Mobile IP Working Group
INTERNET DRAFT
10 July 2001
Expires: 10 January 2002

Theo Pagtzis
Peter Kirstein
University College London

A Framework for Proactive Mobility in Mobile IPv6
draft-pagtzis-mobileip-proactivev6-00.txt

Status of This Memo

This document is a submission by the Mobile-IP Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the MOBILE-IP@STANDARDS.NORTELNETWORKS.COM mailing list.

Distribution of this memo is unlimited.

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

```
The list of current Internet-Drafts can be accessed at:
     http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be
 accessed at:
     http://www.ietf.org/shadow.html.
```

Abstract

This document specifies a scheme to provide support for *proactive mobility* for IPv6. It encompasses a novel approach for seamless handoff and proactive roaming context transfer. The latter establishes a generic method for proactive context transfer and manipulation of different classes of context relating to the state of IP connectivity for a mobile node.

The above scheme is in support of the argument that waiting to start the handoff process until the node has reached the cell boundaries, where the range between a set of neighbouring coverage areas is expected to overlap, is bound to be too late for high speed mobile nodes with irregular mobility patterns. For this reason the proposed mobility scheme enables proactively roaming context *allocation and management* with forward IP-guarding against irregularities of

the mobility pattern of the mobile node, before the latter reaches the boundaries of overlap areas between adjacent coverage areas.

To address such form of proactivity, the proposed model identifies the notion of a **mobility neighbourhood** through a set of *adjacent* coverage areas, that are directly visible during the mobility pattern of a mobile node and probable to be visited by it in the immediate future; since each coverage area is effected by a single access point controlled directly by some routing entity, the proposed mobility model establishes the abstraction of a routing neighbourhood over spanning tree routing topologies, acting as the traffic forwarding medium and effected over some mobility neighbourhood.

Since adjacency over the mobility neighbourhood does not imply adjacency over the underlying routing neighbourhood, the proposed model further establishes an accurate mapping between the mobility neighbourhood that surrounds the current point of attachement of an MN and the correct underlying routing neighbourhood so as to effect proactive mobility through forward context tranfer and manipulation of roaming state or other context class.

The proposed model caters for safeguards over sustained IP connectivity guarantees irrespective of the velocity vector of the mobile node. This is achieved by introducing a 1-level lookahead IP roaming state over the established mapping between existing mobility neighbourhood and the respective routing neighbourhood through the notion of *soft* Care-of Addressing (CoA); in this mapping the latter represents a future instantiation and transition of a mobile node to adjacent coverage areas controlled by different routing entities and subsequently different routable networks.

The proactive allocation of Soft Care-of addresses over the identified routing neighbourhood is further mapped over a **Proactive CoA** onto the Multicast IPv6 domain so as to abstract the plurality of unicast Care-of Addressing onto a single multicast routing identifier. Management of the aformentioned mappings effect seamlessness in the handoff process for a Mobile node, while transfer and manipulation of its roaming context over the identified routng neighbourhood is effected proactively.

The proposed scheme further caters for robustness against **cell-bouncing** effects that can render harmful disruptions at the IP layer in terms of MN relocation latencies and accompanied packet loss.

Contents

1 Introduction

The signalling delays expected currently over the Mobile IPv6 protocol standard in IETF [21], may cause the mobile node to experience significant handoff latencies. The magnitude of such delays is sufficient to disrupt even momentarily the network connectivity of a mobile node or at best degrade significantly the servicing quality of the wireless network connection.

The problem is further exacerbated when the velocity of the mobile node fluctuates beyond the boundaries of the core Mobile IPv6 signaling rate with which the presence of a mobile node is propagated into a new routing IPv6 domain. This implies that reliance in signalling like standard router advertisments may prove insufficient in ensuring sustained performance for time critical applications serving at the mobile node. This may further affect provisioning guarantees in quality of network services as experienced by the mobile node.

Currently Mobile IPv6 does not mandate any scheme that may provide an upper bound over handoff latency and subsequently qualitative guarantees that can assure a minimal measure of sustained quality in wireless network connectivity.

In the scheme proposed in this document we establish a model that promotes *proactive* mobility over IPv6 networks. We argue that mobility should not rely on the reactivity of the upcoming visited network. This is far too slow as soon as the node begins to consider higher and less predictable mobility patterns. On the contrary, we argue that it is the network either current or previous that should work proactively towards managing a mobile node's connectivity state, put simply as *mobility-awareness over mobile node-specific roaming state.*

2 Related work

There have been a number of proposed protocols for Mobility in IPv6 that strive towards minimizing the handoff latency during transition between Access Routers (ARs), especially in the light of IP flows with real-time constraints, between an MN and its peers. Generic design principles for fast handoffs have been proposed by [17]. Such principles are manifested in design specifications from [23] as well as [24] which is targeted specifically for context transfer in the light of seamless mobility.

[27] proposes the comeback of some mobility agent in an IPv6 network that is foreign to the visiting MN; this is very similar to the traditional IPv4 FA mobility entity. It attempts to support fast handoffs by claiming faster reaction by the LMA in terms of IPv6 address allocation and authentication, while enabling some transitional path between Mobile IPv4 and IPv6.

[1] proposes a different approach where an MN informs with bindings its peers (HA+CNs) about its current link as well as possible 'to-be-visited' links. This is achieved by introducing the neighbourhood discovery task at the mobile node (MN). The scheme, further employs route segments in the routing header for packet forwarding towards the MN.

The scheme presents robustness problems in what is described as ping-pong effect[1], since each route segment is effected only once as the packet gets forwarded serially to all CoA in some potential probabilistic ordering. For instance if the MN moves from AR1 to AR2 and back on AR1 then the route segment could only reach from CoA1 to CoA2 but not back to CoA1[2].

Related considerations can also be found in [22]. The scheme elaborates on Foreign Agent (FA)-assisted approach where the old FA communicates with the new FA in order to setup a handoff request between the two foreign agents. The handoff request is initiated either from the old or new FA when the link layer of some network entity detects that it departs the coverage area. The scheme assumes that both FAs fall under the same Gateway foreign agent (GFA) in a regional registration mobility aware IPv4 environment.

Alternative approaches in Mobile IP have also been proposed in [32], [19], [30]. These schemes are employing IP-Multicast between the MN and its peers (end-to-end) for the addressing and routing of packets to mobile nodes. That is, the schemes require that traffic is forwarded to a multicast group on an end-to-end basis for the purposes of minimal latency handoffs during the mobility pattern of the mobile node. To this end a small group multicast solution has been proposed [25] and [13] where each traffic source. Both schemes utilize principles from [4] and they employ a multiple unicast destination option at the routing header of a packet. This notion is similar to the route segments of [1] since both schemes rely on unicast routing to deliver the packet. The schemes rely on the provision of all CoA destinations to the peer entities from the MN.

Our proposed scheme as described in following sections is architecturaly different from all the above schemes; this is both in terms of routing during handoff transitions as well as neighborhood establishment for the purposes of minimal handoff delays as well as signalling overheads over the air interface as well as elimination of ping-pong effects. With these design requirements in mind, we argue that the visited network should be responsible for providing proactively mobility services to the MN *ahead* of its future move. This is the very essence of the proactive model proposed in this document.

3 Proposal for Proactive Seamless Handoffs

3.1 Terminology and Assumptions

Before proceeding to describe the proposed model we provide with some basic terminology that our model adopts. We further provide with fundamental assumptions that our model makes about existing information/messaging that is made available through other protocols.

In IPv6, the notion of a Access Router (AR) is overloaded with more functions than just plain routing [29, 5] and its supporting protocol elements like

[1] oscillation in handoffs between two or more Access Routers

[2] as soon as the last CoA is reached in the route segments the packet is sent back to the HA where it is being discarded

Neighbour Discovery, Stateless address autoconfiguration or other. In a ubiquitous mobility environment it is expected to support other funtions over IP such as context transfer, location management, over functions like paging, admission control closely coupled with AAA functions and so forth. It is for this reason ARs as routing entities will be collectively termed as **Routing Domain Controllers (RDCs)** for the purpose of this document.

For simplicity, the model initially considers that one RDC is controlling a single **Access Point (AP)** that maps to a single coverage area[3]. It is however, possible to expand this model towards control of multiple APs from a single RDC. This, nevertheless, may be safely assumed as a uniform coverage area of simply larger diameter. We define an Access Point as a link layer entity that operates, currently, transparently from the perspective of IP layer.

For this model, the above further implies that semantically the notion of a *Routing Domain* maps currently to the notion of a single IP subnet. That is to say that *an RDC is controlling a single subnetwork in terms of routing.* As such, for the purposes of clarity and simplicity we define a **routing domain** to be a single routable (sub)network that is identified by a unique routing prefix. Thus, the two terms are used interchangeably in the remainder of this document.

From that perspective the model proposed does not distinguish at this stage between intra-domain and inter-domain movement for a mobile node as the basic transition constituent is a routing domain which according to the above definition is a single sub-network[4]. As such the model is concerned only with generic cross network transitions for a mobile node as the fundamental requirement.

Furthermore, a **mobile node (MN)** as a mobile entity that is primarily wireless while expected to rove between different network links it is primarily expected to first either power on and register with its *home* network domain and subsequent RDC in order to obtain its home IPv6 address. Alternatively the MN may be configured manually with the required IP connectivity information that render it service-able within its home network, even if its not expected to power on *at all* within its home network link.

3.1.1 Defining Router and MN Handoff states

To provide a clear designation on the states that a router may transit as it serves a mobile node, we define a set of possible states for both an RDC and the accommodated MN. These states identify the mode of either the MN or the RDC with respect to transition triggers from the former.

In particular, a **mobile node (MN)** may transit between four states, namely: `INCOMING`, `ACTIVE`, `OUTGOING`, `SUSPENDED` and `HANDOFF`. These are illustrated in figure 1. While a network transition is pending, the MN enters the INCOMING state for the purposes of link layer connectivity, admission control, and possible context transfer. Upon completion of the handoff the MN enters

[3] Such coverage area may be modelled as hexagonal for the purposes of adjacency and continuous coverage

[4] single routable link

the ACTIVE state; that also signifies the completion of either admission control or context transfer[5]. Unsuccessfull completion of these functions imply that the MN enters a SUSPENDED state during which seamless handoff guarantees may be revoked. This can also occur in the event suspending the MN while active[6]. The MN may further transit to the OUTGOING state if it shows signs of transition candidacy at the RDC. The MN may also reach this state in the event that the RDC forces the MN to become a handoff candidate for reasons of performance if sufficient overlap between coverage areas allow to do so. The MN enters the HANDOFF state when the RDC has commenced the process of transition over to a new RDC and until the transition has stabilised over the new RDC such that it can proceed to complete.

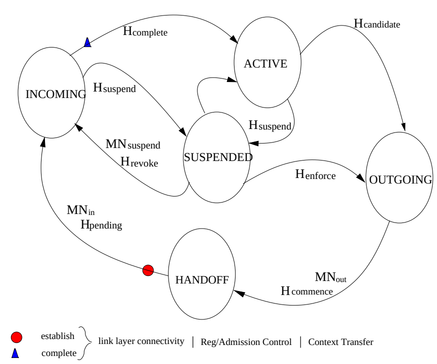

Figure 1: Mobile Node handoff transition states

With respect to the **RDC** entity, consider two routing domains RD_i and RD_{i+k} where each one is being serviced by a different RDC. The visited routing domain and subsequently the visited RDC is assumed as an entity that engages into a three-state transition, described by the following core states: `NEW`, `CURRENT` and `PREVIOUS`.

While MN transits from RDC_i towards RDC_{i+k}, the latter enters into state NEW while RDC_i moves from CURRENT to PREVIOUS state. As MN gets registered and admitted to the new routing domain RDC_{i+k} transits to CURRENT state.

The RDC_{i+k} currently in the NEW state will need to further establish and complete the aformentioned functions for the MN. These two functions are subject to completion, suspension, or even revocation depending on characteristics such as subscription, security compliance, potential flow behaviour. As such, the NEW router is bound to linger between the states of NEW and CURRENT for incoming/associated node. This is further illustrated by figure 2.

[5] it may be that admission control is a context transfer function

[6] due to potential billing issues

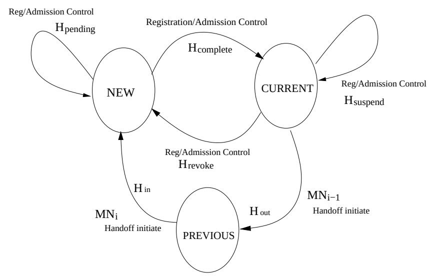

Figure 2: Router transition states during handoff

The mobility pattern of the MN may affect the stability of the state transition process of the RDCs. The most representative of these is what we call *coverage area (CA) bounce*. This is also known as *ping-pong* effect [17]. We envisage that our scheme should cater for the stability of the transition process even under harsh CA bounce conditions.

3.1.2 Defining Proactive Mobility

It is essential to define the semantics conveyed by the term proactive mobility. We are aware that in the particular class of mobility scenarios considered[7], a set of RDCs define a *routing neighbourhood* which spans over some geographic area through their Access Point (AP) instantiations. This is shown if Figure 3. Such instantiations and their underlying RDC define further a *mobility neighbourhood* which is manifested as overlapping coverage areas shown in Figure 4.

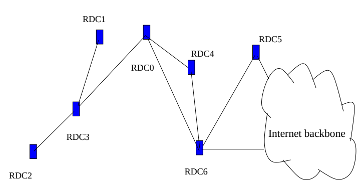

Figure 3: Routing neighbourhood accomodating the MN

While mobile, the MN transits between diferent coverage areas within a mobility neighbourhood. As previously stated, each such coverage area effected through a single AP, maps to a single RDC and thus the mapping between coverage area and RDC is assumed to be one-to-one.

It is assumed that the MN cannot be aware where it may move next both geographically or over the respective network topology. We regard essential that

[7] we do not attack the case where the RDC is mobile also, i.e. ad-hoc nets

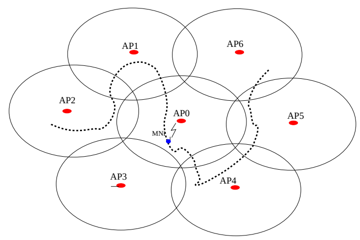

Figure 4: Mobility neighbourhood servicing the MN

the routing neighbourhood of the accomodating networks, acts on its own for the purposes of preparing connectivity state or any peripheral network functions/protocols before the MN actually engages in requesting them. The rationale behind such assumption is simple: the MN's mobility pattern is assumed to be non-deterministic in terms of both speed and direction. Thus, at the time that the MN will be in need of the access services, it's movement will force it off the coverage area of the current visiting network. This is bound to introduce an error prone air interface that can affect significantly in terms of latencies or potential packet loss, the acquisition process of IP connectivity state and thus continuous network connectivity towards a ubiquitous mobility network environment.

The proposed model, thus, considers proactive mobility as the function/set of interactions that preconfigures all or most network functions that are required by a neighbouring network in providing network services with some sustained connectivity safeguards. The ultimate goal in this proactive mobility approach is to effect **seamless** network connectivity. It is defined as the perceived quality of network services remains constant or may degrade within some *perceivably* acceptable boundaries. Given that perceivable quality boundaries are reached, the term may further transcend its qualitative guarantees to objective qualitative metrics that are allowed to degrade within some *objectively* acceptable boundaries.

We examine *proactivity* from the perspective of the fixed network, extended by the last hop (RDC) wireless coverage area (APs) with respect to the movement of an MN. The concept manifests itself in our model by having each routing neighbourhood that is considered mobility-enabled, to interact and communicate proactively **roaming state** about its mobility neighbourhood.

The set of interactions involve exchange of routing information of neighbouring IPv6 routing subnetworks that map accurately over the mobility neighbourhood servicing the MN. This should provide the an architectural substrate for the purposes of enabling *proactive* mobility over IPv6.

Before proceeding with the description of the proactive model, we distinguish between proactive transition establishment and other types of transitions by defining three generic different types of mobility handoffs with respect to a routing domain that may accommodate a mobile node in the immediate future;

that identifies the **candidate new routing domain** effected over a coverage area of the mobility neighbourhood. The three transition types are defined as:

- *reactive handoffs* : handoffs that are effected as a result of a *reaction* of the new routing domain that is detecting an incoming node[8] soliciting a registration request for the purposes of being accomodated on that network.
- *forward reactive handoffs* : handoffs that are effected as a result of a forward hint from some network entity, primarily the current RDC, to the new RDC. Here the current RDC is effectively **pushing** some context state, such that the new RDC is somewhat prepared with respect to the upcoming visiting of the MN. This identifies an informed reaction from the part of the new RDC. Alternatively, and potentially in special cases, hint requests may be initiated from the new RDC (towards the current RDC) such that some context state is **pulled** by the new RDC from the current one.
- *proactive handoffs*: handoffs are administered entirely on the initiative taken from the *candidate* new routing domain to provide sufficient IP routing information which is to be utilized by the current routing domain in establishing IP state. We emphasize on the word candidate since we do not rely specifically on knowledge of the identity of a new RDC and subsequently new routing domain. Here RDCs are conspiring to provide the essential information to each other in view of future handoffs. Such information is self-contained and sufficient to effect a handoff without having the accomodating RDC to react to registration requests from an MN, since they are not needed.

Currently most Mobile IPv6 protocol recommendations consider primarily reactive handoff or at best forward reactive schemes in terms of mobility handoffs. To the best of our knowledge only a few draft recommendations in IETF only begin to address some notion of proactivity. In addition, the IETF Working Group for Seamless Mobility (SeaMoby) has only recently started to consider the notion of seamless mobility through forward context transfer; this is however still at its infancy, since currently the SeaMoby WG is in the process of specifying the semantics and requirements for Seamless mobility as well as the scope of its charter.

It is important to note that the proposed proactive mobility allows complete as well as partial proactive mobility management. This stems from the fact that the proactive mobility model may not be implemented on *every* RDC. In such event, only a *partial* 1-level lookahead view of IP roaming context can be provided. Since proactive mobility may not be effected on every RDC, this lookahead view of roaming context **fades** on the next visited routing domain that does not support it.

[8] either through Router solicitation or registration

4 The Proposed Proactive Mobility model

As bandwidth resources over any wireless interface are commonly accepted to be scarce[9] in relation to their wired counterparts, the mobility model considers that the MN should effect minimal signalling in terms of conveying information to and from the visiting network link, over the interface. This is independent of whether the accomodating routing domain is the current or a future one. For this reason that the model attempts to minimize any signalling dependency on the mobile node. This is especially important under harsh propagation effects that may yield an air interface with increased error rates during the movement pattern of the MN. In such situations, dependency from the MN over mobility signalling may present adverse effects in the connectivity of the MN.

Towards such argument the proactive model extends beyond the current model met in most recommendations that relies generaly on what we term as *previous-new router relationship*. In particular, it encompasses not only the current and new routing domain through the respective RDCs, but the entire set of RDCs[10] that effect a **single contiguous mobility neighbourhood**.

This is shown in Figure 5 where the coverage area effected by the routing domain of RDC_0 is surrounded[11] by a six-cell[12] mobility neighbourhood. Each of the six neighbouring coverage areas is controlled by a single RDC. The list of controlling RDCs may be denoted by the list:

$$\begin{aligned} RNei_{RDC_0} = \{ &< LLA_{rdc1}, IP_{rdc1} >, < LLA_{rdc2}, IP_{rdc2} >, \\ &< LLA_{rdc3}, IP_{rdc3} >, < LLA_{rdc4}, IP_{rdc4} >, \\ &< LLA_{rdc5}, IP_{rdc5} >, ..., < LLA_{rdcN}, IP_{rdcN} > \} \end{aligned} \tag{1}$$

$$MNei_{AP_0} = \{AP_1, AP_2, AP_3, AP_4, ..., AP_n\} \tag{2}$$

$RNei_{RDC_0}$ denotes the **routing neighbourhood vector (RNV)**. Each component of RNV consists of the Link Layer Address (LLA) and the global IPv6 address of the RDC, while N is the number of RDCs mapping to the *mobility neighbourhood vector (MNV)* denoted as $MNei_{RDC_0}$. $RNei_{RDC_0}$ signifies also the complete set of *proactive mobility state* required at each RDC before it can provide full proactive mobility.

4.1 Proactive discovery of the Mobility Neighbourhood

It is important to consider the mapping between the topology of some routing neighbourhood and the available mobility neighbourhood as effected over the

[9] ranging for aggregate bandwidth in the orders of Mbps, while their wired counterparts effect bandwidth in the order of GBps

[10] that may or may not be immediate routing neighbours

[11] assuming initially continuous coverage

[12] the neighborhood may grow larger depending on the range of the wireless cells

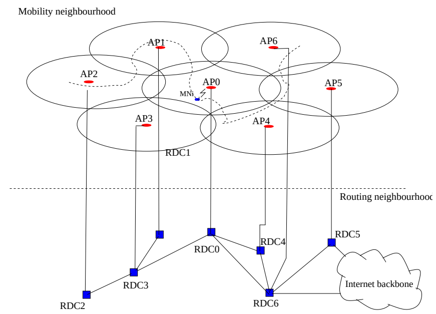

Figure 5: Neighbouring Routing Domains defining a routing neighbourhood

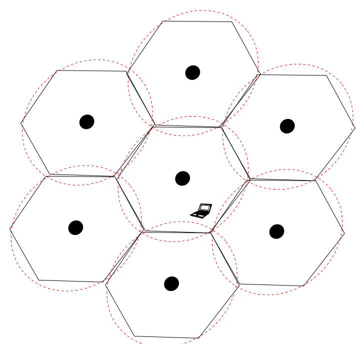

Figure 6: Basic shape of the mobility neighbourhood

former for use by the MN; the significance lies on the perspective of signalling when the mobility neighbourhood (MNV) needs to be configured for some RNV for the proposed proactive model.

For instance, the mobility neighbourhood may be safely assumed to follow the pattern of Figure 6 while the number of neighbours may change depending on the diameter of the coverage area for each cell. In such mobility neighbourhood, *each neighbouring coverage area is directly reachable from the cell that the MN is currently residing.* This is not necessarily true for the underlying routing neighbourhood and the corresponding RNV since it depends highly on the topology.

It can be seen from Figure 7 that a *star* topology maps naturaly over a mobility neighbourhood. However, it may not be realistic to assume only star topologies per MNV. This is because an a-priori mobility neighbourhood would require essentially the existence of a fully connected mesh topology as shown in Figure 7; this is not always the case for routing topologies currently effected in the Internet.

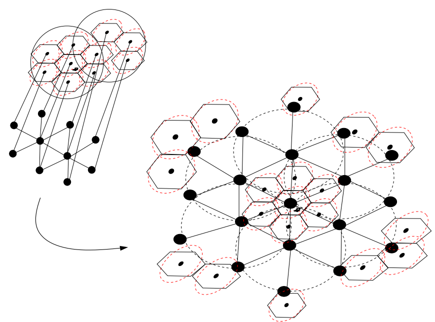

Figure 7: Mobile neighbourhood over fully-connected mesh routing topology

It would, thus, be more realistic to consider and assume *spanning-tree* topologies over the mobility neighbourhood of the form shown in Figure 8.

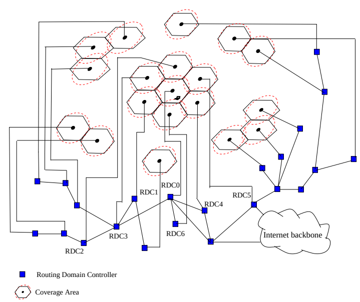

Figure 8: Mobility neighbourhood over spanning tree routing topologies

As it may be seen from the topology of Figure 8 it may not be the case that coverage areas which are neighbouring in a mobility neighbourhood map to an RNV where each RDC is directly connected with each other. That implies that a movement of the MN from one coverage area to another **does not** imply a direct link between the respective RDCs.

In this case, discovery of the mapping between the available mobility neighbourhood and the corresponding routing neighbourhood has to be effected through two different means:

- *static configuration*: The mobility as well as the routing neighbourhood for fixed infrastructure networks (i.e. not ad-hoc) is not expected to change often. As such, it is possible that the MNV and the underlying RNV can be manually established and configured for each RDC.
- *dynamic learning and configuration*: It is possible that static configuration may be avoided by employing dynamic learning on each RDC from

one-time information that are conveyed to each RDC by the MNs that are transiting over the respective coverage area, and are thus natural discoverers of the mobility neighbourhood.

The first case of static configuration is fairly trivial since it each RDC can be configured manually.

In the case of dynamic configuration, each RDC is effectively required to discover its mobility neighbourhood in order to be able to effect proactive mobility. For this it would be essential for the RDC to enter some **probe mode** over the mobility neighbourhood in order to *learn* what is the effective RNV for the MNV of its coverage area (AP). To effect such dynamic learning there are two approaches that may be considered each with different strengths and weaknesses:

- incremental RNV aquisition with reactive dynamic learning
- complete RNV aquisition with proactive learning

The following sections describe each of the two schemes individually.

4.1.1 Incremental RNV aquisition through temporal reactive learning

Through this approach each MN **conditionally** contributes, indirectly, routing information through *new* router advertisments it hears as it transits through some overlap area from the CURRENT RDC to a NEW RDC. The condition pertains to the temporal fact that the MN either has been provided *some partial* or *null* proactive mobility state for the particular NEW RDC well before it hands off from the CURRENT RDC.

Initially each RDC has no proactive mobility state in terms of an RNV mapping to the mobility neighbourhood of its coverage area. This implies that the CURRENT RDC cannot *push*[13] any proactive mobility state to the MN; as such, the MN has *no* proactive mobility state for the particular NEW RDC and subsequenetly the coverage area it is transiting to. This condition *is sufficient and necessary* for the MN and the NEW RDC to engage into the reactive dynamic learning process, during which:

- when no proactive mobility state exists for the NEW RDC, seamless mobility cannot be effected at the latter. The model employs base IPv6 mobility, while the RDCs engage in **probe mode** for the purpose of dynamic but reactive RNV learning.
- Since the MN has had no proactive mobility state for that RDC, as soon as it transits to the NEW RDC, and it obtains IP connectivity, it provides the latter with a unicast **Indirect RNV Update (I-RNV Update)**

[13] the model of pushing proactive mobility state is explained in later sections

message. This message updates the specific RNV for the mobility neighbourhood of its coverage area with the IPv6 as well as the link layer address for the link of the PREVIOUS RDC over which part of the mobility neighbourhood of the CURRENT RDC is effected. The latter may be derived from the IPv6 address of the RDC by using the prefix length and reversing the rules for EUI-64 interface identifiers. Alternatively the MN should set the L flag, which signifies the explicit inclusion of the previous RDC's LLA.

- Receipt of the I-RNV Update at the NEW RDC from the MN, signifies to the former that the PREVIOUS RDC is currently not aware that their coverage areas belong to the same mobility neighbourhood. For that reason the receipt of the I-RNV Update *triggers* further a unicast **Direct RNV Update (D-RNV Update)** from the NEW RDC destined to the PREVIOUS RDC. In this manner both PREVIOUS and NEW RDC are establishing proactive mobility state for either direction. This is a lightweight message that maintains a different code than the I-RNV Update and needs no explicit specification of the mobility-enabled link's IPv6 address for the NEW RDC; this address is retrieved from the sender IPv6 address of the D-RNV Update.

The next MN that transits from the same PREVIOUS RDC to the same NEW one, or vice versa, can now utilise the specific part of the proactive mobility state at the PREVIOUS RDC as described in following sections. This signifies that the both PREVIOUS and NEW RDCs can now effect partial proactive mobility state with respect to the particular neighbouring coverage area of their mobility neighbourhood, while they discovers their remainder. This is shown in Figure 9.

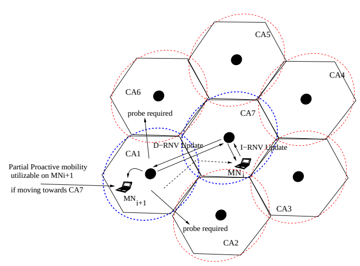

Figure 9: Partial proactive mobility through temporal reactive learning

As it may be seen the MN needs to 'push' I-RNV Updates and subsequently trigger D-RNV ones, only when its has **no** proactive mobility state with respect to the NEW RDC it is transiting to. Otherwise, such messaging is suppressed and proactive mobility state can take effect.

In the worst of scenarios where proactive mobility state in a NEW RDC becomes stale[14], the state is instantly nullified; this is because prefix information available from router advertisment heard by the MN at the NEW RDC cannot match with any of the proactive mobility state supplied at the CURRENT RDC. As such, proactive IPv6 mobility extensions are naturally suspending for the NEW RDC and its coverage area, while a new probe mode is triggered at the transiting MN initiating an new I-RNV Update.

4.1.2 Complete RNV aquisition with fully proactive learning

In the case of fully proactive learning approach, each RDC is building routing information about its own mobility neighbourhood independent of the movement of any MN. In particular, each RDC maintains a preconfigured *coverage area tuple (CAT)*, denoted as:

$$\begin{aligned} CAT_{AP_i} = \{l_i, L_i, r_i, h_i\} & \\ \wedge\, CAT_{AP_i} \,\epsilon\, AP_i & \\ where\ AP_i \,\epsilon\, MNei_{AP_k} & \end{aligned} \tag{3}$$

The CA tuple identifies accurately the coverage area of the RDC's AP and comprizes of the following component information:

- the latitude (l_i) of the position of its AP.
- the longtude (L_i) of the position of its AP.
- the radius (r_i) of its AP air interface.
- the height (h_i) of the position of its AP. This may be employed only in the case of vertical handoffs.

These component information are shown in Figure 10. Vertical handoffs are considered in later versions in the proposed proactive mobility model.

Each RDC is sending its own CAT as an RDC-specific context option into a **D-RNV Update** message to its neighbouring RDCs. The D-RNV Update message also includes the prefix length of the sending RDC. From that it is possible for the receiving RDC to derive the LLA of the sending RDC by reversing the EUI-64 interface identifier generation rules. Alternatively a flag may be added to the message to signal explicitly the existence of LLA of the RDC in the D-RNV Update message. All RDCs are assumed to be configured to listen to the well-known 'all-routers' multicast group with *link-local scope* (i.e FF02::2).

Depending on whether an RDC is the originator of a CAT or it is forwarding a received CAT to other RDCs the following actions are effected:

[14] as in the event of prefix renumbering

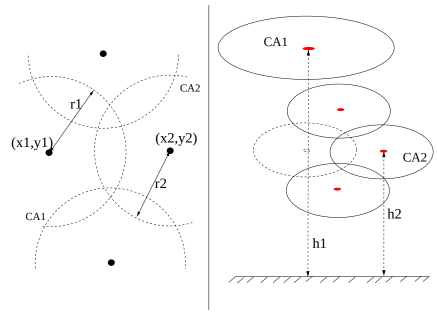

Figure 10: Coverage area tuple components

- If the RDC is the **originator** of the CAT and subsequently of the D-RNV Update, the latter is sent on all interfaces except for the link that the effects the coverage area. The destination address of the D-RNV Update is set to be the well-known 'all-router' multicast address with link-local scope. Each D-RNV Update packet carries a monotonically increasing sequence number.

- On receipt of a D-RNV Update by the RDC, the message is forwarded after processing, on all interfaces except for the incoming one as well as the interface that effects a coverage area, if existent. Similarly, the destination address of the D-RNV Update is set to be the well-known 'all-router' multicast address with link-local scope.

- The receiving RDC, checks a D-RNV Update packet for duplicates by means of maintaining a small table, denoted as **Known RNV Source (KRS)** table. This table maintains the *originator* and the *sequence number* of the D-RNV Update received. Any D-RNV Update with a matching originator entry and a larger sequence number, is silently discarded by the forwarding RDC.

- The RDC originating the CAT in the D-RNV Update provides further a *distance* metric with a constant value of 7. The distance metric is decremented at each RDC receiving the message, before it is forwarded further. When the distance metric decreases to zero, the D-RNV Update is *not* forwarded any further. The value of 7 is derived heuristically from the worst case scenario where a topology mapping to a single mobility neighbourhood is as shown in Figure 11. This ensures that forwarding of CAT state is constrained within segments of the topology that make sense in terms of a mobility neighourhood mapping. This value should change if the cell is considered to be other than hexagonal[15].

[15] this will instigate assymetries in cell overlap and cell radius

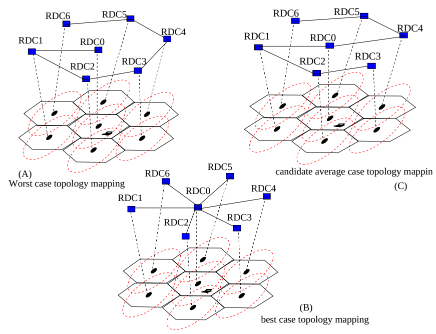

Figure 11: Topologies mapping to mobility neighbourhood

On receipt of the D-RNV Update the RDC extracts the CA tuple and the address of the originator. It then proceeds to calculate the **linear distance** **(d_l)** between its own AP and the originator's AP using (l_i,L_i) from the received CA tuple as well as its own (l_{i+1},L_{i+1}).

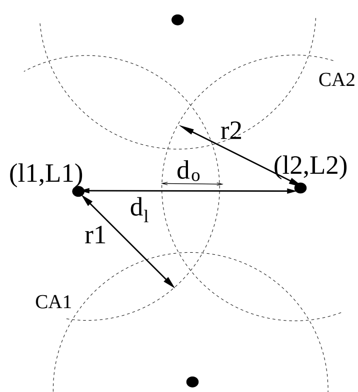

Figure 12: adjacency and overlap calculation

Having calculated the linear distance d_l, the receiving RDC then caclulates the **overlap distance** d_o as follows:

$$k_1 = d_l - r_1 \quad (4)$$
$$k_2 = d_l - r_2 \quad (5)$$
$$d_o = d_l - (k_1 + k_2)$$

The overlap distance can provide with a clear answer to the receiver RDC, whether the two RDC map to adjacent coverage areas. That is, if d_o is larger than zero then there exists some overlap for the particular two RDCs and thus, CA adjacency. In this fashion one RDC can safely augment its proactive mobility state, in terms of its RNV, with respect to its mobility neighbourhood without being assisted from the MN.

It should be noted that the messaging described is currently targeted to be implemented as ICMPv6 messages. However, the messaging may well be integrated with routing protocol specifics and implemented as UDP.

Distance calculation The linear distance between the positioning of two APs, assuming spherical Earth, may be calculated as:

$$\begin{aligned} d_l &= r.d_a \qquad (6) \\ &\quad where\ r\ = 6371.23\ km\ (radius\ of\ Earth)\ \wedge \\ d_a &= arccos(cos(l_i)cos(l_{i+1})cos(\delta L) + sin(l_i)sin(l_{i+1})) \qquad (7) \\ &\quad where\ \delta L = L_{i+1} - L_i \end{aligned}$$

However, for very small distances between APs the angular distance[16] d_a must be optimized by means of half angles to:

$$\begin{aligned} d_a = 2asin(min(1, \sqrt{sin(\frac{\delta l}{2})^2 + cos(l_i)cos(l_{i+1})sin(\frac{\delta L}{2})^2})) \qquad (8) \\ where\ \delta L = L_{i+1} - L_i\ \wedge \\ where\ \delta l = l_{i+1} - l_i \end{aligned}$$

4.2 Establishing proactively full Roaming context for the RDC

Given that each RDC maintains a complete RNV on the mobility neighbourhood effected over the coverage area of its AP, it can then proceed to acquire proactively also, the roaming state of all RDCs mapping in its mobility neighbourhood. This would identify the full *Roaming Context* for the 1-level lookahead-view from the perspective of the RDC utilized for management of complete proactive state to an MN.

With regard to the type of messaging there are two perspectives that should be considered, particularly in the light of differentiation of context classes over a communication protocol for context state transfer:

- Use of ICMP messaging. Each type of context should be 'multiplexed' over ICMP messages through different code types of context. ICMP more tightly bound in the IP stack within the kernel.

[16] angular distance measures in radians. For degrees convert to radians using $rad = deg\ \pi/180$

- Use of IP-Multicast. It provides a more persistent but less reliable perspective of signalling, where each multicast group may identify an individual context type. RDCs that determine their mobility neighbourhood can tune their membership to receive only from specific sources (source specific multicast).

Currently, we identify two individual ICMPv6 [6] messages that are required. These are instantiations of what the model identifies as ***Proactive Context Neighbour (PCN)*** Discovery (PCN-D). With respect to roaming state, PCN Discovery messaging proactively informs the **mobility neighbourhood** of an RDC about the **mobility neighbourhood of their neighbour RDCs** as effected by a set of adjacent coverage areas each controlled by different RDCs. This is done by providing full routing information about the neighbouring RDCs' link interface which belongs to the mobility neighbourhood and over which a router advert is made available.

The exchange of such roaming context is effected *only* between a set of neighbouring RDC; it may be periodicaly advertised or explicitly solicited by an RDC. For the first case, the PCN discovery takes the form of a **PCN Advertisment** (PCN-Ad), while in the second case the message is explicitly solicited as a **PCN Solicitation** (PCN-Sol). Figure 13 depicts the receipt of PCN-Ad/Sol message by a single RDC. We differentiate from the conventional notion of advertisment and solicitation over conventional neighbour discovery in the sense that PCN discovery *need not* take place over the local link. As such globally routable IPv6 addresses may be used for source or destination of such signalling packets unless stated otherwise. N is the number of RDC neighbours mapping to the mobility neighbourhood of the AP's coverage area controlled by the sending RDC.

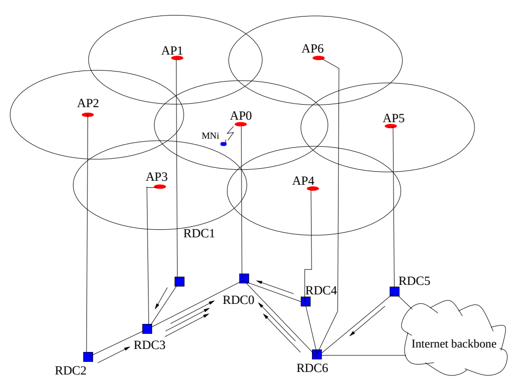

Figure 13: Neighbours inform Proactively the current RDC

The current routing domain RDC, creates a cache of its neighbours of size PCN_k, where k is the size of the mobility neighbourhood This is part of the **PCN Cache** (PCN-C) and is required in each RDC. The PCN Cache is also expected to maintain context state information for other potential context classes; however, the current version of this document focuses specificaly on what has been defined as roaming context.

For the purposes of clarity, an RDC (A) whose IPv6 address is found in the RNV of another RDC (B) is denoted for the remainder of this document as a *neighbour RDC* of B.

At each RDC, the PCN Advertisment message first signifies the existence of roaming context; it further includes the RNV identifying the mobility neighbourhood of each RDC as well as the prefix length on the mobility-enabled link interface where the router advertisment is effected. The latter is combined with EUI-64 rules to deduce the LLA of each neighbouring RDC if needed. The source address of the PCN Advertisment is set to be the IPv6 address of the sending RDC's link interface on which the router advertisment is made available at the particular coverage area.

The PCN Solicitation simply signifies the need for Roaming Context provisioning through a PCN advertisment. The current version of the model proposed does not specify explicitly a subset of the mobility neighbourhood over which it would require specificaly it roaming context. It simply signifies the need to obtain the roaming context of the *entire* mobility neighbourhood.

A PCN Advert may be **periodically** unicast by a single RDC to the RDC neighbours. Alternatively a PCN Solicitation is **explicitly** unicast by an RDC to its neighbour RDCs. A PCN Solicitation should piggyback Roaming context of the sending RDC to the receiver RDC to speedup convergence of roaming context state on all neighbouring RDC. In response, the RDC neighbours reply with a PCN Advert that carries the roaming context of their mobility neighbourhood.

Proactive Context Discovery **does not** await on the MN reaching any coverage boundaries between the CURRENT RDC/AP and any future candidate RDCs. On the contrary, such exchange of neighbour information is happening on the background between RDCs while the MN is registering with the CURRENT RDC.

From a seamless mobility perspective, reactivity over coverage boundaries between RDCs would be considered too late an action on behalf of either the CURRENT or NEW RDC or even for triggers that initiate the handoff process when the MN is transiting the overlap area. This latency is expected to also propagate over potential context tranfer relocation which is bound to be time-consuming. We argue that as the velocity vector of the MN becomes highly irregular, non-proactive mobility recommendations are bound to be severely constrained in terms of reaction times; this is especially true in cases that the MN alternates fast between coverage areas of different RDCs.

4.2.1 PCN-Cache management

With respect to roaming context, PCN-Cache entries MUST not be purged periodically or according to some timer value unless explicitly requested. In addition the periodic advertisment of PCN-Ad or PCN-Sol messages should be injected with some small random delay [16], the magnitude of which is dependent on the frequency of transmission of the PCN-Ad message. In the event that a PCN-Ad *transmission interval* is small (smaller than the observed frequency of transmitted PCN-Sol messages), a PCN-Sol message will be serviced by the next PCN-Ad at time that syncronizes with the *PCN Advertisement interval* plus some random delay [16].

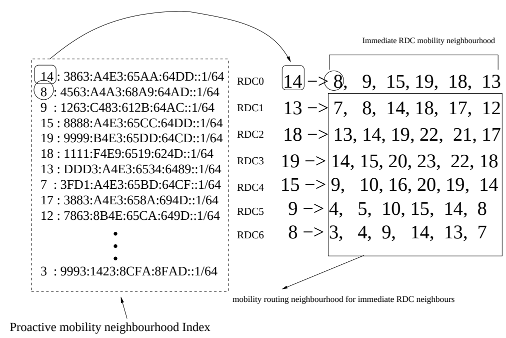

Figure 14: Roaming context in PCN Cache maintained at node 14 (RDC0). Number 14 would point to the RDC's prefix on that link connecting the two RDCs

Figure 14, illustrates an outline of the PCN Cache for the roaming context. The first entry signifies the mobility neighbourhood of the CURRENT RDC (which maintains this cache). Subsequent entries signify the mobility neighbourhood of that RDC's immediate neighbours.

4.3 Registering with the home network

Initially the mobile node bootstraps and registers with its home network and, thus, obtains a home IPv6 address. This allows also the MN to store information about its home network and subsequently Home Agent when designated through during the Home Agent Discovery phase.

Home network registration is important for the MN in the event that it roves away from it with its network interface disabled or when the MN is dormant/switched off. It allows the MN to inform its home network, upon boostrap at a foreign network, by means of a binding update. This also tackles the case of home network renumbering.

The registration process provides upon bootstrap, the **link-layer address (LLA)** of the MN, to the home RDC (or any RDC in the CURRENT state). As soon as the link layer address of the MN has updated the home RDC, the latter may propagate it to any RDC neighbours that the MN may traverse in the future, as part of the roaming state generation process effected at the CURRENT or neighbour RDC that is mapping over the mobility neighbourhood to be visited.

On the reverse direction, during first time registration with the home network, the CURRENT (home) RDC provides to the MN its own LLA and its IP address. The MN is expected to store such information in its neighbour cache and its default router list. The MN must not discard these entries, when it hands off to a new RDC unless explicitly requested to do so, since they would be utilised amongst others[17] for the purposes of probing the routing neighbourhood.

4.4 Proactive Roaming state generation for the MN

Subject to admission control, the registration or bootstraping process of the MN with some routing domain (the home or a foreign one) enables the MN with IP connectivity. Upon completion, the controlling RDC transits from the NEW to the CURRENT state. Transition at this state requires that the CURRENT RDC generates a **tuple** of **soft** (unicast) **Care of Addresses (SCoAT)** for the MN admitted.

Such SCoA tuple comprizes of unique IPv6 unicast addresses, each of which must be topologically correct in one of the CURRENT RDC's neighbours. By producing this SCoA tuple, the CURRENT RDC proactively enables the MN with valid **roaming context** of configured IPv6 addressing over prospective RDC neighbours that map to the mobility neighbourhood of the CURRENT RDC's coverage area. Some part of this mobility neighbourhood is expected to be immediate candidate for the next handoff transition of the MN.

The SCoA tuple may be generated through two alternative ways:

- The CURRENT RDC utilizes the neighbour RDC prefix entries from the roaming context state maintained in its Proactive Context Neighbour Cache. It combines each of the neighbouring RDC routing prefix with the LLA of the MN according to [31] and produces an IPv6 address. Each of the IPv6 addresses created by the CURRENT RDC must now be checked for duplicates (DAD) against the neighbour Cache (NDC) of the neighbouring RDCs. The LLA is expected to be found in the neighbour cache of the CURRENT RDC since during boostrap the MN needs to update the neighbour cache of the RDC with its LLA for on-link forwarding purposes according to the standard neighbour discovery protocol.

  It is intuitive that there will be more than one MNs registering with RDC_0 and as such this RDC will need to perform multiple Duplicate Address Detection checks per neighbouring RDC, each one representing an SCoA for a single MN. For that case, the proposed scheme may be optimized with an aggregate*Proactive Indirect Duplicate Address Detection (PI-DAD)* check per neighbouring RDC for the RDC in the CURRENT state.

  A standard DAD check under IPv6 is effectively address resolution for a tentative address; that includes neighbour solicitation and neighbour

[17] as we see it will also comprize common roaming state between the CURRENT and the NEW RDC

advertisment as described by [33]. This is usually performed on-link upon generation of new IPv6 address which is also topologicaly correct on the local link. If the neighbour solicitation is not responded with a neighbour advertsment within some time interval the address is considered to be unique.

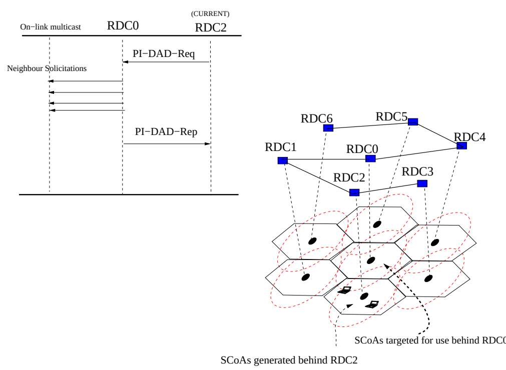

Figure 15: Aggregated Indirect proactive DAD check

With the case of proactive IPv6 address generation, however, the IPv6 addresses are generated off-link and are targeted to be topologically correct on the prospective on-link. As Figure 15 shows there exist some IPv6 SCoAs that are generated behind RDC2, however they are targeted for use behind RDC0. It is for that purpose an extra level of indirection need be generated in terms of signalling at the attachment point where the addresses get generated. This is created by means of the PI-DAD Request message sent by the CURRENT RDC towards the RDC for which the tentative SCoAs are topologically correct. The aggregate PI-DAD Request message comprizes of one or more[18] ICMPv6 packets. Each aggregate PI-DAD Request includes a set of SCoA that are topologically correct at the RDC neighbour. Furthermore for each of the SCoA the CURRENT RDC includes also the link layer address (LLA) of the MN.

On receipt of the PI-DAD Request, the neighbour RDC first checks its neighbour cache entry for each of the SCoAs to see if they are already existent. If not, it creates an entry with the SCoA and the link layer address of the MN. It also adds a P flag to the entry, as an extension, to mark it as PROACTIVE. The entry is set to INCOMPLETE. It then attempts DAD through standard address resolution. In particular it sends neighbour solicitation to the solicited-node multicast address mapping to the SCoA. It further includes a Source Link Layer address option with its own link layer address as the sender. The SCoA address is found to be

[18] if SCoAs are more than can fit in a packet with MAX MTU

unique if no neighbour advertisment is received back within `RetransTime` milliseconds.

As soon as uniqueness of the soft CoA is ensured at the RDC neighbour, the neighbour cache entry is marked as PROACTIVELY REACHABLE. This is a new state introduced for the purposes of enabling proactively the neighbour cache entry *without* requiring a solicited neighbour advertisment. This is illustrated in Figure 16. If a neighbour solicitation is sent towards that RDC (from that link) that requires the LLA of the MN, then the RDC defends the soft CoA with its own LLA. This is because such neighbour solicitation is probably for the purposes of neighbour reachability or DAD rather than traffic forwarding to the soft CoA. This is because usually any host that attempts to contact the MN, will do so through its home address rather than directly using its CoA, since its does not know it. In the event that a node is contacted directly through its soft CoA by a host without going through its home network and while the MN has not yet handed off to that RDC, then buffering at that RDC may be considered.

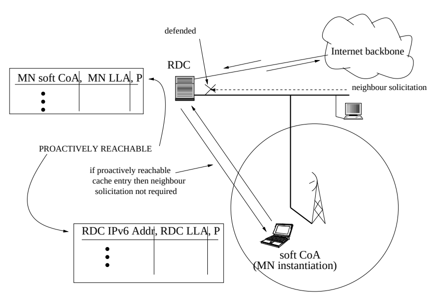

Figure 16: Proactively Reachable neighbour cache entry state

By setting the link layer address of the MN in the neighbour cache proactively and marking it with the P flag, the neighbour RDC requires minimal information in order to effect that LLA when traffic needs to be forwarded to and from the MN's particular soft CoA. This is because both the accomodating RDC neighbour and the MN are configuring their neighbour cache entry proactively with the particular entries marked as proactive and set in the state PROACTIVELY REACHABLE. In this manner, communication between the two entities does not need to go through a solicited neighbour advertisment explicitly; it can effect communication of packet traffic immediatelly. The period for which a neighbour cache entry is set to the PROACTIVELY REACHABLE state is until the MN has sent a Binding Update to towards its peers. This effects a stable active primary CoA (as detailed in following sections) and thus does not require proactive reachability in the neighbour cache of either the MN or the new

CURRENT RDC.

Upon succesfull completion of DAD for each of the SCoAs, instead of configuring an interface, the neighbouring RDC joins the SCoA to both all-nodes and respective solicited-node multicast address. This way, the neighbour RDC can defend with neighbour advertisment the SCoA allocated proactively for the MN.

Each aggregate PI-DAD request message is acknowledged back to the CURRENT RDC with a respective aggregate PI-DAD Reply. The reply contains an address bitmap which describes which addresses have succeded DAD. Each SCoA that has been checked for DAD successfully, is marked with 1 while each unsuccessful one is marked in the bitmap with 0.

An aggregate PI-DAD check is expected to save individual DAD packet header overheads by means of aggregating DAD checks for each neighbouring RDC. An RDC awaits for a maximum period DAD_AGGREGATE_INTERVAL before its triggers a proactive aggregate PI-DAD request; this is done so that some DAD requests have been aggregated before the CURRENT RDC requests the check from the neighbouring RDC.

Alternatively, a PI-DAD request message may be submitted to each neighbour RDC_i found in the PCN Cache, for each SCoA. It is known that the probability of a duplicate IPv6 address detected is very low[19]; however, for reasons of precaution in the event of a vendor MAC address provisioning conflict, the IPv6 community feels that DAD is not necessarily mandatory but would be recommended. For this reason we consider as optional the reliance of every PI-DAD-Req on a returned **Proactive Indirect DAD Acknowledgement (PI-DAD-Ack)**. In the event that both messages are to be effected[20] then the mobility neighbourhoood of the CURRENT RDC will receive for each SCoA tuple mapping to an MN the set of PI-DAD requests denoted by:

$$\begin{aligned} PI-DAD-Req_{RDC_0} = \{ & PI-DAD-Req_{RDC_1}, PDAD-Req_{RDC_2}, \\ & PI-DAD-Req_{RDC_3}, PDAD-Req_{RDC_4}, \\ & PI-DAD-Req_{RDC_5}, PDAD-Req_{RDC_6}\} \end{aligned} \tag{9}$$

while similarly this may be acknowledged by:

$$\begin{aligned} PI-DAD-Rep_{RDC_0} = \{ & PI-DAD-Rep_{RDC_1}, PI-DAD-Rep_{RDC_2}, \\ & PI-DAD-Rep_{RDC_3}, PI-DAD-Rep_{RDC_4}, \\ & PI-DAD-Rep_{RDC_5}, PI-DAD-Rep_{RDC_6}\} \end{aligned} \tag{10}$$

[19] 2^{64}

[20] [need to set a flag that enables the optional messaging]

- The CURRENT RDC unicasts the LLA of the MN to the neighbouring RDCn in a **soft CoA Create** message. This is a relatively light message sent in the unicast packet. Each of the receiving RDCs is empowered with the IP generation task; as such it combines the LLA of the MN into an IPv6 (soft) CoA; it then performs an Indirect DAD on the SCoA by triggering a neighbour solicitation in a fashion similar to the one for aggregate DAD. It then generates a neighbour cache entry that is also marked with P flag enabled as soon as DAD has succeeded for this SCoA. The entry is also set to the PROACTIVELY REACHABLE state. This neighbour cache entry include the LLA of the MN rather than the LLA of the RDC itself. The existence of the P flag together with the PROACTIVELY REACHABLE state for the particular neighbour cache entry, establish behaviour described for aggregate PI-DAD and illustrated in Figure 16.

  In response the neighbour RDC returns the SCoA into a unicast **soft CoA Ready (SCoA-Ready)**. It may be noted that the generation of the Soft CoAs is now distributed, while the DAD function is performed on the spot during address creation time. As such no DAD function need be proxied by the CURRENT RDC. Each address is sent in a unicast message back to the CURRENT RDC which in turn combines them into the SCoA tuple ready to be provided to the MN.

From the two alternatives for SCoA generation the proposed scheme opts for the second approach as more compact and efficient from a signaling perspective. The signal interactions for the two approaches are illustrated in figure 17.

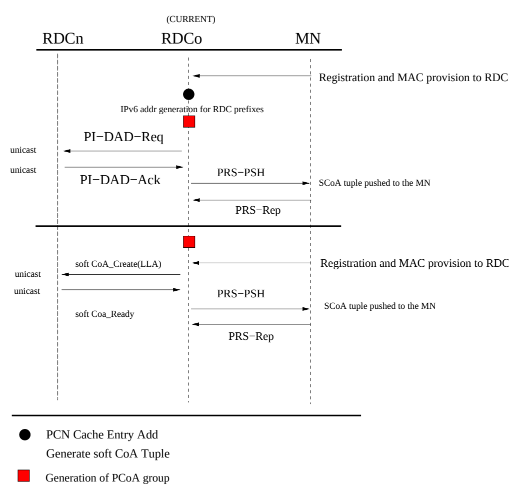

Figure 17: Signaling for the provision of SCoA tuple

The CURRENT RDC must now proceed to *inject* the Roaming context established, to the admitted MN. This may be achieved through a **Proactive Roaming State Push** (PRS-PSH). This message includes the number of neighbours for which proactive roaming context state is established. It further contains an SCoA tuple identifier which uniquely identifies the particular SCoA tuple and is associated with the respective binding cache entry for the MN provided this roaming state. For each neighbour RDC the following established state:

- the IPv6 address of the neighbour RDC interface that effects part of the mobility neighbourhood. This is expected to be used in the update of the default router list, as provisional default router list entries that are provisionally effected during handoff.
- the prefix length valid on the above interface. Essential to derive link layer information for fast update of the neihbour cache of the MN. Such information targets to minimize or eliminate neihgbour discovery signaling that will be required when the MN appears on-link to some neighbouring RDC.
- the generated soft CoA for MN. A globaly routable IPv6 address that can communicate traffic on-link to some neighbouring RDC.

From this information the MN can further reconstruct other routing information that complement the roaming context in the mobility neighbourhood. This is:

- the link layer address of the neighbouring RDC. It may be derived by using the prefix length together with standard EUI-64 rules for interface identifier generation. This is to be used in the neighbour cache of the MN with the flag P set (PROACTIVE) and set in the PROACTIVELY REACHABLE state.
- routing prefix. This may be derived by straightforward use of the prefix length to determine the routing prefix of the neighbouring RDC and populate the prefix list of the MN.

The PRS-PSH message from the RDC in the CURRENT state *must* be acknowledged by the MN with a **PRS Reply** (PRS-Rep). It is important that the CURRENT RDC receives feedback by the MN on the successful receipt of the PRS-PSH since the SCoA tuple will be facilitated to signal the current active CoA upon transit to a neighbouring subnet. This message simply contains the number of neighbours as well as the sequence number included in the original PRS-PSH message.

It should noted that the proposed scheme does not encourage the creation of the SCoA tuple at the MN (stateless address configuration). Reason for that is the number of signaling interactions required for the generation of the SCoA tuple. The second and strongest reason is dependence on the response of the

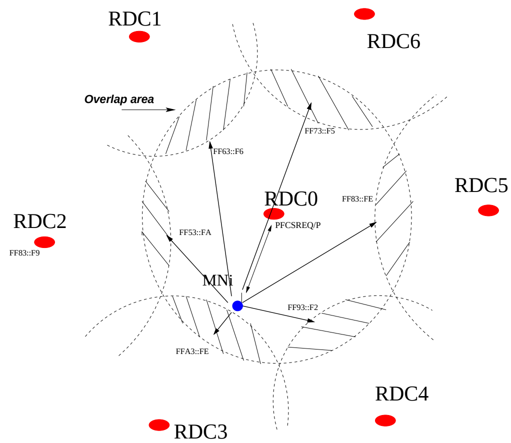

Figure 18: The proactive set of soft CoAs provides 1-domain lookahead network connectivity

MN back to the CURRENT RDC must be avoided at all costs. This is so because such dependence will mean:

- the neighbouring prefixes need to be transmitted to the MN for the purposes of the SCoA tuple generation.
- the probability of latency between RDC and MN increases, should the reply message from the MN (carrying the created SCoAT by the MN) gets corrupted due to harsh fading conditions when the response is propagated back to the CURRENT RDC.
- in the event that a check (such as DAD or ND) of the created (by the MN) SCoAT is required then the MN would not be able to utilise any of the CoA in SCoAT (as descibed in later subsection) unless a response is sent back by the neighbouring RDCs back to the CURRENT RDC, which then has to be sent back to the MN.

4.5 Abstracting the Routing Identifier on the RDC

The CURRENT RDC is also required to map the SCoA tuple generated for the MN, onto a single **Proactive Care of Address (PCoA)**. The PCoA address is *not* a unicast but a **multicast** IPv6 address. It is assumed that all RDCs are multicast-enabled. The IPv6 protocol architecture mandates IPv6 Multicast capabilities on the routers as the replacement of broadcast (in IPv4) for core routing. As such, the previous assumption is valid. The mapping of the SCoA tuple onto a PCoA is effected *only* at an RDC. Such mapping is transparent from the perspective of the peer entities of an MN, namely its HA and CNs. That implies that both HA and CNs continue to send packets towards the MN through the CURRENT RDC with no knowledge about the existance of such mapping. The RDC by effecting the PCoA abstract routing identifier mapping over the SCoA tuple allows the MN to receive traffic on any of the candidate transition coverage areas of the (CURRENT) RDC's mobility neighbourhood. More accurately, to receive traffic on any future coverage area for which the soft CoA allocated participates in the mapping to the PCoA (multicast) routing identifier.

With respect to the aforementioned mapping (Figure 19), this involves:

- allocation of the multicast Proactive CoA (PCoA).
- submission of PCoA group membership reports for the individual soft CoA instantiations that are valid for the MN over the different RDCs.

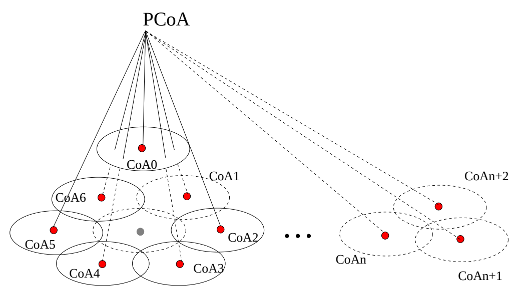

Figure 19: Mapping a multicast address to unicast IPv6 CoA listeners

The PCoA group allocation process takes place only once at the **home** RDC, as soon as it transits into the CURRENT state for the particular MN. PCoA group membership reports are expected to be performed by all candidate RDCs that will be accomodating the MN during its movement from one subnet to another. These candidate RDCs are expected to be the RDC neighbours for the CURRENT RDC that accomodates the visiting mobile node.

While the multicast address allocation and management is also an important task for the scalability of the proposed model to dense populations of MNs, we assume a multicast PCoA is uniquely allocated for the purposes of this scheme. Currently, this aspect is pursued by the IETF Multicast Address Allocation WG (MALLOC); The MADCAP protocol [18] proposed in the MALLOC WG fits exactly this purpose. As such any further analysis on multicast address allocation is out of the scope of this document[21].

The allocated PCoA (multicast group) is placed in the PRS-PSH message before this is sent to the MN; this provides the MN with its abstract routing identifier that is effective during handoffs. The inclusion of the PCoA in the

[21] we investigate multicast address allocation and management issues for the purposes of this scheme in a separate draft

roaming state pushed to the MN allow the latter to be autonomous over its roaming state in the event of losing network connectivity or purposely switching off its network interface while transiting to a different RDC where it attempts to re-establish connectivity. This scenario is fairly common for MNs moving under constrained conditions like, underground, non-continuous coverage areas, forced switch-offs for safety reasons[22].

4.5.1 PCoA membership management for RDC neighbours

Following allocation, the CURRENT RDC needs to inform each of the neighbouring RDCs for which there is a soft CoA listener in the respective tuple, to enable forwarding of traffic destined to the multicast PCoA group in their downstream interface for which the allocated soft CoA is valid. This would allow the respective RDC neighbour to assume that there exists (or is probable to exist) at least one host that is interested in the traffic destined to the multicast PCoA group. Since multicast forwarding is completely independent of the number of listeners as well as their IP identity, the neighbouring RDC needs minimal information to enable multicast forwarding for that PCoA group. In fact, it only requires the multicast PCoA group as well as the IPv6 prefix that maps to the particular network interface that should forward multicast traffic on that cell of the mobility neighbourhood.

The CURRENT RDC employs two potential approaches of informing the entire RDC neighbourhood, including itself in an IGMP-compliant [11, 14, 10] fashion. These are:

- request an **explicit 'join'** from MN
- request an **implicit 'join'** from neighbouring RDCs

In the case of the *explicit join*, the CURRENT RDC *solicits* an explicit request to join the PCoA group. The join request is directed towards the MN; this is because the CURRENT RDC must ensure that the MN will configure its hardware interface for the particular multicast $PCoA_{MN}$ group. This **explicit join solicitation**, is piggybacked in the PRS-PSH message to the MN by means of a join-bit flag (J). Upon receipt of the J flag the MN proceeds with a multicast join for the PCoA group and the particular interface over which the PRS-PSH was received. The particular multicast join effected by the MN will configure its multicast filter on its hardware interface [7] and also provide the CURRENT RDC with an IGMP membership report which enables multicast forwarding for at least one listener on that link. The Multicast Listener Discovery Protocol [8] ensures that the IGMP membership report is sent by the MN using its link-local IPv6 address as the source address of the membership report.

In the case of the *implicit join*, the CURRENT RDC solicits an implicit join request. This join request is solicited to *some or all of its neighbouring RDCs*. Each of the neighbouring RDCs receiving an **implicit join solicitation (I-Join Sol)** need *also* enable multicast forwarding for a particular PCoA group

[22] like aircrafts or any other vehicle prone to RF interference

over the link for which the advertised prefix was originally used to construct a valid soft CoA; i.e. the link on which the soft CoA is topologically correct.

The implicit join solicitation message sent by the CURRENT RDC includes only the PCoA multicast address. This so since in multicast a router needs to learn only that a listener exists on link. It does not need to know the identity of the listener and as such does not need to include either the SCoA or the link-local address that the MN would have on that link.

It is reminded that any standard Multicast Listener discovery (MLD) message in terms of Queries and Reports must be sent with a link-local source address at the local link, for the particular downstream RDC interface to receive traffic sent to a multicast group. That is to say that under traditional MLD signaling, the MN must send an MLD membership Report on-link by using its link-local address as the source address of the MLD message. Figure 20 illustrates the issue where a neighbouring RDC (e.g RDC6) has to enable group membership by implicitly receiving a join solicitation from the CURRENT RDC (RDC0).

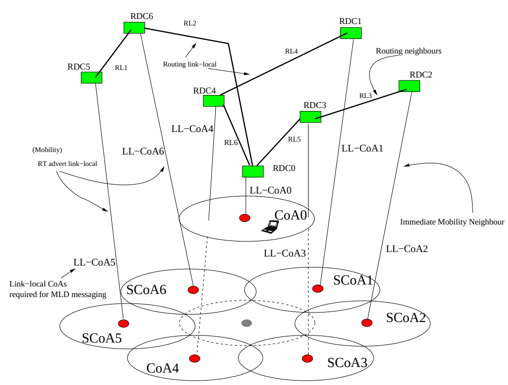

Figure 20: link-local addresses used for conventional MLD messaging

The soft CoAs effected in the mobility neighbourhood are only instantiations of the MN on the link of the neighbouring RDC. The MN is not physically on-link at that neighbour RDC yet. It is thus required that the group membership for the PCoA is managed **indirectly**. The following section presents how the issue may be alleviated by introducing indirect multicast group management.

4.5.2 Requirement for indirect Group management

As seen in Figure 20, a standard MLD Membership Report normally reaches RDC1 through LL-CoA1. However, for the purposes of the proactive mobility model, such report must now come through RL6 and then through RL4, since the MN is not currently residing in the respective coverage area of RDC1 but in the one of RDC0. For that purpose the proactive mobility model proposed, extends the standard MLD protocol to handle indirect multicast listener discovery which is triggered by a solicited, indirect join request.

In particular, on receipt of an I-Join Solicitation, the receiving RDC sets an entry in its multicast group membership list. The entry records the PCoA group and sets a timer for the membership to the `Group_Membership_ interval` as defined by [14]. The receiving RDC then generates an **Indirect MLD Membership Query (I-MLD Query)** with destination address the CURRENT RDC which solicited the I-Join message. The I-MLD Query message is a *Multicast-Address-Specific* query and is periodic; it targets membership on a specific multicast group (in this case the PCoA group) and is sent by the querying neighbour RDC every `Query_interval`. An I-MLD Query contains the PCoA address for which group membership is managed indirectly, as well as a maximum response delay time within which the Reporting RDC must report back.

On receipt of an I-MLD Query by some neighbouring RDC the CURRENT RDC sets a delay timer for that PCoA group to a random value from the range [0, `Max_Response_Delay`]. The timer adjustments obey the rules defined in standard MLD [14]. On expiry of the delay timer the CURRENT RDC transmits an **Indirect MLD Membership Report (I-MLD Report)** with destination IP address the query-originating RDC. This is shown in Figure 21.

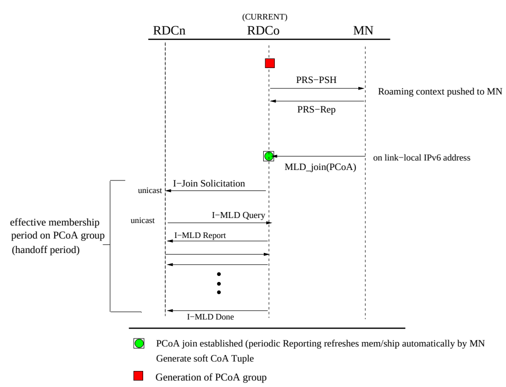

Figure 21: Periodic I-MLD messaging between RDCs

Repeated I-MLD Reports refresh the timer set in the group membership list of the neighbouring RDC that originated the I-MLD Group-specific query. If no I-MLD Reports for the PCoA are received at the querying RDC after the response delay of the last I-MLD Query has passed, the querying neighbour RDC assumes that indirect group membership for that PCoA must cease to be in effect. As such the PCoA address is removed from the membership list and the removal is made known to the multicast routing component (link prune).

It is intuitive that the neighbour RDC that provides a soft CoA in this proactive mobility model becomes also the Querier for indirect group management[23] over the PCoA mapping.

Both I-MLD Query and Report messages must be sent with a max hop limit of 7 and an IPv6 Router Alert option in a Hop-by-Hop Options Header. The router alert value for the P-MLD Report must be set to 13. The hop limit assumes a hexagonal coverage area pattern and thus a mobility neighbourhood of 7 RDCs. If I-MLD messages traverse more hops than the hop limit identified then either the coverage area has a different pattern or there is an error in the realised topology.

I-MLD messaging is exchanged between RDC entities. It must not be exchanged with a host/MN. It also requires the introduction of a P flag in the group membership list of an RDC; this flag signifies a *proactive* group membership entry that is valid on-link for the mobility enabled interface of that RDC.

4.5.3 Indirect group management considerations

It is possible that when the first I-MLD Query message is sent by a neighbour RDC to the CURRENT one, the `Query_interval` is set such that PCoA group membership is marked as **persistent** until instructed otherwise. That implies that the choice of the query interval is possible to instigate persistency in indirect group membership management.

This may be effected by setting the **Query_interval** value for the particular group membership entry to infinity and assume **persistent** membership at the PCoA until instructed otherwise. The query interval that denotes infinity is set to be -1. However, in that event both the original query and the respective I-MLD Report must be acknowledged in order to effect a robust indirect PCoA join. Unacknowledged I-MLD messages are retranmitted. The set of involved interactions are shown in Figure 22.

A value of infinity (-1) for the Query interval of the querying RDC neighbour implies that the latter effectively suppresses periodic I-MLD messages for the particular interface, with multicast forwarding on that interface persisting indefinetely. To revoke this persistency the CURRENT RDC after transiting to the PREVIOUS state must send an explicit **I-MLD Done** message to the neighbouring RDC that initiated the original I-MLD Query. Transition to the PREVIOUS state implies a handoff of the MN to a new RDC.

In this manner, the PREVIOUS RDC explicitly directs some RDC neighbour to stop forwarding traffic, destined to the PCoA group, on its mobility-aware interfaces. This is when an **active primary** CoA transits to the **inactive secondary** state with **lowest priority candidate** designation; namely the CoA is not the CURRENT one and it is not an highest priority candidate for handoff. The I-MLD Done excludes the PREVIOUS RDC itself since it initially received an explicit join (standard MLD Report); as such standard MLD will

[23] it becomes clear that if RDC do not initially provide a soft CoA, they decline participation in this proactive mobility model. For these RDCs proactive mobility will not be effected

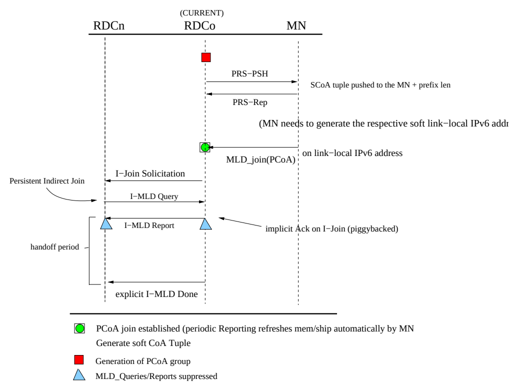

Figure 22: Persistent MLD join on PCoA

query the link every query interval (125 sec) to confirm that membership on-link for the PCoA group is still required.

The primary advantage of this *persistent* join approach is that I-MLD signaling is saved in terms of queries and reports when PCoA membership has to be maintained for all neighbouring RDCs. The disadvantage is, however, that explicit notification upon leave is required rather than invalidating an expired group membership entry for the mobility-aware interface.

On the contrary, the periodicity of queries provides mainly robustness to the group membership protocol; that is, if an initial report gets lost, membership is recovered at the next query interval while on a lost *I-MLD Done* message, membership simply expires since there is no response to periodic queries. Eliminating periodic querying and reports requires that this signalls are made **robust** so that multicast can work effectively. It is for this purpose that the proactive mobility model opts for the non-peristent group membership management.

We should note that MLD is tracked also by availability of different versions of IGMP. It is for this reason the correct version dependent Query interval field is updated accordingly depending on the approach followed above.

4.6 Benefits from mapping of mobility neighbourhood on PCoA group

The gains from mapping between unicast CoAs and a unique multicast PCoA are significant for a number of reasons:

- The PCoA group abstracts bindings to multiple unicast (soft) CoAs at the CURRENT RDC. That implies that in essense handoffs over multicast are employed always at the RDC that is nearest to the MN.
- the above abstraction is hidden from HA and CN which operate transparently.

- eliminate dependency on updated unicast bindings until the MN is well-settled into a new RDC. This is because the MN is identified with a single routing identifier during handoff and for 1 RDC-controlled coverage area ahead.
- the only entity responsible for maintaining the mapping between the soft CoAs and the PCoA group is the CURRENT RDC. This RDC will always be closest to the MN before and during handoff.
- It abstracts more than one possible routing domains behind a single routing identifier. As such that enables the PCoA to identify not only the *current* active CoA but, furthermore, a set of *tendative* CoAs for that MN, comprizing its SCoA tuple.
- the CURRENT RDC, as the nearest point of attachment, has the necessary information to decide which CoA(s) from the SCoA tuple will join the PCoA. This caters for optimization over the selection of the best candidate[24], multicast-forwarding, RDCs that need to receive traffic on the PCoA of the MN as the latter transits between these RDCs.
- As a corollary to the above coverage area bouncing effects are simply eliminated from the handoff process.
- each candidate RDC will receive a *single* copy of the traffic that is destined to the MN.

4.7 Activity on the part of the MN

The receipt of the SCoA tuple through the PRS-PSH message, provides the MN with a one-level depth tree of proactively generated soft CoAs representing a *1-hop lookahead view*, shown in Figure 18, of the RDC neighbourhood with respect to the CURRENT RDC. Each individual soft CoA represents an instantiation of the MN in existence within the coverage area of a neighbouring RDC.

With respect to the SCoA tuple, available to the MN, it is *soft-enabled*; this implies that a single CoA has been *allocated* for that MN at some RDC neighbour but is yet to be used in the near future. Full routing information is also provided to the MN (through PRS-PSH) in terms of default router (the neighbour RDC itself) as well as a prefix size (i.e. netmask). Furthermore, the receipt of the PRS-PSH message implies that each RDC neighbour has proactively updated its neighbour cache with the LLA of the MN as described in previous sections.

The receipt of the PRS-PSH message must trigger the following actions on the part of the MN:

- it must return a PRS-Rep that acknowledges the receipt of the PRS-PSH message so that the CURRENT RDC can ensure that it can proceed robustly with the required set of interactions during handoff. The PRS-Rep

[24] the ones that show the highest probability of transiting to the CURRENT state

message includes the number of RDC neighbours that were received over the PRS-PSH message as well as the sequence number that identified the original PRS-PSH message. If the PRS-Rep message is not received within `Proactive_context_push_time` the original PRS-PSH message must be retransmitted.

- it must join the PCoA group iff the J and M flags area set. This is done by means of a standard MLD Report destined to the PCoA group and with source address the current link-local address. The standard unsolicited MLD Report is also expected to configure the LLA address of the MN for multicast filtering at its hardware network interface. Robusteness of PCoA membership is sustained according to the robustness variable value of the MLD protocol [8].

- it must update its default router list with the set of neighbouring RDC IPv6 addresses. These addresses are expected to be topologically correct on the link that identifies part of the mobility neighbourhood for that RDC neighbour. Each default router entry for the neighbouring coverage areas of the mobility neighbourhood must be placed at the end of the default router list and must be marked with a P flag. When the P flag is set the particular default router entry must not be used. A P flag is removed from a default router entry as soon as a PCoA-Enable message is generated by the MN towards the CURRENT RDC. When a P flag is removed each of the proactively configured default router entries are used according to the standard forwarding mechanisms of [9, 31].

  Alternatively the MN should update its prefix list by deriving each neighbour RDC prefix through the neighbour RDC IPv6 address and its prefix size. Each prefix list entry is also marked with a P flag. The rules defined for the default router list apply also for the prefix list.

- it must update its own neighbour cache with an entry for the LLA of each neighbour RDCs. Each such entry must be marked with the P flag and set to the PROACTIVELY REACHABLE (P-REACHABLE) state. Both P flag and P-REACHABLE state in a neighbour cache entry of the MN, signifies that the former must not be checked for address resolution or neighbour unreachability detection with a neighbour solicitation; this is so because a REACHABLE (albeit proactively) neighbour cache entry exists for the RDC neighbour in the MN. This is in effect until the P flag is removed from this entry. A P flag is removed from a neighbour cache entry if and only if the MN sets a soft CoA to the active state, becoming its primary IPv6 address. The setting of the P flag intends to allow the MN to transmit upstream packets without having to engage into either address resolution or neighbour unreachability detection, when it arrives on-link within the coverage area controlled by a neighbour RDC.

- it may multi-home its interface with the available soft CoAs. In this case, the MN should transmit or receive on any of the soft CoAs **if and only if** a PCoA-Enable message is sent by the MN towards the CURRENT RDC.

4.8 PCoA Activation Lifetimes

Having configured group membership of the mobility neighbourhood for the RDC neighbour, the CURRENT RDC must configure internally the PCoA group with two lifetimes L_s and L_d. The two lifetimes are mobility management specific and extend the standard MLD protocol as follows:

- L_s (PCoA start lifetime): represents the lifetime past which the CURRENT RDC should initiate sending of the unicast packets, that are destined to the MN, through the configured PCoA multicast address. Its initial value should be -1 which is designated as *infinity* for time values in the proposed scheme as described in previous sections.
- L_d (PCoA stop lifetime): represents the lifetime past which the CURRENT RDC should stop sending of packets through the PCoA towards the MN. This situation occurs when the CURRENT RDC transits to the PREVIOUS state, i.e. the MN hands-off to a NEW RDC. Effectively, L_d represents the lifetime for which the CURRENT RDC must forward the traffic destined towards the MN, through the PCoA group of the MN. Its initial value must be 0. It is expected that before the expiry of L_d lifetime, the MN will have informed its peers entities of its new *active* CoA with a standard Binding Update. In this manner, forwarding of traffic towards the MN over the PCoA group can be stopped from that particular RDC which has transited to PREVIOUS state.

While forwarding of the traffic, towards the MN, through the PCoA group is **stopped** *locally* at the PREVIOUS RDC, it only **suspends** *globally*. This is because traffic will be sent to the MN through the PCoA group during **every** handoff, from the next valid CURRENT RDC. This is the reason for requiring that the particular PCoA allocated to the MN should **persist** so as to avoid unnecessary reconfiguration costs of a new PCoA. Future versions, however, of this proactive mobility management scheme do not preclude PCoA reallocation under special conditions. It is beyond the scope of this version of the document to engage in analysis of such conditions.

Furthermore, since the CURRENT RDC changes as the MN moves across different cells[25], the multicast core [3, 35] or RP [12] router will effectively move together with the movement pattern of the MN. Since the average cell size for PCS systems is bound to be reasonably large (around 1-2.3 km), the movement of the multicast RP root[26] (PIM-SM) or core (CBT) will be relatively slow with respect to the movement speed of the MN. Furthermore with respect to PIM-SM if the data rate of the traffic destined to the MN, warrants it, the receiving active CoA will be listening to the shortest path distribution tree [12].

[25] when each cell or cell cluster is controlled by a different RDC
[26] shared tree

L_s and L_d is expected to be conditioned by the mobility vector of the MN in the case that optimal lifetimes should be pursued. The intention for these fields is to provide a time window during which handoff would be in progress as a rough estimate. More accurate lifetime refresh messages may be provided as the MN moves towards a candidate RDC.

4.8.1 Proactive Handoff transitions between neighbouring mobility links

At some point in time, the MN reaches some overlap area between the CURRENT RDC and one or more neighbour RDCs. The router adverts sent from the candidate RDCs (core MIPv6), over the all-nodes (link-local) multicast address, would provide sufficient prefix information to allow the MN to distinguish and enable one or more[27], candidate CoAs from the SCoA tuple. At this point there is no setup required by the MN or the candidate RDC.

The MN would only need to inform the CURRENT RDC (which is soon to transit to the PREVIOUS state) about the *highest priority candidate* (soft) CoAs by simply raising a flag within an internal **soft CoA Flag Bitmap (sCoA-FB)** kept within the MN for the SCoA tuple. That flag sets the respective CoAs in the SCoA tuple as **inactive secondary** with **highest priority candidate** designation.

In the proposed model a CoA may transit between the following states:

- **active primary**. This is a single CoA that is actively used for communications with CNs and HA when the PCoA is not activated or gets de-activated.
- **inactive secondary**. This is one or more CoA(s) that have been allocated for the particular MN over different RDCs. An inactive CoA must in turn transit between *lowest priority candidate* and *highest priority candidate* state before it transits to the active primary state. In particular:
  - **highest priority candidate**. This is one or more inactive CoA that has been determined to be the most likely candidates for activation. The main criterion for candidacy is currently the based on a single RDC advertisment detection[28] and mobility neighbourhood information surrounding that RDC advertisment.
  - **lowest priority candidate**. This is one or more inactive CoA(s) that are least likely candidates for activation. Optimizations may provide some probability-oriented ordering of these least likely candidates, particularly in the event that the probability of candidacy for the *highest priority candidate* CoA thresholds close to lowest priority candidate CoAs. This may be viewed as movement of the MN on the boundaries of more than one neighbour RDC links.

[27] at most 3 for a mobility neighbourhood of 6 cells

[28] Criteria for activation may comprize of link layer SNR stability metrics over some time period

These states effectively denote when a soft CoA is actually promoted towards an active primary CoA for the communications of the MN when the PCoA group is not activated or gets deactivated. This is the case at the end of a *stable* transition of the MN from the previous RDC to a new one.

As soon as the MN has set as *inactive secondary with highest priority* some CoA within its SCoA tuple, it generates a **PCoA-Enable (PCoA-E)** message that includes, the sCoA-FB and a lifetime refresh on both L_s, L_d. In this message L_s is set to 0 while L_d is set to -1 (infinity). The message includes also the SCoA tuple identifier from the original PRS-PSH message to associate the PCoA activation signal with the correct PCoA that is associated with the particular SCoA tuple; the later is also associated with the binding cache entry for the particular MMN. The PCoA-E message is then sent to the CURRENT RDC through a unicast packet. The receiver of the PCoA-E message, (i.e the CURRENT RDC) is expected to enable sending packets immediately to the MN through the PCoA multicast address, and stop sending unicast through its primary active CoA, since the PCoA start lifetime has value $L_s = 0$.

The traffic sent towards the MN is received transparently at the CURRENT RDC and then forwarded to the MN over its PCoA as **encapsulated multicast payload**. To effect such type of forwarding, when the PCoA-E messag has been received by the CURRENT RDC, packets arriving at the CURRENT RDC are forwarded to the PCoA group of the MN. This is done by matching the destination on-link CoA of the MN in each received packet with the respective *Binding Cache entry* at the CURRENT RDC which is now **extended** to hold also:

- the Roaming state allocated for the particular MN. This includes the SCoA tuple allocated to the MN and gets associated with the PCN cache of the CURRENT RDC to locate routing information about the neighbour RDCs.
- the LLA of the MN
- the PCoA mapping of the Roaming state for the particular MN, together with the PCoA start and stop lifetimes

Upon successfull match of the destination CoA, the received packet at the CURRENT RDC is encapsulated into a multicast packet with destination the PCoA group of the node. Since the neighbour RDCs have been configured to listen to the particular PCoA group and subsequently forward traffic for that group on their mobility-enabled links, traffic destined to the MN would be available to all the links of the mobility neighbourhood of the CURRENT RDC and subsequently of the MN. It should be reminded that the configuration of the hardware interface (LLA) multicast filter at the MN, does not depend on the unicast IP CoA allocated for the MN, but only on the PCoA[29]. That implies that the MN would be able to receive traffic on any of the reachable links of the

[29] its last four octets in particular

mobility neighbourhood as soon as link-layer connectivity is established with the particular mobility neighbouring link and the particular soft CoA has been enabled on its **multihomed** interface.

The proposed model introduces the notion of a **multicast encapsulation** for the purposes of delivering the traffic to all neighbouring RDC during activation of the PCoA group. Encapsulation over multicast encapsulates a normal unicast packet as the payload of a multicast packet, while it introduces a special flag (MT) called **multicast tunnel** placed as a destination option in the IPv6 header. On receipt of such packet, the MN must check the destination options of the packet whether the MT flag has been set; if the flag is set, the MN decapsulates the payload by removing both IP and UDP encapsulating headers from the multicast packet, without submitting the packet to the (UDP) transport layer. The decapsulated packet is then submited back to the IP stack for further processing. The packet would now have as destination address the on-link CoA at the CURRENT RDC which the MN must sustain as an active CoA until a new active CoA has been selected and the respective peer entities have informed through the standard Binding Update.

Traffic will be destined to the PCoA group for time L_d. Since L_d has been set to infinity the CURRENT RDC will continue to send traffic towards the MN through the PCoA, until instructed otherwise.

The rationale behind setting L_d to infinity and not to some finite time period is because it is highly likely that the MN does not perform what is termed as a **clean** handoff between two RDC and their underlying coverage areas. Instead, it is expected to bounce multiple times between coverage areas. Such bouncing effect is primarily related to the direction of the MN with respect to the geographical position of these coverage areas. This may be dictated by various factors such as direction-constrained movement, over roads, air, sea. Currently the road, air and sea path planning are completely uncorrelated [**?**] to the positioning of cells in cell positioning and planning. It is, thus, expected that movement of MNs is not bound to follow straight crossings between cells. This is illustrated in Figure 23.

It can be seen that within the overlap area some 3 different coverage areas (mapping to different RDCs) are visited during MN's transition over the overlap area and subsequent handoff. To avoid oscillatory signaling by the CURRENT RDC that enables or disables dispatch of traffic over the PCoA or an active CoA, the initial PCoA-E message sent, effect such PCoA start and stop timings that allow sending traffic to the MN over the PCoA group until explicitly requested to stop doing so.

The MN continues to receive traffic over the PCoA until it detaches from the CURRENT RDC plus some random time T_e that intialy varies between 250 and 500[30] ms. T_e is introduced for the purposes of ensuring reception of traffic through PCoA during cell bouncing effects in the MN's movement pattern between its CURRENT RDC and a neighbouring RDC; The time period T_e

[30] this figure needs further experimentation

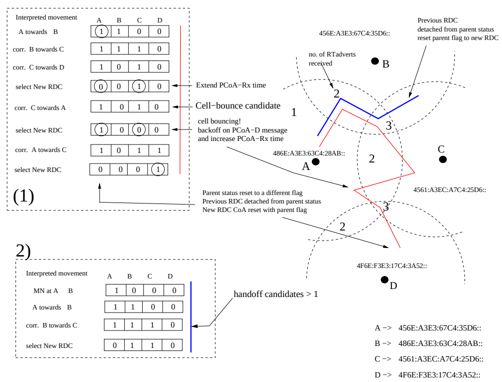

Figure 23: Bouncing effects incorporated in the movement pattern of the MN

is defined as **Extended PCoA-Rx Time**. The MN maintains also a **cell-bounce accumulator (CBA)** that tracks the number of bounces between the two RDCs. This accumulator geometrically increases T_e for each increment of it counter for the individual MN. The delay T_e is bounded by an upper delay maximum defined as `Max_Random_PCoA_Rx Stop_Delay` equal to 3000 ms[31]

When the extended PCoA-Rx time elapses and the cell-bounce accumulator has not increased, the MN proceeds to send a standard Binding Update to the CNs and the HA to inform about the new, stable and topologically correct CoA as activated by the MN (see Figure 24). At the same time the MN sends a **PCoA-Disable (PCoA-D)** message to the PREVIOUS RDC, through its new active CoA (i.e the new CURRENT RDC) to inform about the deactivation of the PCoA in terms of forwarding traffic towards the MN. The PCoA-D message contains the specific PCoA group and updated lifetime values for L_s and L_d which are set to -1 and 0 respectively.

The PCoA-D message instructs further the PREVIOUS RDC to manage group membership of the PCoA group. This mainly manifests itself as a need to 'prune' some of the neighbour RDC 'listeners'[32] that do not match with the current view of the mobility neighbourhood of the new CURRENT RDC. The PCoA-D message includes also the SCoA flag bitmap which signifies which is the new active primary CoA.

4.8.2 SCoA tracking and Cell-Bounce Accumulator

The cell-bounce accumulator operates on the SCoA tuple represented by the SCoA flag bitmap and is maintained at the MN. Such bitmap includes a flag for each soft CoA, including the primary active one; it is not expected to increase

[31] initial and max value should be conditioned by the overlap area between two cells as well as the speed of the mobile

[32] PCoA group forwarders

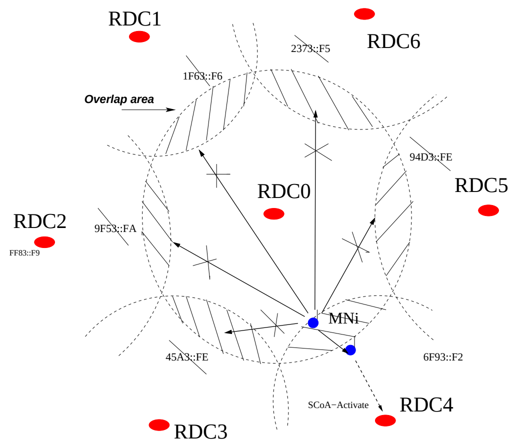

Figure 24: MN activates the correct soft CoA

more than 7 bits for the entire mobility neighbourhood, although it is not limited by such figure. In this bitmap, the accumulator tracks the CURRENT RDC by means of its CoA flag. This is designated a **parent** status and takes a value of 1 at the CURRENT RDC. As long as the parent flag does not flip to 0[33], then the MN is expected to stay associated with its CURRENT RDC even if it is bordering at the cell perimeter. The parent status is not reset to a different flag within the bitmap until the MN signals a new active primary CoA to its peers (HA + CNs).

Receipt of standard router advertisment flip the relevant flag within the bitmap. Thus, upon movement of the MN to an overlap area between two coverage areas, the parent flag is flipped to 0; then the CBA seeks for a *single* CoA flag that has flipped to 1 within the bitmap while the total number of flags marked with 1 stays the same. In other words the CBA scans for an equilateral value change between only two bits within the bitmap. Such change in the bitmap signifies a clean handoff from the previous RDC. Because there is no way to distinguish between a clean handoff and an imminent cell bounce, the reception at the PCoA is sustained for the period defined as Extended PCoA-Rx Time. This would prevent oscillatory signaling from the MN to its peer entities about switching between the PCoA group and some active CoA for the dispatch of packets towards the MN.

If a cell-bounce is, indeed, imminent the CBA will observe the parent flag flipped back to 1, since the parent status must **not** be reset to a new flag until a new CoA has been signaled (as active primary) to the peer entities by the MN. This, however, signifies cell-bouncing candidacy, since the MN is still within the boundaries of the candidate primary CoA activation. It is for this reason, that the cell-bounce accumulator does not consider the movement **as yet** a cell-

[33] due to lack in reception of the respective RDC router advert by the MN

bounce. Thus, the CBA does not yet affect the timer as it should not yet reset so as to backoff from signifying to the peers the new active CoA.

As it may be seen by Figure 23, the parent flag would be flipped back to 1, when the MN would receive again a router advertisment from the PREVIOUS RDC. The CBA identifies a cell-bounce candidate but does not yet affect the PCoA-Rx timer since the MN is still within the cell of the CoA that is set to highest priority candidate and its respective RDC. As soon as it detaches from the latter a cell-bounce has been established. At this point the CBA, at the MN, triggers a backoff for the dispatch of the PCoA-D signal by geometrically increasing the PCoA-Rx timer value and continues tracking parent status, until the timer expires while the SCoA flag bitmap stays unaltered.

On expiry of the PCoA-Rx time, while no CBA backoff has been triggered, the PCoA-D message is sent towards the PREVIOUS RDC (or to the RDC that still remains in the CURRENT state) and the parent status is either reset to a new CoA or stays the same flag in the bitmap; that implies that either the new active primary CoA has been identified with the MN moving to a new coverage area or the old CoA is still in effect and movement still effect the same coverage area. In the event of the former, the MN:

- first configures its link-local address that is valid over the new link.
- sends a neighbour advertisment which removes the P flag from the neighbour cache of the RDC.
- send a Binding Update (BU) to the NEW RDC
- informs its peers about the new CoA with a standard BU, while the CBA is reset to 0 and reinitiates tracking for a new parent (new CURRENT RDC).

It has been previously noted that a clean handoff, even on the verge of an imminent cell-bounce is not always the case. It is possible for instance that a maximum or 3 RDC router advertisments can be heard by the MN before it gets detached from its CURRENT RDC. That implies that there exist *more than one* alternative handoff candidates. This is manifested over the flag bitmap, as shown in part 2 of 23, by means of having two CoA flags flipped to 1 when the parent has been flipped to 0 (i.e. got detached from CURRENT RDC). In that case the MN must select one of the two CoAs as the one to be promoted from first as *highest priority candidate* and then to the *active primary* state. The selection is bound to be probabilistic.

Our model opts for two different selection criteria:

- maximally traversed cell/RDC selection. This implies that, the MN would select to activate that CoA that maps to the cell that has been traversed most. This can be determined by the flag bitmap at the IP layer although it is coarse[34].

[34] however if the neighbouring cell pattern contains more RDC neighbours, like 8 or 10 then traversal determination improves significantly at the IP layer

- minimally traversed cell/RDC selection. The cell/RDC that has not been traversed much will be selected through the respective CoA activation.

We should note that the above selection criteria are serving more the MN rather than the network. For instance it is possible that the network needs to select that RDC which is less loaded. We argue, however, that such selection benefit is bound to be negated if it opposes to the movement pattern of the MN. That is to say, that if the MN needs to move towards the cell that serves its intents better, then the load balancing decision would incur simply more handoffs between the involved cells/RDCs.

4.8.3 Refreshing the soft CoA tuple

The proactive facilitation of the PCoA group for the traffic forwarding towards the MN allows the latter to move around the mobility neighbouhood of the CURRENT RDC with much greater freedom and without the potential of packet loss during cell bounce effects during its movement pattern. To sustain such freedom of movement the MN must maintain a **fresh SCoA tuple** that is valid over the *new* mobility neighbourhood of the new CURRENT RDC.

A straightforward approach would be to drop all allocated soft CoAs except for the active CoA that is valide at the NEW RDC. The NEW RDC would then need to initiate the process of soft CoA allocation according to the model already described. That, however, would require that the NEW RDC invalidates even these soft CoAs that would be common with the routing neighbourhood of the PREVIOUS RDC.

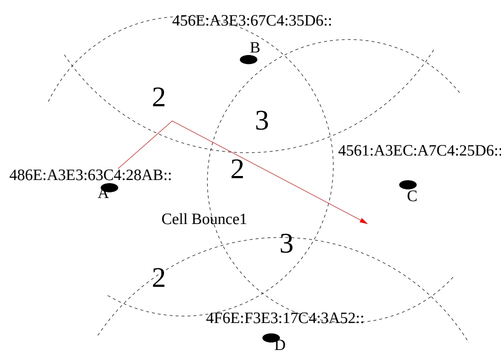

Figure 25: Router Advertisments and link detection at overlap areas

Consider for instance the example of Figure 25. During its transition path to a new RDC the MN is expected to establish receipt of multiple RDC router advertisments over some overlap area between 2 or 3 cells. In the event that the CURRENT RDC (C) opts for invalidating the entire soft CoA tuple (except for its own active CoA), it would essentially invalidate also **re-usable** soft CoA of the common RDC neighbours between A and C. That would require that:

- the NEW RDC informs the PREVIOUS RDC with an **SCoAT invalidation request** for the entire soft CoA tuple, while it excludes the originator of the request.
- it initiates a new SCoA tuple generation from scratch. That would include also the common neighbours of (A) and (C).

The above, while a cleaner solution, would effectively incur more latency in the generation of valid soft CoAs in the newly generated tuple. This is so since, clearly, the NEW RDC could reuse the common soft CoAs between the two neighbouring RDCs as the former transits to the CURRENT state.

Alternatively, the new CURRENT RDC need only provide the MN with a specific **soft CoA tuple update (SCoAT-Update)** that would include only the *new* neighbouring soft CoAs that are valid in the neighbourhood of the new RDC, together with the flag bitmap that invalidates soft CoAs that are redundant in the mobility neighbourhood[35] of that CURRENT RDC. Furthermore, the PREVIOUS RDC should receive invalidation information about the redundant soft CoA RDC neighbours.

We should note that the new CURRENT RDC is aware of the LLA of the MN since it created a soft CoA for that MN in the past and as such it has stored it in some secondary store[36].

Figure 26 illustrates the idea. In a 6-cell mobility neighbourhood the first SCoAT allocation would require six (6) new soft CoAs. At the next handoff the soft CoA tuple need only be refreshed with another 3 soft CoAs. This is half the number of soft CoAs originally allocated.

For the above approach to work efficiently and without extraneous message overhead it is essential to see how the PREVIOUS RDC resolves the identity of the true redundant RDCs and their respective soft CoA.

4.8.4 Resolution of redundant SCoA RDC neighbours

In the proposed model, an SCoAT-Update requires that the PREVIOUS RDC resolves accurately the RDCs that are *truly* redundant. We emphasize on 'truly', because, the cell-bouncing effected by the mobility pattern of the MN is likely to cause an inaccurate resolution of reduntant SCoAs at the PREVIOUS RDC, when the MN 'settles' to a new active primary CoA at some NEW RDC. This of course is higly dependent on the mobility pattern of the MN and the correlation of its movement[37] to the position of the neighbouring cell RDCs.

[35]but valid in the PREVIOUS RDC

[36]cache would probably be an overkill since that RDC does not know when it will needed and it is not bound to be used soon before the MN hands off to that RDC.

[37]for instance road reconing

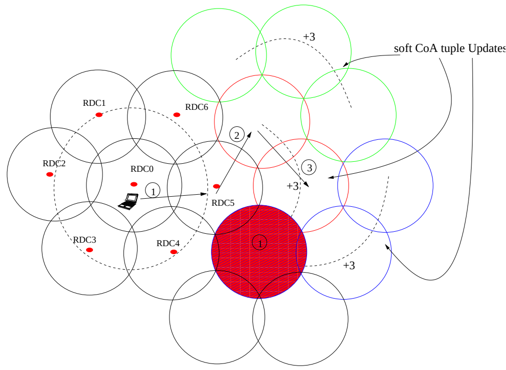

Figure 26: CoA resuse within the SCoA tuple during continuous movement of the MN

We argue that proactive resolution of the ultimate reduntant SCoAs **before the MN 'settles' with a single active primary CoA** (from the allocated SCoA tuple) is bound to result in significantly extraneous signaling that may be rendered invalid by the movement pattern of the MN. An MN is considered to have **settled** within a coverage area, as soon as it has sent a Binding update towards its peers informing about the new primary active CoA.

The Binding Update sent by the MN towards the NEW RDC at the start of the settlement period, provides the NEW RDC with the address of the PREVIOUS RDC. The BU also encompasses a **SCoAT invalidation request (SCoAT-INReq)** flag. By setting such flag, the address of the PREVIOUS RDC provided through the BU to the NEW RDC, triggers the latter to determine the soft CoAs for the neighbouring RDCs that are **common** to both NEW and PREVIOUS RDCs, as indicated by the contents of the NEW RDC's PCN-Cache. This is illustrated in Figures 28 and 27. By determining the common RDCs between the mobility neighbourhood of the NEW and the PREVIOUS RDC, the two entities can proceed to *include* new RDC neighbours and exclude redundant RDC neighbours respectively as follows:

$$
\begin{aligned}
Common &= \bigcap(MN_{14}, MN_8) \qquad (11)\\
Exclude &= MN_{14} - \bigcap(MN_{14}, MN_8)\\
Include &= MN_8 - \bigcap(MN_{14}, MN_8)
\end{aligned}
$$

Where MN_x is the mobility neighbourhood of RDC x maintained within the PCN Cache of each RDC. RDC 14 represents the **PREVIOUS** RDC while RDC 8 represents the **NEW** one. While the common RDC neighbours are determined at both the PREVIOUS and NEW RDC, the **exclude** operation is scheduled at the PREVIOUS RDC where redundant soft CoA are removed from the SCoA tuple. This also implies that the respective RDCs must be removed from listening at the PCoA group. On the contrary, the **include** operation is scheduled at the NEW RDC, where new valid soft CoAs need to be generated and added to the SCoA tuple of the MN. This further implies that the underlying PCN-Cache entries must be amended, while RDC neighbours must be added as listeners to the PCoA accordingly.

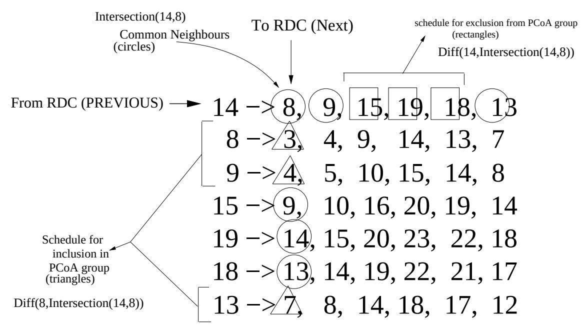

Figure 27: Resolving the RDC participants that need to be included and excluded from the PCoA group

On receipt of the BU with the SCoAT invalidation request flag set, the NEW RDC, generates a similar message which is sent to the PREVIOUS RDC such that the latter can proceed with release of the redundant soft CoA. This is achieved by either suppressing the sending of I-MLD Reports to the redundant RDCs or by explicitly sending an I-MLD Done message to each of them. In case of the former, each redundant RDC neighbour expires from its group membership list the particular PCoA address entry while it removes the neighbour cache entries that pertain to the associated MN after receiving an **indirect neighbour advertisment (I-NA)**. In the case where the I-MLD Done message is sent to the redundant RDC, the PCoA to be removed from the membership list is specified. Similarly an indirect neighbour advertisment invalidates the redundant neighbour cache entry at that RDC.

The inclusion of the NEW RDC neighbours in the PCN-Cache, triggers the generation of a subset of new soft CoA for the SCoA tuple associated with the MN. In this case the new PRS-PSH generated at the CURRENT RDC sets an R flag; this denotes a **refresh** of the SCoA tuple and its suplemental RDC information.

4.8.5 Managing the I-MLD Done at the PREVIOUS RDC

Having identified the RDCs for exclusion at the PREVIOUS RDC, the latter should issue a I-MLD Done unicast message to the specific RDCs that should 'leave' the PCoA group, as shown in Figure 28. The I-MLD Done contains the

PCoA group for which the RDC must stop forwarding of traffic.

It should be noted that the particular management of 'leaves' on the PREVIOUS RDC need not necessarily be manifested through explicit I-MLD Done messages towards redundant RDCs. It may suppress any indirect messaging towards these such that the group membership list entries expire and get discarded.

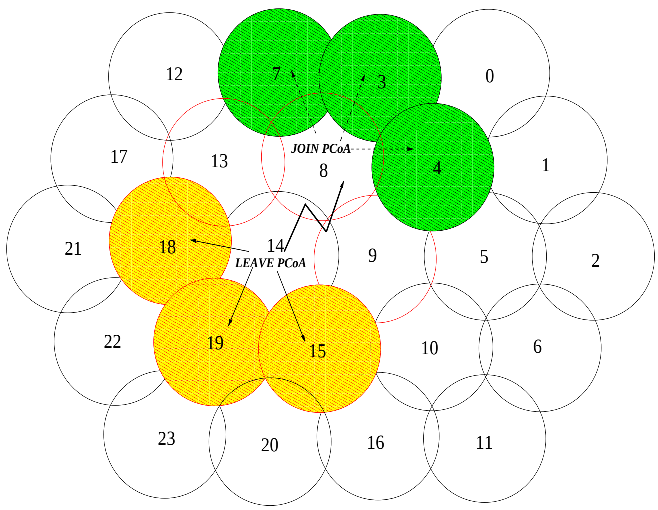

Figure 28: Sustaining accurate mapping of RDC neighbours on MN's PCoA group routing identifier

4.8.6 Managing the PCoA 'joins' at the PREVIOUS RDC

In a much similar fashion the NEW RDC can resolve the new RDCs that need to be included in the PCoA group. This resolution triggers the issue of a new indirect join solicitation message which initiates the process of indirect multicast listener configuration. The message is sent in a unicast packet on a per-included-RDC basis.

Past the inclusion of the NEW RDC neighbours in the PCoA group the CURRENT RDC either constructs the relevant soft CoAs from the respective new RDC prefixes or its distributes the task of soft CoA generation to the included new RDCs as described in previous sections. Part of the soft CoA generation process, i.e. the DAD check, ensures that the new included RDC also amends appropriately its neighbour cache entry with the LLA of the MN, the P flag while it sets the neighbour cache entry to the PROACTIVELY REACHABLE state. We remind that both the P flag and the aforementioned state are removed only at the new CURRENT RDC for which the MN has settled its active primary CoA.

The newly created soft CoAs are pushed as part of the roaming context for the MN, in a PRS-PSH message towards the MN with the SCoAT-Update flag (R) set, to signify that this is only an update to the existing SCoA tuple. This message carries also a **SCoA 'exclude' address bitmap** that signifies to the MN which soft CoAs it needs to discard from its *SCoA cache*.

4.8.7 HA and CN considerations

The proactive mobility model will be transparent to the operation of both the HA and the CNs. Standard binding updates will be sent to the peer HA and CN entities to inform about the new active CoA.

4.8.8 Continuous vs disrupted connectivity during MN movement

It is possible that the MN does not maintain a continuous movement pattern over a set of adjacent (contiguous) adminstrative domains. For instance, the MN travels under such conditions that requires it to turn off its network interface and as such disrupt its connectivity. Typical such examples is movement through road tunnels, by means of aircraft or sea carriers or through underground train stations.

It also possible that the MN encounters black holes of network connectivity in the radio access network coverage area over some specific geographic location.

Under such circumstances the MN needs to maintain persistent state about a minimal set of information. These are:

- its home network HA IP address.
- its allocated PCoA.
- the last CURRENT RDC which upon turning on of the interface or bootstrap it checks whether it is still valid; otherwise the entry transits immediately to the PREVIOUS state.

Depending on whether the MN switches off completely or just its network interface as well as whether is has received any RDC advertisment within DISRUPTED_CONNECTIVITY_TIME, the MN will need to engage in different reconnection approaches. In particular,

- if the MN is switched off completely, and assuming that the visiting RDC in which it will bootstrap is proactive-mobility-aware, the MN will need to obtain on-link IP connectivity before it can contact its HA and CNs according to the IPv6 base mobility rules. The MN must inform its last PREVIOUS RDC about the new CURRENT RDC so that the former can request its neighbouring mobility RDCs to leave the PCoA for that MN. Upon receipt of such message the PREVIOUS RDC may (or may not) send a I-MLD Done since the Querier will expire the PCoA membership on the particular interface should it has not received an I-MLD Report update for some time. If the NEW RDC IP address is not in the PCN-Cache of the PREVIOUS RDC, the latter invalidates the the entire old soft CoA tuple and propagates such invalidation to the respective RDC neighbours in terms of PCoA group membership and release of the allocated SCoAs through I-NA messages. This is because the neighbours of the PREVIOUS and the new CURRENT RDC are completely different.

Thus, a new SCoA tuple is being generated. PCoA group membership is then reinitialised at the new CURRENT RDC from scratch, pending an optional acknowledgement from the PREVIOUS RDC to confirm invalidation of PCoA group membership and SCoAT bindings for the MN at its RDC neighbours.

- the MN has turned off its interface but is not completely switched off. Here the MN needs to track the receipt of the last RDC advertisment or potential paging message[38] for time T_d. This time is defined as the ratio of the cell's diameter C_d over the average movement of the MN per second[39] S_{MN}, times a **cell boundary adjustment coefficient** C_b, currently set to 1.5. That is:

$$T_d = \frac{C_d}{S_{MN}} C_b \tag{12}$$

If no new RDC advertisment is received within time T_d then the MN is considered switched off from the perspective of the network and thus the first case, described above, is followed. The above time allows the MN to remain disconnected for the entire transition of the MN from the CURRENT RDC to a new one without reseting the proactive mobility process. Beyond that any connectivity of the MN with HA and CNs is considered as reset and, thus, must be re-established.

5 Extended model optimizations

In this section we elaborate on further optimizations that would reduce the cycle of interactions required during the proactive setup stage of the aforementioned roaming context. However, the following optimizations are beyond the scope of this document and as such any further analysis and inclusion to the proposed protocol is the object of future work.

5.1 Coupling of MLD proxying and CoA allocation

It should be noted that potential optimizations may be achieved if the indirect join solicitation signal is responded with an indirect MLD Query that contains

[38] for PCS systems

[39] derived for some average speed metrics that are specific for different kinds of MNs (vehicular)

a freshly generated and DAD-checked IPv6 soft CoA. This is because MLD membership does not depend on the specifics of an IP address; it only depends on the existence of a multicast listener on-link at that interface.

In this fashion, the CURRENT RDC combines signaling for I-MLD and the soft CoA generation process. Furthermore, the soft CoA generation is purely distributed at the relevant constituencies[40].

5.2 Pessimistic and Optimistic Proactivity

As currently specified in the proposed model, the intermediate signaling for PCoA group membership interest follows a **pessimistic** approach and declares indirectly receiver interest at *all* immediate neighbouring RDC link interfaces. That indirectly implies that all soft CoA could receive traffic at the PCoA since the particular RDC link would be configured to forward packets destined to the particular PCoA group of the MN. Such conservative approach is primarily due to the possibility of harsh cell bouncing effects manifested during the mobility pattern of the MN. We have argued that since there is no insight in frequency of cell bouncing as the MN moves over some terrain in a direction-constrained fashion[41] the scheme pursues the sound approach of including all RDC neighbours.

It is possible, however, that the scheme could adopt a more **optimistic** approach over RDC membership interest signalling if the model has more information about the position of the MN (and subsequently its direction) in relation to the position of neighbouring cells that are controlled by different RDCs. This should allow the CURRENT RDC to declare PCoA membership interest only for those RDC neighbours that are more probable to serve the MN in the immediate future. The internals of such optimistic approach are beyond the scope of this version of the document.

5.3 Context Transfer over multicast channels

Instead of pushing context state over ICMPv6 message between neighbour RDCs or between the CURRENT RDC and the MN, each ontology of context state may be channelised over a different well-known **Proactive context transfer group (PCT-G)**. This requires that all RDC neighbours of the CURRENT RDC are listening on a different PCT group that is specified as well-known for the relevant context ontology. Management of each group is done also indirectly as specified in previous sections of this document. To restrict, however, receipt of context state for some ontology, over to the routing neighbourhood mapping to the mobility neighbourhood surrounding the CURRENT RDC, the Hop Limit must be restricted to 7, for a hexagonal coverage area shape model. Assymetry in the coverage range of adjacent access points may, however, introduce extra factors that need be considered in the adjacency vector effected in the routing neighbourhood of the CURRENT RDC. This is a topic for further study, and

[40] i.e. only the RDC over which the generated CoA is activatable produces it
[41] for instance road traffic

as such any further analysis for introducing the PCT-G alternative is out of the scope of this version of the document.

6 Description of the messages

In this sections the control signaling that effects the specified proactive mobility scheme is also specified. These are currently ICMPv6 messages. All messages defined maintain a minimum set of 3 fields, namely Type, Code and Checksum. The Type and Code values differ for each ICMP message while the Checksum field takes values as specified in [6].

6.1 Indirect RNV Update

This message is used during incremental RNV acquisition through reactive learning that is effected temporaly during the transition process of an MN.

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Pref len pRDC |L|                Reserved                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Sequence Number           |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+       Previous RDC LLA        |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          Previous RDC                         |
|                     MNV-enabled IPv6 address                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 29: Indirect RNV Update

- Source Address : The globaly routable IPv6 address of the MN.
- Destination Address : The globally routable IPv6 address of the NEW RDC currently transiting to the CURRENT state.
- Hop Limit: 1
- Type: 50 - It identifies indirect RNV information
- Code: 0 - The message type identified by Type is of class Update

- Reserved: Must be initialized to zero.
- Sequence Number: monotonically increasing 16-bit integer which must be return on the respective Acknowledgement
- L: flag to indicate that the LLA of the Previous RDC is also included in the message

6.2 Direct RNV Update

Depending on the method employed for RNV acquisition for the purposes of discovering the RNV that maps to the mobility neighbourhood of the CURRENT RDC, a Direct RNV Update is sent accordingly from the CURRENT RDC to the PREVIOUS one.

If incremental RNV aquisition through temporal reactive learning is effected then a D-RNV Update is sent by the CURRENT RDC to the PREVIOUS on receipt of an I-RNV Update sent by a MN.

If Complete RNV aquisition with fully proactive learning the D-RNV Update is sent periodically from the CURRENT RDC to all of its upstream interfaces, excluding the MNV-enabled interface. In this case, the destination address changes to the 'all-router' multicast address

This message is sent only between RDC. Each Access Point is assumed to have been configured with its Coverage Area tuple defined in previous sections of this document.

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Sequence Number         |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+          CURRENT RDC LLA      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          CURRENT RDC                          |
|                     MNV-enabled IPv6 address                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RDC-specific context/capability option(s)....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 30: Direct RNV Update

- Source Address: It depends on the RNV acquisition scheme employed. For:
  - partial reactive learning (PRL): the global IPv6 address of the CURRENT RDC that receives an I-RNV Update message from some MN.
  - fully proactive learning (FPL): the link-local IPv6 address of the CURRENT RDC.
- Destination Address: It depends on the RNV acquisition scheme employed. For:

  - PRL: the global IPv6 address of the PREVIOUS RDC as provided by the I-RNV Update message sent some MN
  - FPL: the all-router multicast group.

- Hop Limit: It depends on the RNV acquisition scheme employed. For:
  - PRL: 1
  - FPL: 255
- Type: 51 - identifies an Update message type
- Code: identifies a bitmap that encodes one or more RDC-specific context capability options which identify the originating RDC in terms of itself.
  - bit 0 (LSB) - identifies Roaming context information
  - bit 1 - identifies compression context information
  - bit 2 - identifies transport context information
  - bit 3 - identifies security context information
  - bit 4 - identifies session context information
- Sequence Number: monotonically increasing 16-bit integer which must be return on the respective Acknowledgement
- RDC-specific context/capability option(s): contains one or more RDC-specific context option that identify valid context capabilities on the originating RDC.

6.3 Roaming context option

This option identifies information relevant to the resolution the Routing neighbourhood vector of the originating RDC. It specifically provides the Coverage Area tuple which identifies the Access Point of an RDC.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Distance    |           Reserved            | Pref len RDC  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Latitude of sending RDC's AP lat                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Longitude of sending RDC's AP                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Radius of sending RDC's AP                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Height of sending RDC's AP                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 31: Roaming context option

- Distance: 7 - the number of hops that this signal is allowed to be forwarded before it gets discarded. The value it takes represents the nominal number of neighbours. For hexagonal-shaped cells this is assumed to be six.
- Pref len RDC: the prefix length of the CURRENT RDC IPv6 included in the message.
- AP lat: The latitude of the Access Point of the RDC message originator
- AP Lon: The Longitude of the Access Point of the RDC message originator
- AP Rad: The radius (range) of the Access Point of the RDC message originator.
- AP Hei: The height of the Access Point of the RDC message originator

6.4 PCN Advertisment

When the RNV neighbourhood of the CURRENT RDC has been established, this message advertises RDC-specific roaming context through a unicast message to each of the neighbour RDCs. In this manner, each RDC becomes aware implicitly of the mobility and routing neighbourhood of its neighbours.

The PCN-Advertisment includes one or more context ontology options. A context ontology option may be populated with RDC-specific context for the following context ontologies:

- roaming
- compression (header or other)
- transport (tcp, udp or other)
- security
- session (SIP, SAP, SDP or other)
- other

The ontologies identify context capabilities that are enabled on an individual RDC with respect to its neighbours. This message is exchanged only between RDCs.

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Sequence Number        |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Context ontology/(ies) option ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 32: PCN Advertisment

- Source Address: the global IPv6 address of the originating RDC.
- Destination Address: the global IPv6 address of the RDC discovered as its neighbour RDC from its cached RNV vector.
- Hop Limit: 255
- Type: 70 - It identifies a Context Neighbour Discovery message
- Code: identifies a bitmap that encodes one or more context state ontology options specific to the RDC. Thes context state ontology options identify not only the originating RDC but also its neighbours.

  - bit 0 (LSB) - identifies the roaming context ontology
  - bit 1 - identifies the compression context ontology
  - bit 2 - identifies the transport context ontology
  - bit 3 - identifies the security context ontology
  - bit 4 - identifies the session context ontology, like SIP, SAP, etc.

- Sequence Number: monotonically increasing 16-bit integer which must be return on the respective Acknowledgement.

6.4.1 Roaming Context Ontology option

This is the option included in the PCN advertisment when Roaming context needs to be advertised between neighbouring RDC.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|RDC Neigh No.| Pref Len RDC1 | Pref Len RDC2 | ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...                                           | Pref Len RDCn |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                    Neighbour1 RDC IPv6 addr                   |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Neighbour2 RDC IPv6 addr                   |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    NeighbourN RDC IPv6 addr                   |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 33: Roaming Context Ontology option

- RDC Neigh No.: number of RDC neighbours included
- Pref Len RDC1..n : the prefix length for each RDC IPv6 address of the interface that belongs to the MNV of the RDC.
- Neighbour1 RDC IPv6 addr: the globally routable IPv6 address of the RDC neighbour.

6.5 Aggregate PI-DAD Request

As described previously the RDC may employ one two methods for Duplicat Address detection:

- Agggregate Proactive Indirect DAD check
- Plain Proactive Indirect DAD check

This is the message used for an aggregate proactive indirect DAD check. It initiates a request to the neighbour RDC for which the IPv6 addresses created by the CURRENT RDC are valid. The RDC maintains a record of the LLA of the MN for which it produces an IPv6 soft CoA.

- Source Address: the global IPv6 address of the originating RDC
- Destination Address: the global IPv6 address of the RDC neighbour for which the soft CoA generated by the originating RDC, are topologically correct.
- Hop Limit: 255
- Type: 75 - identifies aggregate PI-DAD check
- Code: 0 - identifies a request message for the above type
- SCoA1..n : the soft CoA generated by the originating RDC for some MN, which is topologically correct at the receiving RDC neighbour. The ordering of SCoAs in the request message is important for the purposes of the respective reply.
- MN1..n LLA: the link layer address of the Mobile Node for which the address has been generated. The LLA of the MN is to be used for ND purposes (proactive neighbour caching)
- Aggregate PI-DAD Id : an identifier to map the awaited aggregate PI-DAD reply to the correct set of SCoAs generated.

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Aggregate PI-DAD Id        |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                            SCoA1                              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                            SCoA2                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                            SCoAn                              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|           MN1 LLA             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+        MN2 LLA                |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

continued next page...

..continued from previous page

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|           MNn-1 LLA           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+       MNn LLA                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 34: Aggregate PI-DAD Request

6.6 Aggregate PI-DAD Reply

This message is sent in response to an aggregate PI-DAD check request.

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Aggregate PI-DAD Id       |   Aggregate Address Bitmap    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 35: Aggregate PI-DAD Reply

- Source Address: the global IPv6 address of the RDC neighbour that received the original aggregate PI-DAD request.
- Destination Address: the global IPv6 address of the RDC that originated the aggregate PI-DAD request.
- Hop Limit: 255
- Type: 75 - identifies aggregate PI-DAD check
- Code: 1 - identifies a reply message for the above type
- Aggregate PI-DAD Id: The identifier contained in the original aggregate PI-DAD request message. This identifies will map
- Aggregate Address Bitmap: an address bitmap that identifies the addresses for which the the DAD check was successfull. It is assumed that the originator of the aggregate PI-DAD request maintains ordering information from the original DAD request to resolve this aggregate address bitmap.

6.6.1 Plain PI-DAD Request/Reply

In the case of plain PI-DAD check the originating RDC simply provides the RDC neighbour with the SCoA that must be checked for duplicates and the LLA of the MN. The source and destination address follow the rules provided for an aggregate PI-DAD check message.

Thus the message for plain PI-DAD reduces to the following:

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          PI-DAD Id            |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+       MN  LLA                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                             SCoA                              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 36: Plain PI-DAD Request message

- Type: 76 - identifies PI-DAD check
- Code: 0 - identifies a request message
- PI-DAD Id: identifier to be used in the reply for the PI-DAD check effected at the neighbour RDC for a particular node.
- MN LLA : the link layer address of the MN accomodated.
- SCoA: the soft Care-of Address generated for that MN.

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         PI-DAD Id             |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 37: Plain PI-DAD Reply message

- Type: 76 - identifies aggregate PI-DAD check
- Code: 1 - identifies a reply message for the above type
- PI-DAD Id:
  identifier to be used in the reply for the PI-DAD check request originated.

6.7 Indirect SCoA-Create message

This message is used in the event that address generation is distributed to the RDC neighbours. In this case, the originating RDC does not generate the soft CoA itself, but distributes the task to the RDC neighbours.

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|           MN LLA              +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |       Reserved                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 38: SCoA-Create message

- Source Address: the global IPv6 address of the CURRENT RDC that accomodates the MN.
- Destination Address: the global IPv6 address of the RDC neighbour which is requested to create a soft CoA.
- Hop Limit: 255
- Type: 77 - identifies an indirect SCoA create message type
- Code: 0 - identifies a create request
- MN LLA: the link layer address of the MN for which a soft CoA is to be generated by the RDC neighbour.

6.8 SCoA-Ready message

This message comprizes the response from the RDC neighbour on the SCoA-Create request. The response message simply includes the soft CoA generated indirectly for the particular MN.

```
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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                             SCoA                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 39: SCoA-Ready message

- Source Address: the global IPv6 address of the RDC neighbour that generated the soft CoA.
- Destination Address: the global IPv6 address of the RDC which originated the SCoA-Create request.
- Hop Limit: 255
- Type: 77 - identifies an indirect SCoA create message type
- Code: 1 - identifies a reply to an SCoA-Create request
- SCoA: the soft CoA generated by the neighbour RDC.

6.9 Proactive Roaming State Push

When MN-specific roaming context has been generated by the CURRENT RDC it must be pushed to the MN. This is achieved by means of a Proactive Context State Push message which incorporates a Roaming State option. The Proactive Context State Push can take one or more different context state options for the purposes of injecting different context class state to the MN for the purposes of seamless mobility effected over proactive context transfer.

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Sequence Number           |         Reserved              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Context state options..
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 40: Proactive Context State Push (PRS-PSH)

- Source Address: the global IPv6 address of the CURRENT RDC that is accomodating the MN.
- Destination Address: the current active primary IPv6 address of the MN as effected at the CURRENT RDC.
- Hop Limit: 255
- Type: 90 - It identifies a Proactive Context push message
- Code: identifies a bitmap that encodes one or more context state options that are pushed towards the MN
  - bit 0 (LSB) - identifies the roaming state class for the MN
  - bit 1 - identifies the compression state class for the MN
  - bit 2 - identifies the transport state class for the MN
  - bit 3 - identifies the security state class for the MN
  - bit 4 - identifies the session state class for the MN
- Sequence Number: monotonically increasing 16-bit integer which must be return on the respective Acknowledgement.
- Context State push options: contains one or more context state options that are specific to a single MN and are to be pushed to that MN.

6.9.1 Roaming State option

- RDC Neigh No.: an 8-bit integer which identifies the number of RDC neighbours included.
- SCoA tuple identifier: a 32-bit (or 16-bit) number that uniquely identifies the soft CoA tuple that is being pushed to the MN.
- M : PCoA is included
- I : incomplete proactive roaming state provided
- J : Explicit Join Solicitation flag
- R : SCoAT-Update flag
- E : flag that signifies that eh SCoA 'exclude' address bitmap is present
- Pref Len RDC1..n: one or more 8-bit number that identifies the prefix length of the RDC neighbour IPv6 address that is included in the Roaming state option. The number of prefix-length numbers present depends on the number of RDC neighbours stated in the push message option.
- Neighbour1..n RDC IPv6 addr: one or more global IPv6 address of the RDC neighbours of the CURRENT RDC.
- SCoA1..n: one or more generated soft CoAs that identify the mobility and routing neighbourhood of the CURRENT RDC.
- PCoA: The Proactive CoA multicast group that is mapped to the soft CoA tuple by the CURRENT RDC.
- SCoA 'exclude' address bitmap: an address bitmap that identifies the soft CoA that must be excluded as redundant from the active soft CoA tuple as maintained at the MN. The size of this address bitmap is expected to be 8-bits.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|RDC Neigh No.  |M|I|J|R|E|           Reserved                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      SCoA tuple identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Pref Len RDC1 | Pref Len RDC2 | ...           | Pref Len RDCn |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Neighbour1 RDC IPv6 addr                   |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Neighbour2 RDC IPv6 addr                   |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    NeighbourN RDC IPv6 addr                   |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                            SCoA1                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

continued next page...

..continued from previous page

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                             SCoA2                             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                             SCoAn                             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                             PCoA                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SCoA 'exclude' address bitmap                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 41: Roaming State Ontology Push option

6.10 Proactive Roaming State Reply

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|RDC Neigh No.  |        Sequence Number        |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 42: Proactive Roaming State Reply

- Source Address: the global IPv6 address of the MN that received the PRS-PSH message.
- Destination Address: the global IPv6 address of the CURRENT RDC that effected the PRS-PSH message towards the MN.
- Hop Limit: 255

- Type: 91 - It identifies a Proactive Context reply message
- Code: identifies a bitmap that encodes one or more context state options that are pushed towards the MN
  - bit 0 (LSB) - identifies the roaming state received by the MN
  - bit 1 - identifies the compression state received the MN
  - bit 2 - identifies the transport state received by the MN
  - bit 3 - identifies the security state received by the MN
  - bit 4 - identifies the session state received by the MN
- RDC Neigh No.: an 8-bit integer which identifies the number of RDC neighbours received by the MN.
- Sequence Number: a 16-bit integer which was originaly sent by the CURRENT through the PRS-PSH message.

6.11 Indirect MLD Messaging

6.11.1 Implicit Join Solicitation

1

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     PCoA multicast address                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 43: Implicit Join Solicitation

- Source Address: the global IPv6 address of the CURRENT RDC where the MN is accomodated.
- Destination Address: the global IPv6 address of the RDC neihgbour which is required to maintain group membership for the provided PCoA group.
- Hop Limit: 255
- Type: 101 - It identifies a implicit join solicitation message
- Code: 0
- PCoA: A Proactive CoA multicast group that is has been mapped to some soft CoA tuple by the CURRENT RDC for some MN.

6.11.2 Indirect MLD Membership Query

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Max Response Delay        |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     PCoA multicast address                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 44: Indirect MLD Membership Query

- Source Address: the global IPv6 address of the RDC neighbour that originaly received an implicit join solicitation message.
- Destination Address: the global IPv6 address of the implicit join solicitation originator. This is expected to be the source address of the implicit join solicitation.
- Hop Limit: 255
- Type: 102 - It identifies a implicit MLD Query message
- Code: 0
- Max Response Delay: The maximum time that the receiving RDC can spend before it provides the sender RDC neighbour with an Indirect MLD report.
- PCoA: The Proactive CoA multicast group for which the neighbour RDC is required to maintain membership, pending receipt of an indirect MLD Report message.

6.11.3 Indirect MLD Report/Done

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     PCoA multicast address                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 45: Indirect MLD Report/Done message

- Source Address: the global IPv6 address of the RDC that originaly received the I-MLD Query message. This is the CURRENT RDC which accomdates currently the MN.
- Destination Address: the global IPv6 address of the RDC neighbour that sent an I-MLD Query.
- Hop Limit: 255
- Router Alert value: 13
- Type: 103 - It identifies a implicit MLD Report/Done message
- Code:
  - 0 - identifies a report message
  - 1 - identifies a done message
- PCoA: The Proactive CoA multicast group for which the neighbour RDC is required to maintain membership, as received in the I-MLD Query message.

6.12 PCoA-Enable/Disable message

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Reserved                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      SCoA flag bitmap         |     SCoA tuple identifier     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     PCoA Start lifetime       |      PCoA stop lifetime       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 46: PCoA-Enable/Disable message

- Source Address: the global IPv6 address of the MN that received the original PRS-PSH message.
- Destination Address: the global IPv6 address of the RDC that effected the PRS-PSH message towards that MN. If the message Code is:
  - 0: this RDC destination is the CURRENT one.
  - 1: this RDC destination has transitted to the PREVIOUS state and thus is the previous one.
- Type: 110 - It identifies a PCoA enable/disable message
- Code:
  - 0 - identifies an enable request
  - 1 - identifies a disable request.
- SCoA flag bitmap: The flag bitmap, depending whether the message identifies an enable or disable request, provides the for an enable request the highest priority canditate(s) soft CoA(s), while for a disable request provides the active primary CoA of the MN at the new CURRENT RDC.
- SCoA tuple identifier: This is a 16-bit integer which tracks at the CURRENT/PREVIOUS RDC the SCoA tuple generate for a particular MN. This identifier is only unique per RDC and depends on the PRS-PSH message.
- PCoA Start lifetime: This is a 16-bit time value on expiry of which the CURRENT/ PREVIOUS RDC must effect the enable or disable request for the PCoA group forwarding activation.

- For message code:
  - 0: the CURRENT RDC must enable forwarding of traffic towards the MN over the PCoA group.
  - 1: the PREVIOUS RDC must disable forwarding of traffic towards the MN over the PCoA group.
- PCoA stop lifetime: This is a 16-bit time value which identifies the duration of an enable or disable request. For message code:
  - 0: the CURRENT RDC must set a duration for which the enabled forwarding of traffic towards the MN over the PCoA group will be valid.
  - 1: the PREVIOUS RDC must disable forwarding of traffic towards the MN over the PCoA group for that duration.
- Valid value for both PCoA start and stop lifetime are also the valies:
  - -1: is the only valid negative time value and represents infinity
  - 0: effecting immediate enable/disable over the PCoA group.

6.13 Indirect Neighbour advertisment

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S|I|L|        Reserved         |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+          MN  LLA              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                             SCoA                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 47: Indirect Neighbour advertisment

- Source Address: the global IPv6 address of the PREVIOUS RDC.
- Destination Address: the global IPv6 address of some RDC neighbour of the PREVIOUS RDC.
- Type: 120 - It identifies a Indirect Neighbour advertisment
- Code: 0
- MN LLA: the link layer address for which a neighbour cache must be set proactively. It requires the presence of the S and L flag.
- SCoA: the global/link-local IPv6 address for which a neighbour cache entry must either be created or invalidated. It requires either the existence of the S or I flag, but not both.
- S: set a neighbour cache entry proactively
- I: invalidate the neighbour cache entry
- L: LLA of MN. The flag must be set if configuring a neighbour cache entry proactively

References

[1] M. Parthasarathy A. E. Yegin and C. Williams. Mobile ipv6 neighborhood routing for fast handoff. Internet Draft, Internet Engineering Task Force, October 2000.

[2] C. Diot B. Levine, J. Crowcroft, J.J Garcia-Lune-Aceves, and J. Kurose. Consideration of Receiver Interest for ip Multicast Delivery. In *Proceeding of INFOCOM 2000*, July 2000.

[3] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing. RFC 2189, Internet Engineering Task Force, September 1997.

[4] R. Boivie and N. Feldman. Small Group Multicast. Internet Draft, February 2001.

[5] R. Coltun, D. Ferguson, and J. Moy. OSPF for IPv6. RFC 2740, Internet Engineering Task Force, December 1999.

[6] A. Conta and S. Deering. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 2463, Internet Engineering Task Force, December 1998.

[7] M. Crawford. Transmission of IPv6 Packets over Ethernet Networks. RFC 2464, Internet Engineering Task Force, December 1998.

[8] S. Deering, W. Fenner, and B. Haberman. Multicast Listener Discovery (MLD) for IPv6. RFC 2710, Internet Engineering Task Force, October 1999.

[9] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, Internet Engineering Task Force, December 1998.

[10] S.E. Deering. Host extensions for IP multicasting. RFC 1112, Internet Engineering Task Force, August 1989.

[11] Steve Deering, Bill Fenner, Brad Cain, A. Thyagarajan, and I Kouvelas. Internet Group Management Protocol, Version 3. Internet Draft, March 2001.

[12] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. RFC 2117, Internet Engineering Task Force, June 1997.

[13] Yutaka Ezaki and Yuji Imai. Mobile IPv6 handoff by Explicit Multicast. Internet Draft, May 2001.

[14] W. Fenner. Internet Group Management Protocol, Version 2. RFC 2236, Internet Engineering Task Force, November 1997.

[15] William Fenner. IGMP-based Multicast Forwarding ('IGMP Proxying'). Internet Draft, April 2001.

[16] S. Floyd and V. Jacobson. The synchronization of periodic routing messages. In *IEEE/ACM Transactions on Networking*, April 1994.

[17] C. Perkins G. Tsirtsis, A. Yegin, G. Dommety, and K. El-Malki. Fast handovers for mobile ipv6 (work in progress). Internet Draft, Internet Engineering Task Force, October 2000.

[18] S. Hanna, B. Patel, and M. Shah. Multicast Address Dynamic Client Allocation Protocol (M ADCAP). RFC 2730, Internet Engineering Task Force, December 1999.

[19] A. Helmy. A multicast-based protocol for ip mobility support. In *ACM 2nd International Workshop on Networked Group Communication (NGC2000)*, November 2000.

[20] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 2373, Internet Engineering Task Force, July 1998.

[21] D. Johnson and C. Perkins. Mobility support in IPv6 (work in progress). Internet Draft, Internet Engineering Task Force, November 1998.

[22] J. Kempf and P. Calhoun. Foreign agent assisted hand-off (work in progress). Internet Draft, Internet Engineering Task Force, January 2000.

[23] R. Koodli and C. Perkins. Fast handovers in mobile IPv6 (work in progress). Internet Draft, Internet Engineering Task Force, October 2000.

[24] R. Koodli and C. Perkins. A framework for smooth handovers with mobile IPv6 (work in progress). Internet Draft, Internet Engineering Task Force, November 2000.

[25] Jiwoong Lee. SGM support in Mobile IP. Internet Draft, October 2000.

[26] Jiwoong Lee. Explicit Multicast Extension (Xcast+) Supporting Receiver Initiated Join. Internet Draft, February 2001.

[27] K Leung, G Dommety, Madhavi Subbarao, and Raj Patil. Local Mobility Agents in IPv6. Internet Draft, October 2000.

[28] J. Malinen and C. Perkins. Mobile IPv6 regional registrations (work in progress). Internet Draft, Internet Engineering Task Force, July 2000.

[29] G. Malkin and R. Minnear. RIPng for IPv6. RFC 2080, Internet Engineering Task Force, January 1997.

[30] J. Mysore and V. Bharghavan. A new multicasting-based architecture for internet host mobility. In *Proceedings of ACM MobiCom 97*, September 1997.

[31] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). RFC 2461, Internet Engineering Task Force, December 1998.

[32] S. Seshan, H. Balakrishnan, and R. Katz. Handoffs in cellular wireless networks: The daedalus implementation and experience, 1996.

[33] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 2462, Internet Engineering Task Force, December 1998.

[34] Kazuaki Tsuchiya, Hidemitsu Higuchi, Sunao Sawada, and Shinji Nozaki. An IPv6/IPv4 Multicast Translator based on IGMP/MLD Proxying (mtp). Internet Draft, May 2001.

[35] D. Waitzman, C. Partridge, and S.E. Deering. Distance Vector Multicast Routing Protocol. RFC 1075, Internet Engineering Task Force, November 1988.

[36] Imai Yuji. Multiple Destination option on IPv6(MDO6). Internet Draft, September 2000.

7 Acknowledgements

We would like to thank Hannu Flinck, Jari Malinen, Vijay Devarapali and Charlie Perkins (Nokia Research), for their valuable discussions on problems encountered during the design of this draft specification. This work has been sponsored by Nokia Research, NRC Mountain View, USA.

We would also like to thank Bob Quinn (StartDust), Jon Crowcroft (UCL) and Bill Fenner (ATT) for their comments over issues related to multicast forwarding.

8 Addresses

The working group can be contacted via the current chairs:

Basavaraj Patil
Nokia Corporation
6000 Connection Drive
M/S M8-540
Irving, Texas 75039
USA
Phone: +1 972-894-6709
Fax : +1 972-894-5349
EMail: Basavaraj.Patil@nokia.com

Phil Roberts
Motorola
1501 West Shure Drive
Arlington Heights, IL 60004
USA
Phone: +1 847-632-3148
EMail: QA3445@email.mot.com

Questions about this memo can also be directed to the authors:

Theo Pagtzis
Dept. of Computer Science
University College London
Gower Street
London WC1E 6BT
United Kingdom
Phone: +44 (0)20 7679 3704
EMail: t.pagtzis@cs.ucl.ac.uk
Fax: +44 (0)20 7387 1397

Peter Kirstein
Dept. of Computer Science
University College London
Gower Street
London WC1E 6BT
United Kingdom
Phone: +44 (0)20 7679 7286
EMail: p.kirstein@cs.ucl.ac.uk
Fax: +44 (0)20 7387 1397