          |                      *  *
      |         update        |                Pref1 *  *Pref2
      |           |           |                     +*--*+
      |           |           |           ---move-->|*MN*|
      |           |           |                     +----+

          Operations sequence                 Data Packets flow
   PBU/PBA Messages with * contain
        a new mobility option
```

Figure 4: Scenario after a handover, CMD as proxy

3.5. De-registration

The de-registration mechanism devised for PMIPv6 cannot be used as is in this solution. This is motivated by the fact that each MAAR handles an independent mobility session (i.e., a single or a set of prefixes) for a given MN, whereas the aggregated session is stored at the CMD. Indeed, when a previous MAAR initiates a de-registration procedure, because the MN is no longer present on the MAAR's access link, it removes the routing state for that (those) prefix(es), that would be deleted by the CMD as well, hence defeating any prefix continuity attempt. The simplest approach to overcome this limitation is to deny an old MAAR to de-register a prefix, that is, allowing only a serving MAAR to de-register the whole MN session. This can be achieved by first removing any layer-2 detachment event, so that de-registration is triggered only when the session lifetime expires, hence providing a guard interval for the MN to connect to a new MAAR. Then, a change in the MAAR operations is required, and at this stage two possible solutions can be deployed:

- A previous MAAR stops the BCE timer upon receiving a PBU from the CMD containing a "Serving MAAR" option. In this way only the Serving MAAR is allowed to de-register the mobility session, arguing that the MN left definitely the domain.

- Previous MAARs can, upon BCE expiry, send de-registration messages to the CMD, which, instead of acknowledging the message with a 0 lifetime, send back a PBA with a non-zero lifetime, hence re-newing the session, if the MN is still connected to the domain.

3.6. The Distributed Logical Interface (DLIF) concept

One of the main challenges of a network-based DMM solution is how to allow a mobile node to simultaneously send/receive traffic which is anchored at different MAARs, and how to influence on the preference of the mobile node selecting the source IPv6 address for a new communication, without requiring special support on the mobile node stack. This document defines the Distributed Logical Interface (DLIF), which is a software construct that allows to easily hide the change of anchor from the mobile node.

```
     +----------------------------------------------------+
    (                      Operator's                      )
    (                         core                         )
     +----------------------------------------------------+
               |                               |
       +---------------+     tunnel    +---------------+
       |   IP  stack   |===============|   IP  stack   |
       +---------------+               +-------+-------+
       |    mn1mar1    |--+ (DLIFs) +--|mn1mar1|mn1mar2|--+
       +---------------+  |         |  +-------+-------+  |
       | phy interface |  |         |  | phy interface |  |
       +---------------+  |         |  +---------------+  |
             MAAR1       (o)       (o)       MAAR2       (o)
                                      x                 x
                                        x             x
                           prefA::/64     x         x   prefB::/64
                         (AdvPrefLft=0)     x     x
                                             (o)
                                              |
                                            +-----+
                                prefA::MN1  | MN1 |  prefB::MN1
                               (deprecated) +-----+
```

        Figure 5: DLIF: exposing multiple routers (one per P-MAAR)

The basic idea of the DLIF concept is the following. Each serving MAAR exposes itself towards a given MN as multiple routers, one per P-MAAR associated to the MN. Let's consider the example shown in Figure 5, MN1 initially attaches to MAAR1, configuring an IPv6 address (prefA::MN1) from a prefix locally anchored at MAAR1 (prefA::/64). At this stage, MAAR1 plays both the role of anchoring and serving MAAR, and also it behaves as a plain IPv6 access router. MAAR1 creates a distributed logical interface to communicate (point-to-point link) with MN1, exposing itself as a (logical) router with a specific MAC (e.g., 00:11:22:33:01:01) and IPv6 addresses (e.g., prefA::MAAR1/64 and fe80:211:22ff:fe33:101/64) using the DLIF mn1mar1. As explained below, these addresses represent the "logical" identity of MAAR1 towards MN1, and will "follow" the mobile node while roaming within the domain (note that the place where all this information is maintained and updated is out-of-scope of this draft; potential examples are to keep it on the home subscriber server -- HSS -- or the user's profile).

If MN1 moves and attaches to a different MAAR of the domain (MAAR2 in the example of Figure 5), this MAAR will create a new logical interface (mn1mar2) to expose itself towards MN1, providing it with a locally anchored prefix (prefB::/64). In this case, since the MN1 has another active IPv6 address anchored at a MAAR1, MAAR2 also needs

to create an additional logical interface configured to exactly resemble the one used by MAAR1 to communicate with MN1. In this example, there is only one P-MAAR (in addition to MAAR2, which is the serving one): MAAR1, so only the logical interface mn1mar1 is created, but the same process would be repeated in case there were more P-MAARs involved. In order to maintain the prefix anchored at MAAR1 reachable, a tunnel between MAAR1 and MAAR2 is established and the routing is modified accordingly. The PBU/PBA signaling is used to set-up the bi-directional tunnel between MAAR1 and MAAR2, and it might also be used to convey to MAAR2 the information about the prefix(es) anchored at MAAR1 and about the addresses of the associated DLIF (i.e., mn1mar1).

```
+------------------------------------------+ +----------------------+
|                  MAAR1                   | |        MAAR2         |
|+----------------------------------------+| |+--------------------+|
||+------------------++------------------+|| ||+------------------+||
|||+-------++-------+||+-------++-------+||| |||+-------++-------+|||
||||mn3mar1||mn3mar2||||mn2mar1||mn2mar2|||| ||||mn1mar1||mn1mar2||||
|||| LMAC1 || LMAC2 |||| LMAC3 || LMAC4 |||| |||| LMAC5 || LMAC6 ||||
|||+-------++-------+||+-------++-------+||| |||+-------++-------+|||
|||    LIFs of MN3   ||    LIFs of MN2   ||| |||   LIFs of MN1    |||
||+------------------++------------------+|| ||+------------------+||
||             HMAC1   (phy if MAAR1)    || ||HMAC2 (phy if MAAR2)||
|+----------------------------------------+| |+--------------------+|
+------------------------------------------+ +----------------------+
                    x        x                             x
                   x          x                           x
                 (o)          (o)                       (o)
                  |            |                         |
               +--+--+      +--+--+                   +--+--+
               | MN3 |      | MN2 |                   | MN1 |
               +-----+      +-----+                   +-----+
```

Figure 6: Distributed Logical Interface concept

Figure 6 shows the logical interface concept in more detail. The figure shows two MAARs and three MNs. MAAR1 is currently serving MN2 and MN3, while MAAR2 is serving MN1. MN1, MN2 and MN3 have two P-MAARs: MAAR1 and MAAR2. Note that a serving MAAR always plays the role of anchoring MAAR for the attached (served) MNs. Each MAAR has one single physical wireless interface.

As introduced before, each MN always "sees" multiple logical routers -- one per P-MAAR -- independently of to which serving MAAR the MN is currently attached. From the point of view of the MN, these MAARs are portrayed as different routers, although the MN is physically attached to one single interface. The way this is achieved is by the

serving MAAR configuring different logical interfaces. If we focus on MN1, it is currently attached to MAAR2 (i.e., MAAR2 is its serving MAAR) and, therefore, it has configured an IPv6 address from MAAR2's pool (e.g., prefB::/64). MAAR2 has set-up a logical interface (mn1mar2) on top of its wireless physical interface (phy if MAAR2) which is used to serve MN1. This interface has a logical MAC address (LMAC6), different from the hardware MAC address (HMAC2) of the physical interface of MAAR2. Over the mn1mar2 interface, MAAR2 advertises its locally anchored prefix prefB::/64. Before attaching to MAAR2, MN1 visited MAAR1, configuring also an address locally anchored at this MAAR, which is still being used by the MN1 in active communications. MN1 keeps "seeing" an interface connecting to MAAR1, as if it were directly connected to the two MAARs. This is achieved by the serving MAAR (MAAR2) configuring an additional distributed logical interface: mn1mar1, which behaves exactly as the logical interface configured by the actual MAAR1 when MN1 was attached to it. This means that both the MAC and IPv6 addresses configured on this logical interface remain the same regardless of the physical MAAR which is serving the MN. The information required by a serving MAAR to properly configure this logical interfaces can be obtained in different ways: as part of the information conveyed in the PBA, from an external database (e.g., the HSS) or by other means. As shown in the figure, each MAAR may have several logical interfaces associated to each attached MN, having always at least one (since a serving MAAR is also an anchoring MAAR for the attached MN).

In order to enforce the use of the prefix locally anchored at the serving MAAR, the router advertisements sent over those logical interfaces playing the role of anchoring MAARs (different from the serving one) include a zero preferred prefix lifetime (and a non-zero valid prefix lifetime, so the prefix remains valid, while being deprecated). The goal is to deprecate the prefixes delegated by these MAARs (which will be no longer serving the MN). Note that on-going communications keep on using those addresses, even if they are deprecated, so this only affects to new sessions.

The distributed logical interface concept also enables the following use case. Suppose that access to a local IP network is provided by a given MAAR (e.g., MAAR1 in the example shown in Figure 5) and that the resources available at that network cannot be reached from outside the local network (e.g., cannot be accessed by an MN attached to MAAR2). This is similar to the local IP access scenario considered by 3GPP, where a local gateway node is selected for sessions requiring access to services provided locally (instead of going through a central gateway). The goal is to allow an MN to be able to roam while still being able to have connectivity to this local IP network. The solution adopted to support this case makes use of RFC 4191 [RFC4191] more specific routes when the MN moves to a

MAAR different from the one providing access to the local IP network (MAAR1 in the example). These routes are advertised through the distributed logical interface representing the MAAR providing access to the local network (MAAR1 in this example). In this way, if MN1 moves from MAAR1 to MAAR2, any active session that MN1 may have with a node of the local network connected to MAAR1 will survive, being the traffic forwarded via the tunnel between MAAR1 and MAAR2. Also, any potential future connection attempt towards the local network will be supported, even though MN1 is no longer attached to MAAR1.

4. Message Format

This section defines extensions to the Proxy Mobile IPv6 [RFC5213] protocol messages.

4.1. Proxy Binding Update

A new flag (D) is included in the Proxy Binding Update to indicate that the Proxy Binding Update is coming from a Mobility Anchor and Access Router and not from a mobile access gateway. The rest of the Proxy Binding Update format remains the same as defined in [RFC5213].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |            Sequence #         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|H|L|K|M|R|P|D| Reserved      |            Lifetime           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                         Mobility options                      .
.                                                               .

|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

MAAR Flag (D)

The D Flag is set to indicate to the receiver of the message that the Proxy Binding Update is from a MAAR. When an LMA that does not support the extensions described in this document receives a message with the D-Flag set, the PBU in that case MUST NOT be processed by the LMA and an error MUST be returned.

Mobility Options

Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The encoding and format of defined options are described in Section 6.2 of [RFC6275]. The MAAR MUST ignore and skip any options that it does not understand.

4.2. Proxy Binding Acknowledgment

A new flag (D) is included in the Proxy Binding Acknowledgment to indicate that the sender supports operating as a Mobility Anchor and Access Router. The rest of the Proxy Binding Acknowledgment format remains the same as defined in [RFC5213].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |   Status      |K|R|P|D| Reser |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Sequence #            |           Lifetime            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                        Mobility options                       .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

MAAR (D)

The D is set to indicate that the sender of the message supports operating as a Mobility Anchor and Access Router. When a MAG that does not support the extensions described in this document receives a message with the D-Flag set, the PBA in that case MUST NOT be processed by the MAG and an error MUST be returned.

Mobility Options

Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The encoding and format of defined options are described in Section 6.2 of [RFC6275]. The MAAR MUST ignore and skip any options that it does not understand.

4.3. Anchored Prefix Option

A new Anchored Prefix option is defined for use with the Proxy Binding Update and Proxy Binding Acknowledgment messages exchanged between MAARs and CMDs. Therefore, this option can only appear if the D bit is set in a PBU/PBA. This option is used for exchanging the mobile node's prefix anchored at the anchoring MAAR. There can be multiple Anchored Prefix options present in the message.

The Anchored Prefix Option has an alignment requirement of 8n+4. Its format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |   Length      |   Reserved    | Prefix Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                        Anchored Prefix                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

TBD1.

Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 18.

Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

Prefix Length

8-bit unsigned integer indicating the prefix length of the IPv6 prefix contained in the option.

Anchored Prefix

A sixteen-byte field containing the mobile node's IPv6 Anchored Prefix.

4.4. Local Prefix Option

A new Local Prefix option is defined for use with the Proxy Binding Update and Proxy Binding Acknowledgment messages exchanged between MAARs. Therefore, this option can only appear if the D bit is set in a PBU/PBA. This option is used for exchanging a prefix of a local network that is only reachable via the anchoring MAAR. There can be multiple Local Prefix options present in the message.

The Local Prefix Option has an alignment requirement of 8n+4. Its format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |   Length      |   Reserved    | Prefix Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                          Local Prefix                         +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

TBD2.

Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 18.

Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

Prefix Length

8-bit unsigned integer indicating the prefix length of the IPv6 prefix contained in the option.

Local Prefix

A sixteen-byte field containing the IPv6 Local Prefix.

4.5. Previous MAAR Option

This new option is defined for use with the Proxy Binding Acknowledgement messages exchanged by the CMD to a MAAR. This option is used to notify the S-MAAR about the previous MAAR's global address and the prefix anchored to it. There can be multiple Previous MAAR options present in the message. Its format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                |      Type     |     Length    | Prefix Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      P-MAAR's address                         +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                     Home Network Prefix                       +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

TBD3.

Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 33.

Prefix Length

8-bit unsigned integer indicating the prefix length of the IPv6 prefix contained in the option.

Previous MAAR's address

A sixteen-byte field containing the P-MAAR's IPv6 global address.

Home Network Prefix

A sixteen-byte field containing the mobile node's IPv6 Home Network Prefix.

4.6. Serving MAAR Option

This new option is defined for use with the Proxy Binding Update and Proxy Binding Acknowledgement messages exchanged between the CMD and a Previous MAAR. This option is used to notify the P-MAAR about the current Serving MAAR's global address. Its format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |      Type     |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      S-MAAR's address                         +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

TBD4.

Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 16.

Serving MAAR's address

A sixteen-byte field containing the S-MAAR's IPv6 global address.

4.7. DLIF Link-local Address Option

A new DLIF Link-local Address option is defined for use with the Proxy Binding Update and Proxy Binding Acknowledgment messages exchanged between MAARs. This option is used for exchanging the link-local address of the DLIF to be configured on the serving MAAR so it resembles the DLIF configured on the P-MAAR.

The DLIF Link-local Address option has an alignment requirement of 8n+6. Its format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |   Type        |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                  DLIF Link-local Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

TBD5.

Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 16.

DLIF Link-local Address

A sixteen-byte field containing the link-local address of the logical interface.

4.8. DLIF Link-layer Address Option

A new DLIF Link-layer Address option is defined for use with the Proxy Binding Update and Proxy Binding Acknowledgment messages exchanged between MAARs. This option is used for exchanging the link-layer address of the DLIF to be configured on the serving MAAR so it resembles the DLIF configured on the P-MAAR.

The format of the DLIF Link-layer Address option is shown below. Based on the size of the address, the option MUST be aligned appropriately, as per mobility option alignment requirements specified in [RFC6275].

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Type        |    Length     |          Reserved             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    +                   DLIF Link-layer Address                     +
    .                              ...                              .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

TBD6.

Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields.

Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

DLIF Link-layer Address

A variable length field containing the link-layer address of the logical interface to be configured on the S-MAAR.

The content and format of this field (including byte and bit ordering) is as specified in Section 4.6 of [RFC4861] for carrying link-layer addresses. On certain access links, where the link-layer address is not used or cannot be determined, this option cannot be used.

5. IANA Considerations

This document defines new mobility options that require IANA actions.

TBD.

6. Security Considerations

The protocol extensions defined in this document share the same security concerns of Proxy Mobile IPv6 [RFC5213]. It is recommended that the signaling messages, Proxy Binding Update and Proxy Binding Acknowledgment, exchanged between the MAARs are protected using IPsec using the established security association between them. This essentially eliminates the threats related to the impersonation of a MAAR.

7. Acknowledgments

The authors would like to thank Marco Liebsch, Dirk von Hugo, Alex Petrescu, Daniel Corujo, Akbar Rahman, Danny Moses, Xinpeng Wei and Satoru Matsushima for their comments and discussion on the documents [I-D.bernardos-dmm-distributed-anchoring] and [I-D.bernardos-dmm-pmip] on which the present document is based.

The authors would also like to thank Lyle Bertz for his deep review of this document and his very valuable comments and suggestions.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <https://www.rfc-editor.org/info/rfc5213>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <https://www.rfc-editor.org/info/rfc6275>.

8.2. Informative References

[I-D.bernardos-dmm-distributed-anchoring]
Bernardos, C. and J. Zuniga, "PMIPv6-based distributed anchoring", draft-bernardos-dmm-distributed-anchoring-09 (work in progress), May 2017.

[I-D.bernardos-dmm-pmip]
Bernardos, C., Oliva, A., and F. Giust, "A PMIPv6-based solution for Distributed Mobility Management", draft-bernardos-dmm-pmip-09 (work in progress), September 2017.

[I-D.ietf-dmm-deployment-models]
Gundavelli, S. and S. Jeon, "DMM Deployment Models and Architectural Considerations", draft-ietf-dmm-deployment-models-04 (work in progress), May 2018.

[I-D.ietf-dmm-ondemand-mobility]
Yegin, A., Moses, D., Kweon, K., Lee, J., Park, J., and S. Jeon, "On Demand Mobility Management", draft-ietf-dmm-ondemand-mobility-15 (work in progress), July 2018.

[RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", RFC 7333, DOI 10.17487/RFC7333, August 2014, <https://www.rfc-editor.org/info/rfc7333>.

[RFC7429] Liu, D., Ed., Zuniga, JC., Ed., Seite, P., Chan, H., and CJ. Bernardos, "Distributed Mobility Management: Current Practices and Gap Analysis", RFC 7429, DOI 10.17487/RFC7429, January 2015, <https://www.rfc-editor.org/info/rfc7429>.

Appendix A. Comparison with Requirement document

In this section we describe how our solution addresses the DMM requirements listed in [RFC7333].

A.1. Distributed mobility management

"IP mobility, network access solutions, and forwarding solutions provided by DMM MUST enable traffic to avoid traversing a single mobility anchor far from the optimal route."

In our solution, a MAAR is responsible to handle the mobility for those IP flows started when the MN is attached to it. As long as the MN remains connected to the MAAR's access links, the IP packets of such flows can benefit from the optimal path. When the MN moves to

another MAAR, the path becomes non-optimal for ongoing flows, as they are anchored to the previous MAAR, but newly started IP sessions are forwarded by the new MAAR through the optimal path.

A.2. Bypassable network-layer mobility support for each application session

"DMM solutions MUST enable network-layer mobility, but it MUST be possible for any individual active application session (flow) to not use it. Mobility support is needed, for example, when a mobile host moves and an application cannot cope with a change in the IP address. Mobility support is also needed when a mobile router changes its IP address as it moves together with a host and, in the presence of ingress filtering, an application in the host is interrupted. However, mobility support at the network layer is not always needed; a mobile node can often be stationary, and mobility support can also be provided at other layers. It is then not always necessary to maintain a stable IP address or prefix for an active application session."

Our DMM solution operates at the IP layer, hence upper layers are totally transparent to the mobility operations. In particular, ongoing IP sessions are not disrupted after a change of access network. The routability of the old address is ensured by the IP tunnel with the old MAAR. New IP sessions are started with the new address. From the application's perspective, those processes which sockets are bound to a unique IP address do not suffer any impact. For the other applications, the sockets bound to the old address are preserved, whereas next sockets use the new address.

A.3. IPv6 deployment

"DMM solutions SHOULD target IPv6 as the primary deployment environment and SHOULD NOT be tailored specifically to support IPv4, particularly in situations where private IPv4 addresses and/or NATs are used."

The DMM solution we propose targets IPv6 only.

A.4. Existing mobility protocols

"A DMM solution MUST first consider reusing and extending IETF standard protocols before specifying new protocols."

This DMM solution is derived from the operations and messages specified in [RFC5213].

A.5. Coexistence with deployed networks/hosts and operability across different networks

"A DMM solution may require loose, tight, or no integration into existing mobility protocols and host IP stacks. Regardless of the integration level, DMM implementations MUST be able to coexist with existing network deployments, end hosts, and routers that may or may not implement existing mobility protocols. Furthermore, a DMM solution SHOULD work across different networks, possibly operated as separate administrative domains, when the needed mobility management signaling, forwarding, and network access are allowed by the trust relationship between them"

The partially distributed DMM solution (distributed data plane and centralized control plane) can be extended to provide a fallback mechanism to operate as legacy Proxy Mobile IPv6. It is necessary to instruct MAARs to always establish a tunnel with the same MAAR, working as LMA. The fully distributed DMM solution (distributed data and control plane) can be extended as well, but it requires more intervention. The partially distributed DMM solution can be deployed across different domains with trust agreements if the CMDs of the operators are enabled to transfer context from one node to another. The fully distributed DMM solution works across multiple domains if the same signalling scheme is used in both domains.

A.6. Operation and management considerations

"A DMM solution needs to consider configuring a device, monitoring the current operational state of a device, and responding to events that impact the device, possibly by modifying the configuration and storing the data in a format that can be analyzed later.

The proposed solution can re-use existing mechanisms defined for the operation and management of Proxy Mobile IPv6.

A.7. Security considerations

"A DMM solution MUST support any security protocols and mechanisms needed to secure the network and to make continuous security improvements. In addition, with security taken into consideration early in the design, a DMM solution MUST NOT introduce new security risks or amplify existing security risks that cannot be mitigated by existing security protocols and mechanisms."

The proposed solution does not specify a security mechanism, given that the same mechanism for PMIPv6 can be used.

A.8. Multicast

"DMM SHOULD enable multicast solutions to be developed to avoid network inefficiency in multicast traffic delivery."

This solution in its current version does not specify any support for multicast traffic, which is left for study in future versions.

Appendix B. Implementation experience

The network-based DMM solution described in section Section 3.4 is now available at the Open Distributed Mobility Management (ODMM) project (http://www.odmm.net/), under the name of Mobility Anchors Distribution for PMIPv6 (MAD-PMIPv6). The ODMM platform is intended to foster DMM development and deployment, by serving as a framework to host open source implementations.

The MAD-PMIPv6 code is developed in ANSI C from the existing UMIP implementation for PMIP. The most relevant changes with respect to the UMIP original version are related to how to create the CMD and MAAR's state machines from those of an LMA and a MAG; for this purpose, part of the LMA code was copied to the MAG, in order to send PBA messages and parse PBU. Also, the LMA routing functions were removed completely, and moved to the MAG, because MAARs need to route through the tunnels in downlink (as an LMA) and in uplink (as a MAG).

Tunnel management is hence a relevant technical aspect, as multiple tunnels are established by a single MAAR, which keeps their status directly into the MN's BCE. Indeed, from the implementation experience it was chosen to create an ancillary data structure as field within a BCE: the data structure is called "MAAR list" and stores the previous MAARs' address and the corresponding prefix(es) assigned for the MN. Only the CMD and the serving MAAR store this data structure, because the CMD maintains the global MN's mobility session formed during the MN's roaming within the domain, and the serving MAAR needs to know which previous MAARs were visited, the prefix(es) they assigned and the tunnels established with them. Conversely, a previous MAAR only needs to know which is the current Serving MAAR and establish a single tunnel with it. For this reason, a MAAR that receives a PBU from the CMD (meaning that the MN attached to another MAAR), first sets up the routing state for the MN's prefix(es) it is anchoring, then stops the BCE expiry timer and deletes the MAAR list (if present) since it is no longer useful.

In order to have the MN totally unaware of the changes in the access link, all MAARs implement the Distributed Logical Interface (DLIF) concept. Moreover, it should be noted that the protocols designed in the document work only at the network layer to handle the MNs joining

or leaving the domain. This should guarantee a certain independency to a particular access technology. The implementation reflects this reasoning, but we argue that an interaction with lower layers produces a more effective attachment and detachment detection, therefore improving the performance, also regarding de-registration mechanisms.

It was chosen to implement the "proxy" solution because it produces the shortest handover latency, but a slight modification on the CMD state machine can produce the first scenario described ("relay") which guarantees a more consistent request/ack scheme between the MAARS. By modifying also the MAAR's state machine it can be implemented the second solution ("locator").

An early MAD-PMIPv6 implementation was shown during a demo session at the IETF 83rd, in Paris in March 2012. An enhancement version of the prototype has been presented at the 87th IETF meeting in Berlin, July 2013. The updated demo included a use case scenario employing a CDN system for video delivery. More, MAD-PMIPv6 has been extensively used and evaluated within a testbed employing heterogeneous radio accesses within the framework of the MEDIEVAL EU project. MAD-PMIPv6 software is currently part of a DMM test-bed comprising 3 MAARs, one CMD, one MN and a CN. All the machines used in the demos were Linux UBUNTU 10.04 systems with kernel 2.6.32, but the prototype has been tested also under newer systems. This testbed was also used by the iJOIN EU project.

Appendix C. Applicability to the fog environment

Virtualization is invading all domains of the E2E 5G network, including the access, as a mean to achieve the necessary flexibility in support of the E2E slicing concept. The ETSI NFV framework is the cornerstone for making virtualization such a promising technology that can be matured in time for 5G. Typically, virtualization has been mostly envisaged in the core network, where sophisticated data centers and clouds provided the right substrate. And mostly, the framework focused on virtualizing network functions, so called VNFs (virtualized network functions), which were somewhat limited to functions that are delay tolerant, typically from the core and aggregation transport.

As the community has recently been developing the 5G applications and their technical requirements, it has become clear that certain applications would require very low latency which is extremely challenging and stressing for the network to deliver through a pure centralized architecture. The need to provide networking, computing, and storage capabilities closer to the users has therefore emerged, leading to what is known today as the concept of intelligent edge.

ETSI has been the first to address this need recently by developing the framework of multi-access edge computing (MEC).

Such an intelligent edge could not be envisaged without virtualization. Beyond applications, it raises a clear opportunity for networking functions to execute at the edge benefiting from inherent low latencies.

Whilst it is appreciated the particular challenge for the intelligent edge concept in dealing with mobile users, the edge virtualization substrate has been largely assumed to be fixed or stationary. Although little developed, the intelligent edge concept is being extended further to scenarios where for example the edge computing substrate is on the move, e.g., on-board a car or a train, or that it is distributed further down the edge, even integrating resources from different stakeholders, into what is known as the fog. The challenges and opportunities for such extensions of the intelligent edge remain an exciting area of future research.

Figure 7 shows a diagram representing the fog virtualization concept. The fog is composed by virtual resources on top of heterogeneous resources available at the edge and even further in the RAN and end-user devices. These resources are therefore owned by different stakeholders who collaboratively form a single hosting environment for the VNFs to run. As an example, virtual resources provided to the fog might be running on eNBs, APs, at micro data centers deployed in shopping malls, cars, trains, etc. The fog is connected to data centers deeper into the network architecture (at the edge or the core).

```
         +--------------------------------+  +--------------------+
         | -------   --------   --------  |  |      ----------    |
         | |     |   |      |   |      |  |  |    ---------- |    |
         | | @UE |   | @car |   | @eNB |  |  |  ---------- | |    |
         | -------   --------   --------  |  |  |  Data  | | |    |
         |                                |  |  | Center | | -    |
         | -------- Heterogeneous ------- |  |  |  (DC)  |-       |
    phy  | |      |   computing   |     | |  |  ----------        |
   infra | |@train|    devices    | @AP | |==|      ----------    |
         | --------    forming    ------- |  |    ---------- |    |
         |             the fog            |  |  ---------- | |    |
         | ---------        ------------  |  |  |  Data  | | |    |
         | |       |        |          |  |  |  | Center | | -    |
         | | @mall |        | @localDC |  |  |  |  (DC)  |-       |
         | ---------        ------------  |  |  ----------        |
         |              FOG               |  |       CLOUD        |
         +--------------------------------+  +--------------------+
         <--------- fog and edge ----------------->
                                     <--- edge & central cloud --->
```

Figure 7: Fog virtualization

In this context, where the "fog" hosts functions and resources, it is important to enable their mobility. In future versions of this document we will describe how to apply the solution described here to the fog environment.

Authors' Addresses

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: http://www.it.uc3m.es/cjbc/

Antonio de la Oliva
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Phone: +34 91624 8803
Email: aoliva@it.uc3m.es
URI: http://www.it.uc3m.es/aoliva/

Fabio Giust
Athonet S.r.l.

Email: fabio.giust.2011@ieee.org

Juan Carlos Zuniga
SIGFOX
425 rue Jean Rostand
Labege 31670
France

Email: j.c.zuniga@ieee.org
URI: http://www.sigfox.com/

Alain Mourad
InterDigital Europe

Email: Alain.Mourad@InterDigital.com
URI: http://www.InterDigital.com/