Network Working Group
Internet-Draft
Expires: December 28, 2006

J. Laganier
DoCoMo Euro-Labs
S. Narayanan
Panasonic
F. Templin
Boeing Phantom Works
June 26, 2006

Network-based Localized Mobility Management Interface between Mobile Node and Access Router
draft-ietf-netlmm-mn-ar-if-01

Status of this Memo

Copyright Notice

Abstract

This document specifies an IP layer interface between mobile nodes

(MN) and access routers (AR) of a network-based localized mobility management (NetLMM) domain. Such an interface is subject to a certain number of threats, amongst which are attacks on the mapping between the MN Identifier and IPv6 address set. A binding enforcement mechanism between those two is hence required to prevent malicious nodes to carry on various attacks like service theft or denial-of-service attacks. In the absence of link-layer specific mechanisms enforcing this binding, it is required to implement such mechanism at the IP layer MN-AR interface. Moreover, it is required that no NetLMM specific software support is present on MNs. The IP layer MN-AR interface described in this document fulfills these two requirements by using the SEND public key as the MN identifier, while being solely based on standard track IPv6 protocols (DNA and SEND.)

Table of Contents

1. Introduction

It is suggested in [I-D.ietf-netlmm-nohost-ps] that it would be desirable to have a localized mobility management protocol in which the host is not involved. The requirements for such a protocol have been analyzed in [I-D.ietf-netlmm-nohost-req]. Accordingly, a protocol for network-based localized mobility management (NetLMM) of IPv6 nodes will be specified by the NetLMM working group; until this occurs, this document assumes [I-D.wood-netlmm-emp-base] as a strawman NetLMM protocol in use in a NetLMM domain. Further revisions of this document will need to be adapted to the NetLMM protocol proposal chosen by the working group. Because the NetLMM protocol is network based, the mobile node (MN) is not required to implement new mechanism in its IP stack, nor to change its IP address when it attaches to a new access router (AR.)

Because the IPv6 MN will use a vanilla IPv6 stack, the interface between a MN and its AR has to be preserved. This means that standard IPv6 should work seamlessly with the network-based localized mobility support. More specifically, we require the proposed solution to be compatible with the mechanisms specified in:

- Neighbor Discovery for IP version 6 [I-D.ietf-ipv6-2461bis]

- IPv6 Stateless Address Autoconfiguration [I-D.ietf-ipv6-2462bis]

- Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315]

- Privacy Extensions for Stateless Address Autoconfiguration in IPv6 [I-D.ietf-ipv6-privacy-addrs-v2]

- Detecting Network Attachment in IPv6 - Best Current Practices for Hosts [I-D.ietf-dna-hosts]

- Detecting Network Attachment in IPv6 - Best Current Practices for Routers [I-D.ietf-dna-routers]

- Detecting Network Attachment with Unmodified Routers: A Prefix List based approach [I-D.ietf-dna-cpl]

- Detecting Network Attachment in IPv6 Networks [I-D.pentland-dna-protocol]

- SEcure Neighbor Discovery [RFC3971]

- Cryptographically Generated Addresses [RFC3972]

This document specifies an IP layer interface between MNs and ARs of

a NetLMM domain. Such an interface is subject to a certain number of threats, amongst which are attacks on the mapping between the MN Identifier and IPv6 address set. A binding enforcement mechanism between those two is hence required to prevent malicious nodes to carry on various attacks like service theft or denial-of-service attacks. In the absence of link-layer specific mechanisms enforcing this binding, it is required to implement such mechanism at the IP layer MN-AR interface. Moreover, it is required that no NetLMM specific software support is present on MNs. The IP layer MN-AR interface described in this document fulfills these two requirements by using the SEND public key as the MN identifier, while being solely based on standard track IPv6 protocols (DNA and SEND.)

1.1. Terminology

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 [RFC2119].

The following terms are defined within the scope of this document:

Mobile Node (MN)
: an IPv6 node moving in the NetLMM domain.

Access Router (AR)
: a default router that connects the MN to the NetLMM domain.

access interface
: a network interface of an AR attached to the link used by the MN.

Mobility Anchor Point (MAP)
: a router located in the NetLMM domain that handles packet exchanges with nodes in the domain.

Network-based Localized Mobility Management Domain (NetLMM domain)
: an administrative domain spanning links served by a set of MAPs (and their associated ARs and MNs) that provision addresses from the same IP subnet prefix(es).

Network-based Localized Mobility Management Protocol (NLMP)
: The NetLMM Protocol used in the backhaul of the NetLMM domain between ARs and MAP.

1.2. Abbreviations

The following abbreviations are used throughout this document:

NetLMM: Network-based Localized Mobility Management

ND: Neighbor Discovery.

NS: Neighbor Solicitation.

NA: Neighbor Advertizement.

RS: Router Solicitation.

RA: Router Advertisement.

NDP: Neighbor Discovery Protocol.

SLAAC: StateLess Address AutoConfiguration

DHCP: Dynamic Host Configuration Protocol

SEND: SEcure Neighbor Discovery.

DNA: Detecting Network Attachment.

CGA: Cryptographically Generated Address.

CGA_LL: The link-local unicast CGA generated by the MN with its public key (It is assumed that the MN is using a single public key to configure all of its link-local unicast and global unicast CGAs.)

CGA_1: One of the Global Unicast CGA generated by the MN with its public key.

CGA_2: Another one of the Global Unicast CGA generated by the MN with its public key (e.g. with a different subnet prefix.)

CGA_*: Any Unicast CGA generated by the MN with its public key (i.e. link-local or global.)

MNID: MN identifier set to the public key used by the MN for generating its CGAs.

1.3. Operating Environment

The MN-AR NetLMM interface is used between a MN and an AR of a NetLMM domain. In the absence of link-layer specific mechanism, it allows the AR and/or MN to detect network attachment, causing the AR to use NLMP to update routing at the MAP so that the MN stays reachable when it roams across the NetLMM domain.

```
         /-------------------------\
        /         Internet          \
        \                           /
         \-------+---------+-------/
                 |         |
         /-------+---------+-------\  ----
        /                           \    ^
       /         +-----+             \   |
       |         | MAP |-+           |   N
       |         +-----+ |-+         |   E
       |           +-----+ |         |   T
       |             +-----+         |   L
       |       Backhaul Network      |   M
       |    +-----+       +-----+    |   M
       |- - | AR1 | ..... | ARn | - -|
       |    +-----+       +-----+    |   d
       |      / \  Access   / \      |   o
       |     /   \ Network /   \     |   m
       |    /     \       /     \    |   a
       |    +----+                   |   i
       |    | MN | ------->          |   n
       \    +----+ AR change         /   |
        \                           /    v
         \-------------------------/  ----
```

Figure 1: Reference Network Diagram

The deployment scenario is shown in Figure 1 above: Several ARs are attached to an IP routing domain connected to the outside Internet via a MAP. The MNs, ARs, MAPs, and in-between routing fabric constitute the NetLMM domain. Each AR announces on its access interface a common set of prefix(es) which are routed to the MAP from the outside Internet. Packets arriving at the MAP and destined to a MN are tunneled to the appropriate AR.

In the absence of a link-layer specific MN-AR interface, it is required to have a common interface defined at the IP layer. Because no NetLMM specific software support is assumed to be present on MNs, this interface has to rely only on standard tracks IPv6 protocols such as ND, DHCP, SEND, and DNA. Interactions of these components with NetLMM are represented in Figure 2 below (note that hints received by DNA from other layers are omitted for clarity):

```
                  MN|AR
                Interface
                    |

                    |     +------------+      +----------+
                          |            |      |          |
                    |     | +--------+ | NLMP | +------+ |
                          | | NetLMM |<-------->|NetLMM| |
                    |     | +--------+ |      | +------+ |
                          |   ^     ^  |      |    ^     |
   +----------+     |     |   |     |  |      |    |     |
   |          |           |   v     |  |      |    |     |
   | +------+ |     |     | +-----+ |  |      |    |     |
   | |  DNA | |  NDP/DHCP | | DNA | |  |      |    |     |
   | | SEND |<------|------>|SEND | |  |      |    |     |
   | |  ND  | |           | | ND  | |  |      |    |     |
   | | DHCP | |     |     | |DHCP | |  |      |    |     |
   | +------+ |           | +-----+ |  |      |    |     |
   |    ^     |     |     |   ^     |  |      |    |     |
   |    |     |           |   |     |  |      |    |     |
   |    v     |     |     |   v     |  |      |    v     |
   | +------+ |           | +----+  |  |      | +------+ |
   | |      | |     |     | |    |<-+  |      | |      | |
   | |      | |   IPv6    | |    |     | IPv6 | |      | |
   | | IPv6 |<------|------>|IPv6|<------------>| IPv6 | |
   | +------+ |           | +----+     |      | +------+ |
   |          |     |     |            |      |          |
   |    MN    |           |    AR      |      |   MAP    |
   +----------+     |     +------------+      +----------+

                    |
```

Figure 2: NetLMM Component Interactions

2. Protocol Overview

The following subsections present the different situations in which an IP-based MN-AR interface is used to trigger the NetLMM protocol.

In the following figures it is assumed that the MN and AR clocks are synchronized enough to allow verification of ND messages via SEND timestamps. If that would not be the case, in order to verify freshness of ND signaling sent by the MN, the AR would be required to solicit the MN by sending to it an NS with a fresh nonce, to which the MN would reply with an NA containing the same fresh nonce.

2.1. MN powers on in a NetLMM domain

2.1.1. SLAAC Method

```
MN                        AR                      MAP
|                         |                        |
|    NS(DAD:CGA_LL)       |  UPDATE(MNID,CGA_LL)   |
|------------------------>|----------------------->| bind(CGA_LL,MNID)
|                         |REPLY[OK](MNID,CGA_LL)  | route(CGA_LL->AR)
|                         |<-----------------------|
|RS(CGA_LL->All_routers)  |                        |
|------------------------>|                        |
| RA(AR->{MN,All_nodes})  |                        |
|<------------------------|                        |
|                         |                        |
|    NS(DAD:CGA_1)        |   UPDATE(MNID,CGA_1)   |
|------------------------>|----------------------->| bind(CGA_1,MNID)
|                         | REPLY[OK](MNID,CGA_1)  | route(CGA_1->AR)
|                         |<-----------------------|
|                         |                        |
|    NS(DAD:CGA_2)        |   UPDATE(MNID,CGA_2)   |
|------------------------>|----------------------->| bind(CGA_2,MNID)
|                         | REPLY[OK](MNID,CGA_2)  | route(CGA_2->AR)
|                         |<-----------------------|
|                         |                        |
```

Figure 3: MN powers on and configures a Link-Local and two Global Unicast CGAs using SLAAC

As shown in Figure 3 above, when a MN using SLAAC powers on for the first time, it will generate a link local address based on its public key (CGA_LL) as per [RFC3972], and perform DAD on the address as per [RFC2462]. The NS(DAD) message generated will contain the public key in the CGA option as defined by SEND [RFC3971]. Upon reception of this NS message, the AR MUST generate an UPDATE to the MAP with the

public key as the MNID along with CGA_LL. The MAP MUST bind the CGA_LL to the MNID and establish a route binding for the CGA_LL to the AR. The MAP acknowledges the receipt of the UPDATE message.

While waiting for the completion of DAD, the MN may generate an RS message as per [RFC2461] with the unspecified address as the source address. Such a message will not contain a CGA option. The AR will respond with a multicast RA as per [RFC2461]. Alternatively, the MN will wait for completion of DAD and generate an RS message with its CGA-LL as the source address. With the prefix information received in the RA message, the MN can cryptographically generate one or more global addresses (CGA_*). For each of these addresses, the MN will perform DAD as the IID is likely to be different for each of these cryptographically generated addresses. For every NS(DAD) received from the MN, the AR will generate an UPDATE message to the MAP establishing binding in the MAP.

The use of multicast RAs may however not be acceptable in all NetLMM domains, e.g., when multiple MAPs and/or prefixes are used. In that case, the network has to somehow force the MN to source RSs from its CGA-LL, so that the AR can send to that CGA-LL a unicast RA containing the appropriate prefix information. This can be achieved by having the AR simply discard any RS sourced from the unspecified address, so that eventually the MN will complete DAD for its CGA-LL and start to use it as a source address while retransmitting RSs.

2.1.2. DHCP Method

```
MN                       AR                      MAP
|                        |                       |
|    NS(DAD:CGA_LL)      |  UPDATE(MNID,CGA_LL)  |
|----------------------->|---------------------->| bind(CGA_LL,MNID)
|                        |REPLY[OK](MNID,CGA_LL) | route(CGA_LL->AR)
|                        |<----------------------|
|RS(CGA_LL->All_routers) |                       |
|----------------------->|                       |
| RA(AR->{MN,All_nodes}) |                       |
|<-----------------------|                       |
|                        |                       |
|  DHCP SOLICIT(CGA_*)   |   UPDATE(MNID,CGA_*)  |
|----------------------->|---------------------->| bind(CGA_1,MNID)
|  DHCP REPLY(STATUS)    | REPLY[OK](MNID,CGA_*) | route(CGA_1->AR)
|<-----------------------|<----------------------| route(CGA_2->AR)
|                        |                       |
```

Figure 4: MN powers on and configures a Link-Local and two Global Unicast CGAs using DHCP with two-message exchange

As shown in Figure 4 above, when a MN using DHCP powers on for the first time it will cryptographically generate a CGA_LL and perform an RS/RA exchange as specified for the SLAAC method in Section 2.1.1.

The MN will then use its public key to generate a DHCP Unique Identifier (DUID) and Identity Association (IA) per ([RFC3315], Sections 9 and 10). If prefix information is included in the RA message, the MN can next cryptographically generate one or more global addresses (CGA_*). (The MN can additionally request delegation of prefixes per [RFC3633].) The MN will then issue a DHCP SOLICIT message including the DUID, IA and IA Address options that encode any CGA_*s as options. (Alternatively, the MN can omit IA Address options and allow the network to delegate non-CGA addresses.) If a two-message exchange is preferred, the MN will also include a Rapid Commit option in the DHCP SOLICIT per ([RFC3315], Section 17.1.2).

When the AR receives the DHCP SOLICIT (using two-message exchange) or DHCP REQUEST (using four-message exchange), it performs the same UPDATE/REPLY procedure as specified in Section 2.1.1, and returns a DHCP REPLY message with an appropriate status code to the MN.

The issues involved with the use of multicast RAs as described in Section 2.1.1 might be valid when DHCP is used for address configuration.

2.2. First attachment of MN moving into a new NetLMM domain

2.2.1. SLAAC Method

```
MN                     AR                       MAP
|                       |                        |
|        RS             |  UPDATE(MNID,CGA_LL)   |
|---------------------->|----------------------->| bind(CGA_LL,MNID)
|        RA             |REPLY[OK](MNID,CGA_LL)  | route(CGA_LL->AR)
|<----------------------|<-----------------------|
|   NS(DAD:CGA_LL)      |                        |
|---------------------->|                        |
|                       |                        |
|                       |                        |
|   NS(DAD:CGA_1)       |   UPDATE(MNID,CGA_1)   |
|---------------------->|----------------------->| bind(CGA_1,MNID)
|                       | REPLY[OK](MNID,CGA_1)  | route(CGA_1->AR)
|                       |<-----------------------|
|                       |                        |
|   NS(DAD:CGA_2)       |   UPDATE(MNID,CGA_2)   |
|---------------------->|----------------------->| bind(CGA_2,MNID)
|                       | REPLY[OK](MNID,CGA_2)  | route(CGA_2->AR)
|                       |<-----------------------|
|                       |                        |
```

Figure 5: MN moves into a NetLMM domain and configures a Link-Local and two Global Unicast CGAs using SLAAC

As shown in Figure 5 above, when a MN using SLAAC moves into a NetLMM domain for the first time, it will initiate link change detection as specified in [I-D.pentland-dna-protocol] by multicast transmission of an RS message. When the MN receives an RA message in response, it will figure out that it has changed to a link in a new NetLMM domain as defined by the DNA specification [I-D.pentland-dna-protocol]. Once the MN realizes it has changed to a new NetLMM domain, it will discard its current IP addresses and will execute DAD for its link-local address and new global addresses based on the prefix information in the received RA messages.

The global address configuration procedures of the MN, AR and MAP are the same as specified in Section 2.1.1.

2.2.2. DHCP Method

```
   MN                     AR                       MAP
   |                       |                        |
   |RS(CGA_LL->All_routers)|  UPDATE(MNID,CGA_LL)   |
   |---------------------->|----------------------->| bind(CGA_LL,MNID)
   |        RA             |REPLY[OK](MNID,CGA_LL)  | route(CGA_LL->AR2)
   |<----------------------|<-----------------------|
   |   NS(DAD:CGA_LL)      |                        |
   |---------------------->|                        |
   |                       |                        |
   |  DHCP SOLICIT(CGA_*)  |   UPDATE(MNID,CGA_*)   |
   |---------------------->|----------------------->| bind(CGA_1,MNID)
   |  DHCP REPLY(STATUS)   | REPLY[OK](MNID,CGA_1)  | route(CGA_1->AR)
   |<----------------------|<-----------------------| route(CGA_2->AR)
   |                       |                        |
```

Figure 6: MN moves into a NetLMM domain and configures a Link-Local and two Global Unicast CGAs using DHCP

As shown in Figure 6 above, when a MN using DHCP moves into a NetLMM domain for the first time, it will initiate link change detection as specified in [I-D.pentland-dna-protocol] by multicast transmission of an RS message. When the MN receives an RA message in response, it will figure out that it has changed to a link in a new NetLMM domain as defined by the DNA specification [I-D.pentland-dna-protocol] and/or by sending a DHCP CONFIRM message as specified in Section 2.3.2. Once the MN realizes it has changed to a new NetLMM domain, it will discard its current IP addresses and will execute DAD for its link-local address and configure new global addresses/ prefixes using DHCP.

The global address configuration procedures of the MN, AR and MAP are the same as specified in Section 2.1.2.

2.3. MN handovers in a NetLMM-domain

2.3.1. MN using SLAAC getting handover hint

```
MN                        AR                      MAP
|                         |                        |
|RS(CGA_LL->All_routers) |   UPDATE(MNID,CGA_*)  |
|----------------------->|----------------------->| route(CGA_LL->AR)
|                         |REPLY[OK](MNID,CGA_LL, | route(CGA_1->AR)
|    RA(AR->CGA_LL)      |          CGA_1,CGA_2) | route(CGA_2->AR)
|<-----------------------|<-----------------------|
|                         |                        |
```

Figure 7: MN using SLAAC getting handover hint and receives a unicast RA

As shown in Figure 7, when MN using SLAAC moves within the NetLMM domain, it will send an RS message with the source address as its link-local address as specified by [I-D.pentland-dna-protocol]. The AR again can use the public key in the CGA option to infer the MNID and send UPDATEs to the MAP. If the AR chooses to respond with a unicast RA, all required steps are done.

```
MN                        AR                      MAP
|                         |                        |
|RS(CGA_LL->All_routers) |   UPDATE(MNID,CGA_*)  |
|----------------------->|----------------------->| route(CGA_LL->AR)
|                         |REPLY[OK](MNID,CGA_LL, | route(CGA_1->AR)
|   RA(AR->All_nodes)    |          CGA_1,CGA_2) | route(CGA_2->AR)
|<-----------------------|<-----------------------|
|     NS(CGA_LL->AR)     |                        |
|----------------------->|                        |
|     NA(AR->CGA_LL)     |                        |
|<-----------------------|                        |
|                         |                        |
```

Figure 8: MN using SLAAC getting handover hint and receives a multicast RA

In a similar scenario, as shown in Figure 8, if the AR chooses to respond with a multicast RA, the MN will send an NS to learn about the AR and confirm reachability.

2.3.2. MN using DHCP getting handover hint

When a MN using the DHCP access method moves within the NetLMM domain, it receives the same handover hints as specified in Section 2.3.1.

```
MN                       AR                       MAP
|                         |                        |
|  DHCP CONFIRM(CGA_*)    |  UPDATE(MNID,CGA_*)    |
|------------------------>|----------------------->| route(CGA_LL->AR)
|  DHCP REPLY(STATUS)     | REPLY[OK](MNID,CGA_*) | route(CGA_1->AR)
|<------------------------|<-----------------------| route(CGA_2->AR)
|                         |                        |
```

Figure 9: DHCP CONFIRM message exchange

As shown in Figure 9, when the MN figures out that it has changed link, it sends a DHCP CONFIRM message containing its IA and all of the CGAs/prefixes it has previously registered per ([RFC3315], Section 18.1.2). The AR will generate an UPDATE message to the MAP and will send a DHCP REPLY message to the MN with appropriate status codes.

2.3.3. AR getting handover hint

```
MN                        AR                      MAP
|                         |                        |
|     NS(AR->CGA_*)       |                        |
|<------------------------|                        |
|     NA(CGA_*->AR)       |  UPDATE(MNID,CGA_*)    |
|------------------------>|----------------------->| route(CGA_LL->AR)
|                         |REPLY[OK](MNID,CGA_LL,  | route(CGA_1->AR)
|                         |           CGA_1,CGA_2) | route(CGA_2->AR)
|                         |<-----------------------|
|                         |                        |
```

Figure 10: AR getting handover hint of MN whose IP address is known

As shown in Figure 10, instead of the MN receiving the hint in scenarios where the AR receives the hint with the IP address of the handing over MN, the AR can send an NS to that IP address. The NA message received in response will contain the public key of the MN with which the AR can send an UPDATE message to the MAP.

```
   MN                       AR                     MAP
   |                        |                       |
   |   RA(AR->All_nodes)    |                       |
   |<-----------------------|                       |
   |     NS(CGA_*->AR)      |   UPDATE(MNID,CGA_*)  |
   |----------------------->|---------------------->| route(CGA_LL->AR)
   |                        |REPLY[OK](MNID,CGA_LL, | route(CGA_1->AR)
   |     NA(AR->CGA_*)      |          CGA_1,CGA_2) | route(CGA_2->AR)
   |<-----------------------|<----------------------|
   |                        |                       |
```

Figure 11: AR getting handover hint of MN whose IP address is unknown

As shown in Figure 11, if the AR does not receive the IP address information of the handing over MN along with the hint, the AR can schedule a multicast RA. The MN will try to fill its neighbor cache information with the AR and confirm its reachability by initiating an NS message to the AR. The AR can then send an UPDATE message to the MAP based on the public key in the NS message.

2.4. MN configuring additional CGAs/prefixes

If the MN chooses to configure new global addresses/prefixes at any point in time, it will contact the AR to configure the new addresses/prefixes as specified in Section 2.1.

2.5. MN configuring CGA that is in use by another MN in the NetLMM domain

2.5.1. MN using SLAAC configuring colliding CGA

```
   MN1       AR1                 MAP               AR2       MN2
    |         |                   |                 |         |
    | NS(DAD) |UPDATE(MNID,CGA,NS)|                 |         |
    |-------->|------------------>| collision(MNID) |         |
    |         |                   |                 |         |
    |         |                   | QUERY[DAD](NS)  | NS(DAD) |
    |         |                   |---------------->|-------->|
    |         |                   | REPLY[DAD](NA)  |  NA     |
    |         |                   |<----------------|<--------|
    |   NA    |REPLY[COLLIDE](NA) |
    |<--------|<------------------|
    |         |                   |
```

Figure 12: MN using SLAAC configuring a colliding CGA

As shown in Figure 12, AR1 learns about new global addresses configured by an MN MN1 from the NS(DAD) message sent by MN1. When AR1 sends an UPDATE to the MAP based on this NS(DAD), it also includes the entire NS in the message, and waits for a positive acknowledgment from the MAP. If the MAP has an entry for the same CGA with a different MNID, it will proxy this NS(DAD) up to the AR where the duplicate occurs (AR2). AR2 will then proxy the NS(DAD) by sending it to the solicited-node multicast address of the colliding MN MN2, and will receive back a signed NA from MN2. AR2 will then forward this signed NA to AR1 via the MAP. At that point, AR1 can securely defend the duplicate address on behalf of MN2 by sending to MN1 the signed NA.

2.5.2. MN using DHCP configuring colliding global CGA

```
MN                       AR                      MAP
|                        |                        |
|  DHCP SOLICIT(CGA_*)   |  UPDATE(MNID,CGA_*)    |
|----------------------->|---------------------->| collision(CGA_*)
|  DHCP REPLY(status)    | REPLY[COLLIDE](CGA_*) |
|<-----------------------|<----------------------|
|                        |                        |
```

Figure 13: MN using DHCP configuring a colliding global CGA

As shown in Figure 13, when a MN using DHCP configures one or more global CGAs, the MAP sends a REPLY to the AR with an indication for each global CGA that collided. The AR then sends a DHCP REPLY message to the MN with the appropriate status code for each colliding CGA.

2.6. MN unconfigures CGAs, powers off, crashes or leaves the domain

The AR SHOULD do periodic reachability testing with the MN using Neighbor Unreachability Detection (NUD) to learn about addresses being unconfigured or the MN being powered off or crashing. The trigger for this test could be neighbor cache entry timeout or a MLDv2 [RFC3810] unsubscribe for the solicited-node multicast address matching the MN's CGA.

```
   MN                       AR                      MAP
   |                        |                        |
   |     NS(AR->CGA_LL)     |                        |
   |<-----------------------|                        |
   |     NA(CGA_LL->AR)     |                        |
   |----------------------->|                        |
   |        NS(AR->CGA_1)   |                        |
   |   X <------------------|                        |
   |     NS(AR->CGA_2)      |                        |
   |<-----------------------|                        |
   |     NA(CGA_2->AR)      |                        |
   |----------------------->|                        |
   |     NS(AR->CGA_3)      |                        |
   |<-----------------------|                        |
   |     NA(CGA_3->AR)      |                        |
   |----------------------->|                        |
   |                        |UPDATE[DEL](MNID,CGA_1)|
   |                        |---------------------->| del_route(CGA_1)
   |                        |    REPLY[OK](MNID)    |
   |                        |<----------------------|
   |                        |                        |
```

Figure 14: MN unconfigures a CGA

As shown in Figure 14, the MN stops using the address CGA_1 and when the AR tries NUD for each of these addresses, it doesn't receive a response for CGA_1, resulting in an UPDATE message to the MAP to remove the mapping between MNID and CGA_1.

```
   MN                        AR                     MAP
   |                        |                        |
   |        NS(AR->CGA_LL)  |                        |
   |   X <------------------|                        |
   |        NS(AR->CGA_1)   |                        |
   |   X <------------------|                        |
   |        NS(AR->CGA_2)   |                        |
   |   X <------------------|                        |
   |        NS(AR->CGA_3)   |                        |
   |   X <------------------|                        |
   |                        |   UPDATE[DEL](MNID)   |
   |                        |---------------------->| del_route(CGA_LL)
   |                        |    REPLY[OK](MNID)    | del_route(CGA_1)
   |                        |<----------------------| del_route(CGA_2)
   |                        |                       | del_route(CGA_3)
   |                        |                       | del_bind(MNID)
```

Figure 15: MN crashes, powers off or leaves the domain

As shown in Figure 15, if the MN crashes, powers off or leaves the domain, the NUD will fail for all the associated addresses. In this case, the AR can remove the entry for the MN from the MAP by initiating an UPDATE message.

3. MN Specification

NetLMM place few specific requirements on an MN in a NetLMM domain. However, for the smooth operation of the NetLMM MN-AR interface, the MN MUST behave as specified in the following documents:

- Neighbor Discovery for IP version 6 [RFC2461] (MUST) and [I-D.ietf-ipv6-2461bis] (SHOULD)

- IPv6 Stateless Address Autoconfiguration [RFC2462] (MUST) and [I-D.ietf-ipv6-2462bis] (SHOULD)

- Privacy Extensions for Stateless Address Autoconfiguration in IPv6 [I-D.ietf-ipv6-privacy-addrs-v2]

- Detecting Network Attachment in IPv6 - Best Current Practices for Hosts [I-D.ietf-dna-hosts]

- Detecting Network Attachment in IPv6 - Best Current Practices for Routers [I-D.ietf-dna-routers]

- Detecting Network Attachment with Unmodified Routers: A Prefix List based approach [I-D.ietf-dna-cpl]

- Detecting Network Attachment in IPv6 Networks [I-D.pentland-dna-protocol]

- SEcure Neighbor Discovery [RFC3971]

- Cryptographically Generated Addresses [RFC3972]

Also, for MNs attached to networks that use DHCP, the MN MUST support the DHCP client message exchanges specified in:

- Dynamic Host Configuration Protocol for IPv6 [RFC3315]

The MN MUST use a single public key to generate all of its CGAs. This requirement is necessary to make it possible for the AR and MAP to bind together different addresses of the MN. That way, when a MN attaches to a new AR, the MAP will correctly update routing for all MN CGAs even if the MN is currently using only one of those (e.g. its link-local CGA) to send an RS.

With respect to the MUST support [RFC2461] and [RFC2462], and SHOULD support [I-D.ietf-ipv6-2461bis] and [I-D.ietf-ipv6-2462bis], the reason is that SEND avoids complication with the "DAD once per IID" optimization of [RFC2462]. This is because IIDs of CGAs with different subnet prefixes are different (subnet prefix is used as an

input parameter to the CGA generation algorithm.)

For NBMA links, links over which multicast is not well supported or for selection of specific neighbors, MNs and ARs can send packets addressed to the pre-defined multicast addresses specified in ([RFC4291], Section 2.7.1) to the Layer-2 unicast address(es) of one or more neighbors.

4. AR Specification

A NetLMM AR MUST behave as specified in the following documents:

- o Neighbor Discovery for IP version 6 [I-D.ietf-ipv6-2461bis]
- o IPv6 Stateless Address Autoconfiguration [I-D.ietf-ipv6-2462bis]
- o Privacy Extensions for Stateless Address Autoconfiguration in IPv6 [I-D.ietf-ipv6-privacy-addrs-v2]
- o Detecting Network Attachment in IPv6 - Best Current Practices for Hosts [I-D.ietf-dna-hosts]
- o Detecting Network Attachment in IPv6 - Best Current Practices for Routers [I-D.ietf-dna-routers]
- o Detecting Network Attachment with Unmodified Routers: A Prefix List based approach [I-D.ietf-dna-cpl]
- o Detecting Network Attachment in IPv6 Networks [I-D.pentland-dna-protocol]
- o SEcure Neighbor Discovery [RFC3971]
- o Cryptographically Generated Addresses [RFC3972]

Also, ARs MUST respond to DHCP client messages in a manner that is consistent with the DHCP relay/server messaging specified in:

- o Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315]

In addition, the AR MUST conform to the supplementary NetLMM specific requirements which follow in this section.

4.1. Promiscuous and all-multicast modes

The AR SHOULD put its access interface (the one exposed to MNs) in snooping/promiscuous mode so that it can receive most of the packets exchanged on the link it is serving. If a layer 2 switch is present between the AR and MNs, the port to which the AR is connected SHOULD be put in snooping/promiscuous mode. At the minimum, the AR MUST put its interface into a "receive all-multicast traffic" mode, and registers with MLDv2 [RFC3810] to all link-local solicited node multicast addresses to which a MN registers to with MLDv2. This insures that the AR can receive NSs so that it can proxy solicited NAs when the target MN is off-link.

4.2. Receiving ND Messages from MN

The NetLMM specific processing of received ND Messages depends on whether a packet is an NS part of the DAD procedure, or any other ND message. Section 4.2.1 defines the processing rules for NSs sent as part of the DAD procedure. Section 4.2.2 defines the processing rules for all others ND messages.

4.2.1. Receiving DAD NSs

If the AR receives a DAD NS which is secure according to [RFC3971], it MUST try to register the target address with the MAP. If the registration fails because this address is used by a different MN, the AR MUST defend the target address by sending a proxy NA as described in Section 4.3.2.

4.2.2. Receiving All Others ND Messages

If the AR receives any other ND message than those enumerated above, the message is secure according to [RFC3971], and the source address of the packet is not the unspecified address, it MUST try to register its source address with the MAP.

4.3. Sending ND Messages to MN

4.3.1. Sending NSs

An AR sends an NS to a MN in the following cases:

- o The AR receives from the MN a SEND-protected ND message which does not allow the AR to verify the MN CGA ownership. This can occur if the MN includes a Nonce parameter which does not correspond to the Nonce sent by the AR to the MN, or if the MN includes a Timestamp parameter which fails because the MN and AR clocks are desynchronized.

- o The AR receives from the MN an IP packet which is not a ND or DHCP Message before the MN registers the IP packet's source address.

- o The AR is performing the periodic reachability test of a MN it has precedently registered with the MAP. If the MN is unreachable, the AR MUST deregister this MN with the MAP.

In all the cases described above, the AR MUST verify MN CGA ownership by sending to the MN CGA an NS message including the MN CGA as a target address and a fresh Nonce.

4.3.2. Sending Proxy NAs

An AR SHOULD send a proxy NA to a MN performing DAD for an IP address which belongs to a MN which is known to be off-link by the AR in order to defend that address, as specified in Section 5.4. of [I-D.ietf-ipv6-2462bis].

To allow SEND MNs to accept proxy NS sent by the AR, the AR should follow the procedure described in Figure 12.

4.3.3. Sending RAs

All Prefix Information options included in RAs sent by an AR SHOULD have the "on-link" flag (L) set to 0 (zero.) This ensures that all packets sent by a MN are sent via the AR.

When the RAs contain no Prefix Information options, or when the MN wishes to procure additional prefixes, the MN can use DHCP prefix delegation mechanisms per [RFC3633].

4.3.4. Sending Redirects

An AR SHOULD NOT send a redirect message ([I-D.ietf-ipv6-2461bis], Section 8.2) unless it can determine that the sending node and better first-hop node reside on the same link and will remain on the same link.

4.4. Receiving All Other IPv6 Packets from MN

If the AR receives any other IPv6 packet than those enumerated above from a MN, and the source IP address is not registered yet with the AR, the AR MUST initiate a reachability test with the MN as specified in Section 4.3.1 to verify the MN CGA ownership.

4.4.1. Authenticated Packets

If the AR receives any other IPv6 packet than those enumerated above, and the MN origin of this packet is authenticated (by another security mechanism such as 802.11i or IPsec) and tied by any means to the public key used to generate the source CGA of that packet, then the AR MAY update the MAP based on reception of such packets.

4.4.2. Unauthenticated Packets

Unauthenticated IPv6 packets MUST NOT trigger any action in the NetLMM Domain.

4.4.3. Forwarding Packets

[RFC4291] states that:

> ARs MUST NOT forward any packets with Link-Local source or destination addresses to other links.
>
> Link-Local multicast scope spans the same topological region as the corresponding unicast scope.

This specification does not modify that behavior, i.e. an AR MUST NOT forward packets sent by a MN from or to a link-local address (unicast or multicast).

4.5. MN Identifier and IP addresses

All NLMP messages generated by an AR upon reception of triggers described in this document SHOULD use the SEND public key in the MNID field of NLMP messages. An alternative would be to use a truncated (say 128 bits) secure hash of the public key to reduce message size while keeping an equivalent security level. This public key MNID is hence securely bound to the set of IP addresses used by the MN, therefore preventing different redirection attacks.

In some deployments where MNs do not use ND and SEND (e.g. some cellular systems [RFC3316]), ARs and MAPs in the NetLMM domain SHOULD enforce the binding between an authenticated MN identity and the set of IP addresses used by the MN. In other words the network keeps track of IP addresses allocated to a specific MN identity. In the case of DHCP address allocation, DHCP requests and replies should be protected by a link-layer security context indexed by the authenticated MN identity.

5. Multilink Subnet Considerations

Multilink subnet issues are analyzed in [I-D.thaler-intarea-multilink-subnet-issues].

When each MN assigns addresses from separate IP prefixes, (e.g., per [I-D.thaler-autoconf-multisubnet-manets]) there are no multilink subnet issues.

When multiple MNs assign addresses from a shared IP prefix, multilink subnet issues can be avoided if ARs and MAPs act as neighbor discovery proxies as described in Figure 12, and ARs do not advertize subnet prefixes as "on-link" as described in Section 4.3.3.

6. IANA Considerations

There are no IANA considerations.

7. Acknowledgments

As usual in the IETF, this document is the result of a collaboration between many people. The authors would like to thanks (in alphabetical order) James Kempf, Alexandru Petrescu and Christian Vogt for discussion and/or comments that helped with first version of this document.

Ian Chakeres contributed the reference network diagram. Portions of this work were supported by the Boeing IRAD program and Boeing colleagues.

Julien Laganier is partly funded by Ambient Networks, a research project supported by the European Commission under its Sixth Framework Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks project or the European Commission.

8. References

8.1. Normative references

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[I-D.ietf-ipv6-2461bis] Narten, T., "Neighbor Discovery for IP version 6 (IPv6)", draft-ietf-ipv6-2461bis-07 (work in progress), May 2006.

[RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[I-D.ietf-ipv6-privacy-addrs-v2]
Narten, T., "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", draft-ietf-ipv6-privacy-addrs-v2-04 (work in progress), December 2005.

[I-D.ietf-dna-hosts]
Narayanan, S., "Detecting Network Attachment in IPv6 - Best Current Practices for hosts.", draft-ietf-dna-hosts-03 (work in progress), May 2006.

[I-D.ietf-dna-routers]
Narayanan, S., "Detecting Network Attachment in IPv6 - Best Current Practices for Routers", draft-ietf-dna-routers-02 (work in progress), March 2006.

[I-D.ietf-dna-cpl]
Nordmark, E. and J. Choi, "DNA with unmodified routers: Prefix list based approach", draft-ietf-dna-cpl-02 (work in progress), January 2006.

[I-D.pentland-dna-protocol]
Narayanan, S., "Detecting Network Attachment in IPv6 Networks (DNAv6)", draft-pentland-dna-protocol-01 (work in progress), July 2005.

[I-D.wood-netlmm-emp-base]
Wood, J. and K. Nishida, "Edge Mobility Protocol (EMP)", draft-wood-netlmm-emp-base-00 (work in progress), October 2005.

[I-D.ietf-ipv6-2462bis]
Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", draft-ietf-ipv6-2462bis-08 (work in progress), May 2005.

8.2. Informative references

[RFC3316] Arkko, J., Kuijpers, G., Soliman, H., Loughney, J., and J. Wiljakka, "Internet Protocol Version 6 (IPv6) for Some Second and Third Generation Cellular Hosts", RFC 3316, April 2003.

[I-D.ietf-netlmm-nohost-ps]
Kempf, J., "Problem Statement for Network-based Localized Mobility Management", draft-ietf-netlmm-nohost-ps-04 (work in progress), June 2006.

[I-D.ietf-netlmm-nohost-req]
Kempf, J., "Goals for Network-based Localized Mobility Management (NETLMM)", draft-ietf-netlmm-nohost-req-01 (work in progress), April 2006.

[I-D.ietf-netlmm-threats]
Kempf, J. and C. Vogt, "Security Threats to Network-based Localized Mobility Management", draft-ietf-netlmm-threats-00 (work in progress), April 2006.

[I-D.thaler-intarea-multilink-subnet-issues]
Thaler, D., "Issues With Protocols Proposing Multilink Subnets", draft-thaler-intarea-multilink-subnet-issues-00 (work in progress), March 2006.

[I-D.thaler-autoconf-multisubnet-manets]
Thaler, D., "Multi-Subnet MANETs", draft-thaler-autoconf-multisubnet-manets-00 (work in progress), February 2006.

Appendix A. Version history

A.1. -00 to -01

- added DHCP access method including DHCP prefix delegation.
- added new network reference diagram.
- added definitions for NetLMM domain and NLMP.
- updated NA proxying method for colliding CGAs.
- added text on sending IP multicast messages to a Layer-2 unicast address.
- added new Section 4.5 text on MNID/IP address binding.
- added new Section 5. on multilink subnet issues.
- various editorial changes."

Authors' Addresses

Julien Laganier
DoCoMo Communications Laboratories Europe GmbH
Landsberger Strasse 312
Munich 80687
Germany

Phone: +49 89 56824 231
Email: julien.ietf@laposte.net
URI: http://www.docomolab-euro.com/

Sathya Narayanan
Panasonic Digital Networking Lab
Two Research Way, 3rd Floor
Princeton, NJ 08536
USA

Phone: +1 609 734 7599
Email: sathya@research.panasonic.com

Fred L. Templin
Boeing Phantom Works
P.O. Box 3707 MC 7L-49
Seattle, WA 98124
USA

Email: fred.l.templin@boeing.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.