Protocol Specification of the MOMBASA Software Environment

A. Festag, L. Westerhoff
{festag|westerhoff}@ee.tu-berlin.de

Berlin, October 2001
Version 1.3
# Contents

1 Introduction 5

2 System Overview 7

3 Protocol Description 11
   3.1 Addressing .................................................. 11
   3.2 Multicast ..................................................... 12
   3.3 Tables for Paging, Registration and Routing .............. 13
   3.4 Timers ......................................................... 14
   3.5 Policies ....................................................... 15
   3.6 Buffer Management ........................................... 15
   3.7 Overview of Protocol Operations ......................... 15
      3.7.1 Initial Registration and Address Allocation ....... 15
      3.7.2 Packet Transport ....................................... 16
      3.7.3 Handover ................................................ 16
      3.7.4 Transition from Active to Inactive Mode ......... 17
      3.7.5 Transition from Inactive to Active Mode .......... 17
      3.7.6 Paging .................................................. 18
   3.8 Protocols .................................................... 18

4 Message Formats 20
   4.1 LHP Message Formats ........................................ 20
      4.1.1 MEP Advertisements .................................... 20
      4.1.2 MEP Solicitation ........................................ 21
      4.1.3 MH Registration Request ............................... 21
      4.1.4 MH Registration Reply .................................. 23
   4.2 IMP Message Formats ........................................ 24
      4.2.1 Inter-MEP Advertisement ............................... 24
   4.3 MEP-GWP Message Formats .................................. 24
      4.3.1 Paging Request .......................................... 24
      4.3.2 Paging Update .......................................... 25
   4.4 IGMP Message Formats ....................................... 26
      4.4.1 IGMP ...................................................... 26
      4.4.2 IGMP Membership Query Message ..................... 26
      4.4.3 IGMPv3 Membership Report Message .................. 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.4</td>
<td>IGMPv2 Leave Group Message</td>
<td>28</td>
</tr>
<tr>
<td>4.5</td>
<td>PIM Message Formats</td>
<td>30</td>
</tr>
<tr>
<td>4.5.1</td>
<td>PIM Hello Message</td>
<td>30</td>
</tr>
<tr>
<td>4.5.2</td>
<td>PIM Register Message</td>
<td>31</td>
</tr>
<tr>
<td>4.5.3</td>
<td>PIM Register-Stop Message</td>
<td>32</td>
</tr>
<tr>
<td>4.5.4</td>
<td>PIM Join/Prune Message</td>
<td>32</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Other PIM Messages</td>
<td>33</td>
</tr>
<tr>
<td>5</td>
<td>Protocol specification</td>
<td>34</td>
</tr>
<tr>
<td>5.1</td>
<td>Mobile Agent State Machine</td>
<td>48</td>
</tr>
<tr>
<td>5.2</td>
<td>MEP State Machine</td>
<td>59</td>
</tr>
<tr>
<td>5.3</td>
<td>Gateway Proxy State Machine</td>
<td>65</td>
</tr>
<tr>
<td>5.4</td>
<td>Multicast Router State Machine</td>
<td>73</td>
</tr>
<tr>
<td>6</td>
<td>Simulation Trace</td>
<td>76</td>
</tr>
<tr>
<td>7</td>
<td>Acronyms</td>
<td>91</td>
</tr>
</tbody>
</table>
List of Figures

2.1 System architecture ................................................. 8
2.2 Paging Area: Non-overlapping (left) and Overlapping (right) ............... 9
3.1 Address translation by means of encapsulation/decapsulation ................. 12
3.2 Address translation by means of NAT .................................... 13
3.3 MOMBASA protocols involved in signaling for handover .................... 19
Chapter 1

Introduction

In general IP multicast supports location–independent addressing and routing. This ability of multicast is similar to the requirements of mobility support in IP–based networks, though in a different context. MOMBASA stands for Mobility support – A Multicast–Based Approach. MOMBASA intends to utilize multicast for network–level mobility support. Contrary to the classic mobility approach of address translation and indirect routing (e.g. IETF Mobile IP \[16\], \[22\]), MOMBASA eliminates the need for IP address translation. Therefore MOMBASA has three main advantages:

- Rerouting for handover is executed in a network node where the path to the old and the new access point diverge (and not in an mobility agent on the mobile’s home network which might be distant to the current location of the mobile).

- As a matter of principle, a (possibly future) multicast infrastructure and signaling is reused for handover. The multicast is complemented by handover–specific functionality (such as paging, last-hop-signaling etc.). Nevertheless it is not required to replace the multicast by a handover–specific infrastructure and signaling at all.

- Multicast offers flexible mechanisms to control the service interruption. In the utmost case packets can be distributed efficiently to potential new access points which buffer the packets and the handover latency is reduced to an absolute minimum.

The MOMBASA software environment is a generic platform to investigate the utilization of multicast for host mobility in IP-based networks. This document specifies protocols for this environment. The specification is the basis for the implementation.

The MOMBASA software environment provides the following features:

- The MOMBASA software environment provides mobility support in an access network. Multicast is used for micro–mobility\(^2\) support.

- Even though multicast might be used as a sole mechanism for routing of packets in mobile networks (see e.g. \[20\]), typical mobility–specific functionalities are added complementing the basic multicast to support location management, handover initiation, paging, indirect registration and other functionalities.

\(^1\)This work has been partly supported by SIEMENS.

\(^2\)Handover between cells of an access network. Micro–mobility is transparent to the macro–mobility protocol.
• The software environment provides a generic interface to the multicast protocol. The interface can be adapted to support certain classes of multicast approaches. At this stage, the implementation of the MOMBASA software environment employs IGMPv2 / PIM-SM \cite{12,13}; IGMPv3 and SSM \cite{3,4} will be examined at the coming stage. However, the specification is independent of the particular multicast routing and access protocols as much as possible.

• The MOMBASA software environment works consequently with soft states.

• Policies grant a high degree of freedom (e.g. for rerouting, paging, interface selection, etc.)

The outline of the document is as follows: In the second chapter, the overall system is described and important terms defined. In chapter three, the overall protocol is explained. Message formats are specified in chapter four. Finally, the protocol specification can be found in chapter five and a simulation trace described by Message Sequence Charts (MSCs) in chapter six.
Chapter 2

System Overview

The system consists of the following core elements, as shown in figure 2.1. These include:

- **Mobile User (MU).** Person equipped with a single or several communication devices and using communication services. A Mobile User traverses the coverage of a cellular network while on the move. A Mobile User is identified by a unique personal identifier.

- **Mobile Host (MH).** An IP-based communication device equipped with a single or multiple wireless network interfaces. Multiple interfaces usually support different technologies. With the success of software radio an interface might adapt to different wireless technologies. A Mobile Host may vary from a dumb terminal to a powerful stand-alone machine.

- **Access Point (AP).** Stationary network node with a single or multiple down-link interfaces and a wire-line interface. It can be considered as a router which routes packets between the wireless and the wire-line network part. An Access Point performs Network Address Translation between multicast and unicast addresses. Additionally mobility-related tasks are performed by means of signaling. With respect to the Multicast Routers in the Handoff Domain, the Access Points act as multicast receivers. Access Points cooperate to support seamless handover.

- **Multicast Router (MR).** Stationary network node with multiple wire-line interfaces between which packets (both multicast and unicast) are routed. It is assumed that a Multicast Router is a standard device and has no specific mobility-related functionality.

- **Gateway (GW).** A stationary network node with multiple wire-line interfaces. A Gateway interconnects the Handover Domain with the global Internet. A Gateway performs Network Address Translation between unicast and multicast addresses. It acts as a Multicast Router and has additional mobility-related tasks.

Additionally the following terms are important:

- **Handover Domain.** An IP network under the control of a single authority. It consists of Access Points, Multicast Routers and Gateways. It may consist of multiple subnets.

---

1In this document it is assumed that the last hop between MH and AP is wireless though this is not a necessary for mobility.
• **Paging Area.** Set of wireless cells with a specific geographic coverage. When the exact location of the Mobile Host is not known, the Mobile Host is located by *Paging* within a Paging Area. Paging avoids locating a Mobile Host by broadcast.

• **Access Point Group.** Access Points which are potential candidates for handover form an Access Point group. When a Mobile Host registers with the actual Access Point the other Access Points belonging to the group will also subscribe to the multicast channel. The current Access Point is called *active*, the other Access Points *passive*. A group of Access Points is usually formed of adjacent and/or overlapping wireless cells.

Hosts and servers in the Internet communicating with Mobile Hosts in the Handover Domain are usually characterized as *Corresponding Hosts*.

It is assumed that a Mobile User can be identified by a unique personal identifier. Moreover, a Internet-wide location service is assumed which resolves the unique personal identifier of the Mobile User into IP addresses identifying the Mobile Host. Future mobility-aware applications will be able to communicate with the Mobile User directly (via the shortest path). Non-mobility-aware applications can communicate with the Mobile User via a Personal Proxy. Routing via a Personal Proxy causes indirect routing with all the disadvantages known from Mobile IP. Nevertheless a Personal Proxy can convert media (fax to email, voice to email, etc.) and preserves the location privacy of the Mobile User. Details of this architecture are beyond the scope of this document and it is referred to [18][23].

The software components of the system are executed on the Mobile Hosts, Access Points, Multicast Routers and Gateways. The components are:

• **Mobile Agent.** A Mobile Agent is executed on the Mobile Host and performs mobility-related tasks.

• **Mobility-Enabling Proxy.** A MEP is executed on the Access Point. A MEP advertises its services in the wireless cell(s), manages registration tables, initiates and controls handoffs, exchanges routing informations with the Gateway Proxy and with other MEPs and controls...
traffic between the wired and wireless networks part. A MEP executes address translation
between the Mobile Host’s multicast and unicast address.\footnote{The term MEP group and Access Point group is used interchangeable.}

- **Multicast Router demon.** A Multicast Router demon is executed on the Multicast Router. The
type of demon depends only on the supported multicast protocol and need not be mobility
aware.

- **Gateway Proxy (GWP).** The GWP is executed on the Gateway. The Gateway Proxy (or Paging
demon) performs mobility-related tasks, such as location management and paging.

A Mobile Host may move freely through the coverage of wireless cells, while being associated
with an Access Point and communicating. When it crosses cell boundaries a handover to the new
Access Point is executed.

The main features of the system are:

- Addressing and routing based on IP- and IP-style multicast

- Multicast proxies in access points to disburden the mobile host from multicast group manage-
ment for mobility support,

- Advertisements/Solicitations to advertise the availability of Mobility-Enabling Proxies (MEPs),

- Inter-MEP advertisements to register mobile hosts in advance,

- Packets can be distributed in advance to candidate Access Points by means of multicast. These
passive Access Points buffer the packets. When the Mobile Host registers with one of the
Access Points, it becomes active and the buffered packets will be forwarded to the Mobile
Host. The other Access Points drop the buffered packets. This method reduces the handover
latency to a minimum.

- The system supports heterogeneous wireless technologies, in particular multiple interfaces in
the Mobile Host are supported.
- The system differentiate between active and inactive mobile hosts. Inactive Mobile Hosts may have passive connectivity, where the frequency of re-registrations is decreased. This reduces the energy consumption of Mobile Hosts and signaling load on the wireless link.

- **Paging** is used to determine the exact location of a Mobile Hosts before delivering packets destined to an inactive Mobile Host. Paging allows the Mobile Host to update its location information less frequently at the cost of providing the network with only approximate location information. For paging in MOMBASA, wireless cells are grouped to Paging Areas and are identified by a multicast channel (see figure 2.2 for Paging Areas with non-overlapping and overlapping wireless cells). When locating a Mobile Host by paging, a paging request is sent via the paging channel. The usage of multicast facilitates a flexible and dynamic composition of Paging Area. In particular, the Paging Areas are not tied to the wired network topology in the Handover Domain.

- In principle, the system facilitates policies to provide flexibility and efficiency and to control system behavior. The policies separate decision making from algorithms. In particular, policies are used to select the *optimal* access point among several possible ones and when to handover, and for control of buffering, forwarding and paging algorithms.

- The system facilitates soft-state of the location-management tables as well as of the multicast routing tables. Whereas the soft-state is an inherent feature of the multicast routing protocol, its consequent application for the location management allows the storage of network state in a distributed manner across various nodes and protocols.
Chapter 3

Protocol Description

3.1 Addressing

It is assumed that a Mobile User is identified by a permanent and unique Personal Identifier, which can be resolved to application-specific addresses and to IP addresses, respectively. We do not constrain to a specific address format; a Network Access Identifier (NAI) \[^1\] might be reasonable. This address type is widely used as a userID (e.g. user@realm.com) submitted by the client during PPP authentication.

Moreover, it is assumed that a Mobile Host is identified by two addresses: multicast channel and a unicast IP address. A multicast channel is a combination of a unicast IP source address S and the SSM destination address G, which form a (S, G) pair. In the described architecture the source address S is usually the Gateway’s unicast IP address. For G the address realm 232/8 (232.0.0.0 - 232.255.255.255) is reserved\[^1\].

When a Mobile Host enters the Handover Domain the first time, a multicast channel and a unicast IP address is assigned to the Mobile Host. Both addresses are quasi-permanent: they are assigned once and remain unchanged as long as the Mobile Host is registered within the Handover Domain. Within the handover domain, the multicast channel is used to transport down-link data packets, the unicast IP address is used for up-link data packets. Outside the Handover Domain the unicast IP address is used for both up-link and down-link packets. Therefore the unicast address must be a globally valid IP address. The multicast channel is regarded as a private address and valid inside the Handover Domain, only.

In the MOMBASA software environment a specific addressing concept is used: Suppose, for the Handover Domain a pool of unicast IP addresses is reserved which can be assigned to Mobile Hosts. Then, each unicast IP address has a corresponding IP address in the multicast address realm (232/8 addresses). Both the unicast and multicast address pools are continuous thus the multicast address of a Mobile Host can be simply determined by address mapping. See table\[^3.1\] for an example.

\[^1\]Designated as Source-Specific Multicast (SSM) destination addresses \[^2\]
Table 3.1: Address mapping between unicast addresses and multicast channels

<table>
<thead>
<tr>
<th>Unicast IP address</th>
<th>Multicast channel (S,G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>130.149.49.1</td>
<td>(S,232.149.49.1)</td>
</tr>
<tr>
<td>130.149.49.2</td>
<td>(S,232.149.49.2)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>130.149.255.254</td>
<td>(S,232.149.255.254)</td>
</tr>
</tbody>
</table>

The use of multicast addresses in the Handover Domain requires an address translation by IP encapsulation [21] or by Network Address Translation (NAT) [9]. See figures 3.1 and 3.2 for examples.

Figure 3.1: Address translation by means of encapsulation/decapsulation

One part of the multicast address space is reserved for signaling operations, i.e. for

- MEP groups
- Paging Area

3.2 Multicast

Multicast is an essential part of the system. Multicast is provided for:

- Transport of data packets in the down-link direction from the Gateway Proxy towards the MEPs. (In particular multicast provides efficient data distribution to a group of MEPs in order to support seamless handover),
- Transport of signaling packets to locate inactive Mobile Hosts with paging,
- Management of multicast groups, such as paging areas and MEP groups,
3.3 Tables for Paging, Registration and Routing

It is distinguished between paging, registration and routing tables. All network nodes in the Handover Domain make use of their standard routing tables MRIB and RIB (multicast and unicast routing information base). Additionally each MEP manages a registration table and the Gateway Proxy a paging table. The paging and registration tables have soft states. They require a periodic refresh to keep network state alive. Moreover a multicast routing protocol uses soft state mechanisms to adapt to the underlying network conditions and to dynamics of multicast group membership.

The registration table of a MEP has an entry for each active Mobile Host registered directly and additionally entries for Mobiles which have been pre-registered by other MEPs. An entry consists of:

- Link-level address of the Mobile Host,
- Interface of the last received registration,
- Unicast IP address of the Mobile Host,
- Flag if the Mobile Host has registered directly or was indirectly pre-registered via another MEP,
- Requested registration lifetime,
- Remaining registration lifetime of a pending or current registration.

A new entry is inserted in the registration table when a Mobile Host registers directly with a MEP or is indirectly pre-registered by another MEP. An entry is updated when an entry already exists: Three cases can be distinguished:

1. The Mobile Host was directly registered and re-registers.
2. The Mobile Host was directly registered and is pre-registered by another MEP.
3. The Mobile Host was indirectly pre-registered by another MEP and registers directly.

Note that inactive Mobile Hosts do not have an entry.
When the remaining lifetime is less than zero or a registration message with lifetime zero is received, the entry will be removed from the registration table.

The paging table of the Gateway Proxy has an entry for each Mobile Host in the Handover Domain. An entry consists of:

- Unicast IP address of the Mobile Host,
- Paging Area id of last paging-update,
- Requested paging-update lifetime,
- Remaining lifetime of the current paging-update.
- Registration identification for reordering detection.

A new entry is inserted into the paging table and the entry is updated when a paging update is sent by a Mobility-Enabling Proxy (MEP) for an inactive Mobile Host. When the remaining lifetime of the current paging table entry is less than zero or a paging update with lifetime zero is received, the entry is removed from the paging table.

For routing of packets, the standard routing tables are used. Unicast packets are routed by means of the standard unicast routing table (RIB); multicast packets use the standard multicast routing table (MRIB).

For a Mobile Host two cases have to be considered, whether the Mobile Host is **active** or **inactive**.

For an **active** Mobile Host, entries in registration tables and entries for the Mobile Host’s multicast channel in the multicast routers exist. For an **inactive** Mobile Host, the multicast routing table entries of the multicast routers and the registration table entries are timed out, whereas the paging table entry still exists. The re-establishment of the multicast routing table entries can be triggered by paging.

For efficiency reasons the multicast router provides a multicast forwarding cache (MFC) at kernel level and additionally a multicast routing table (MRIB) at user space level in the multicast routing daemon. However, here the paging functionality is realized in a paging demon and requires an interaction with the multicast functionality.

### 3.4 Timers

Timers are used to realize soft-state in a distributed manner across the network nodes. In general two types of timers are distinguished: Timers that control the sending of signaling messages for registration and advertisements and timers for aging out state in registration and paging tables. In table 3.2 the MOMBASA-specific timers are listed. Note that table 3.2 does not include timers which control states in multicast routing tables (MRIB) or multicast forwarding caches (MFC).
### 3.5 Policies

In general, policies are a set of rules to administer, manage, and control access to network resources [19]. In MOMBASA, policies are used to:

- select the interface in order to determine the optimal network access,
- determine when to handover (including hysteresis),
- allocate the optimal buffer space in passive Access Points buffering packets,
- select the optimal rerouting policies: break-before-make, make-before-break or predictive handover.

### 3.6 Buffer Management

For predictive handover, passive Access Points buffer packets. The buffer is allocated when the Mobile Host is pre-registered with the MEP and de-allocated when the Mobile Host’s entry is removed from the registration list or the Mobile Host registers directly with the Mobility-Enabling Proxy. Determining the allocated buffer size is policy-controlled and for further study.

### 3.7 Overview of Protocol Operations

The next section provides a rough overview of the protocol operations. It is divided into initial registration and address allocation, packet transport, handover and paging.

#### 3.7.1 Initial Registration and Address Allocation

1. MEPs advertise their availability on each link for which they provide service. Alternatively, a Mobile Host may solicit a MEP Advertisement from a MEP through a MEP solicitation.

2. When a Mobile Host enters the Handover Domain, it associates a unicast IP address with its interfaces. The address may be dynamically acquired by the Mobile Host through a protocol

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
<th>Representative value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>re-registration timer</strong></td>
<td>Expected inter-arrival time of registrations in active/inactive mode. Triggers a re-registration with the MEP.</td>
<td>10 sec./30 sec.</td>
</tr>
<tr>
<td><strong>registration timeout</strong></td>
<td>Validity of a registration table entry.</td>
<td>30 sec.</td>
</tr>
<tr>
<td><strong>paging timeout</strong></td>
<td>Validity of a paging entry.</td>
<td>90 sec.</td>
</tr>
<tr>
<td><strong>activity timeout</strong></td>
<td>Time the host remains in active state w/o incoming data.</td>
<td>10 sec.</td>
</tr>
<tr>
<td><strong>advertisement timer</strong></td>
<td>Expected inter-arrival time of advertisements.</td>
<td>100 msec.</td>
</tr>
</tbody>
</table>

Table 3.2: Timers in the MOMBASA software environment
such as DHCP [8] or may be owned as a long-term address for its use only in the Handover Domain. This step should be tied with authorization and authentication.

3. The Mobile Host sends a registration request to the MEP indicating an active state.

4. The MEP processes the registration request and inserts a new entry in its registration table, sends a reply to the Mobile Host, determines the multicast channel assigned to the mobile host’s IP address and subscribes for the multicast channel on behalf of the mobile host. Additionally the MEP triggers the other MEPs in its group to likewise subscribe for the multicast channel.

Alternatively, a Mobile Host may go immediately into inactive state after acquiring the unicast IP address. In this case the Mobile Host sends an inactive registration to the MEP which sends a paging update to the Gateway Proxy on behalf of the Mobile Host.

As long as the Mobile Host remains in the active state, a re-registration to the MEP is sent frequently. This, in turn triggers the MEP to send a paging update to the Gateway Proxy.

### 3.7.2 Packet Transport

At first it is assumed that the Gateway and MEP applies IP-IP encapsulation and decapsulation, respectively (see figure 3.1). The case where the Gateway performs NAT (see figure 3.2) is similar.

- **Downstream traffic directed to the Mobile Host:**
  1. A CH may send packets to the unicast IP address of the Mobile Host.
  2. The Gateway Proxy intercepts the packets and performs IP-IP encapsulation. If an entry in the multicast forwarding cache (MFC) for the channel exists, the packet is forwarded directly towards the multicast channel. Otherwise the Gateway Proxy is informed and packets are queued. If no entry exists in the paging table the multicast routing demon is notified. If an entry in the multicast routing table in the demon exists, the queued packets are forwarded. In the case that an entry in the paging table exists, the packets are buffered in the Gateway Proxy and the Mobile Host will be paged. (The paging case will be considered separately. See section 3.7.6)
  3. Intermediate multicast routers forward the packet along the multicast channel.
  4. When the active MEP receives a multicast packet, the MEP decapsulates the packet and forwards it to the Mobile Host’s link local address. A passive MEP buffers the packet.
  5. Finally, the Mobile Host receives the packet.

- **Upstream traffic is originated by the Mobile Host and is sent via unicast.

### 3.7.3 Handover

- **Mobile Host initiated handover**
  1. The need for handover is detected by one of the algorithms for link availability detection: monitoring of layer-2 parameters, advertisements-based or polling.
2. If the Mobile Host is in active state, it sends an active registration request to the MEP.

3. The MEP processes the registration request and updates its registration table. If an entry in the MEP’s registration table was already existing due to a pre-registration by another MEP then buffered packets (if any) will be forwarded to the Mobile Host. Additionally the MEP triggers the other MEPs in its group to likewise subscribe for the multicast channel.

4. When the Mobile Host’s entry in the registration table of the old MEP times out then the entry is removed and the MEP triggers other MEPs in its group to un-subscribe the Mobile Host’s multicast channel. The Mobile Host may also send an explicit de-registration message (registration message with lifetime zero).

5. When a MEP has both a direct and an indirect registration, the direct registration has higher priority than the indirect one. However, the indirect registration is not discarded but noted in the registration entry. When the direct registration expires (or is deleted by a de-registration) and the indirect one is still valid the MEP becomes a passive MEP. This avoids certain race conditions after handovers.

Alternatively, a handover can be initiated by the MEP.

### 3.7.4 Transition from Active to Inactive Mode

The transition from active to inactive mode happens as follows:

1. The Mobile Host detects that for some time data was neither sent nor received.

2. The Mobile Host sends a registration request denoting inactive to a MEP.

3. If the MEP has a registration entry for the Mobile Host this entry is deleted and MEPs in the same MEP group are notified. The MEP un-subscribes from the corresponding multicast channel. A paging update is generated and sent to the Gateway Proxy.

4. The Gateway Proxy receives the paging update and generates an entry in the paging table.

5. When the Mobile Host’s entry in the registration table of the old MEP times out then the entry is removed and the MEP triggers other MEPs in its group to un-subscribe the Mobile Host’s multicast channel. The Mobile Host may (and should if possible) send an explicit de-registration message (registration message with lifetime zero) to the old MEP.

### 3.7.5 Transition from Inactive to Active Mode

Wakeup (the transition from inactive to active mode) is performed when the inactive Mobile Host wants to send data or when it is paged (see next section).

1. The Mobile Host sends a registration request indicating active state and that the registration is a wakeup to a MEP of its choice.

2. The MEP processes the registration request and inserts a new entry in its registration table, sends a reply to the Mobile Host, determines the multicast channel assigned to the mobile host’s IP address and subscribes for the multicast channel on behalf of the mobile host. Additionally
the MEP triggers the other MEPs in its group to likewise subscribe for the multicast channel. Finally the MEP sends a paging update with lifetime zero to the Gateway Proxy.

3. The Gateway Proxy receives the paging update and removes its entry in the paging table.

### 3.7.6 Paging

For paging it is assumed that all MEPs belonging to a specific paging area have subscribed to the corresponding multicast channel.

1. When a Gateway Proxy receives a packet and has an entry for the destination Mobile Host in its paging table the Gateway Proxy buffers the packet. It sends a paging request to the paging area. For this the Gateway Proxy determines the IP multicast address corresponding to the paging area and forwards the paging request on the multicast channel.

2. The MEPs receive the paging request and forwards it on their wireless interface.

3. When the Mobile Host finds it unicast IP address in the paging request, the Mobile Host performs the wakeup procedure as described in the previous section.

### 3.8 Protocols

Consider a typical MOMBASA network which uses IGMP and PIM-SSM as multicast protocols. Then the overall network uses five protocols

- *Last-Hop Protocol (LHP)*. Defined protocol between Mobile Host and MEP.
- *Inter-MEP Protocol (IMP)*. Defined protocol between MEPs.
- *MEP-GWP protocol*. Defined protocol between MEP and GWP.
Figure 3.3: MOMBASA protocols involved in signaling for handover
Chapter 4

Message Formats

4.1 LHP Message Formats

The messages of the Last-Hop Protocol (LHP) are based on IETF Mobile IP message [22] types with minor modifications. Numeric values are always in network byte order (Big Endian).

4.1.1 MEP Advertisements

A MEP Advertisement is sent by a MEP to advertise its availability on each link it provides service. A MEP Advertisement is formed by extending a ICMP Router Advertisement message [6].

IP fields:
- Source Address: IP address of the corresponding MEP interface.
- Destination Address: Broadcast address 255.255.255.255
- TTL: 1

ICMP fields:
- Type: 9: ICMP Router Advertisement
- Code: 17 This is a MEP, not a normal router and it does not route common traffic.
- Lifetime: The maximum time in milliseconds that the MEP advertisement is considered valid in the absence of further MEP Advertisements.
- Router Address: MEPs own address.
- Num Addr: Number of router addresses advertised in the ICMP message. Set to 1.

The MEP Advertisement extensions follows the ICMP Router Advertisement fields.
4.1.2 MEP Solicitation

A MEP solicitation is sent by a Mobile Host to solicit a MEP Advertisement from a MEP. A MEP solicitation is identical to a ICMP Router Solicitation with the restriction that the IP TTL field is set to 1.

4.1.3 MH Registration Request

A registration request is sent by a Mobile Host to the MEP. A Mobile Host registers when a Mobile Host

- is in the initial or active state and detects that it has moved to a new MEP,
- detects that its registration-request timeout is about to expire, indicating that an expected reply to a sent registration-request is missing,
- detects that its re-registration timer is about to expire,
- is in active state and detects that its current MEP has rebooted,
- transits from active to inactive state or vice versa.

When the Mobile Host is in active state it sets the appropriate flag in the registration request message.

IP fields:

Source Address: Mobile Hosts unicast IP address.
Destination Address: MEP IP address from the corresponding MEP Advertisement.
TTL: Set to 1.
UDP fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Port</td>
<td>Variable</td>
</tr>
<tr>
<td>Destination Port</td>
<td>434</td>
</tr>
</tbody>
</table>

The UDP header is followed by:

```
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
 0                   1                   2                   3
             Type    |  Reserved |  Reserved |  Reserved |  Mobile Host Unicast IP Address |
                  |           |           |           |                                |
                  |           |           |           |  Identification               |
                  |           |           |           |                                |
                  |           |           |           |  Extensions...                 |
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>2 (Registration Request)</td>
</tr>
<tr>
<td>S</td>
<td>Not used (Simultaneous Binding, for IETF Mobile IP only)</td>
</tr>
<tr>
<td>B</td>
<td>Broadcast datagrams</td>
</tr>
<tr>
<td>D</td>
<td>Not used (Decapsulation by Mobile Host, for IETF Mobile IP only)</td>
</tr>
<tr>
<td>M</td>
<td>Not used (Minimal encapsulation, for IETF Mobile IP only)</td>
</tr>
<tr>
<td>G</td>
<td>Not used (GRE encapsulation, for IETF Mobile IP only)</td>
</tr>
<tr>
<td>V</td>
<td>Not used (VanJacobson Header Compression, for IETF Mobile IP only)</td>
</tr>
<tr>
<td>P</td>
<td>Indicates if the Mobile Host supports Paging or not (Not available in IETF Mobile IP).</td>
</tr>
<tr>
<td>A</td>
<td>Indicates that the Mobile Host is active. (Not available in IETF Mobile IP).</td>
</tr>
<tr>
<td>Lifetime</td>
<td>Number of seconds remaining before the registration is considered expired. The lifetime for the inactive mode should be larger than for the active mode. Zero indicates a de-registration request. 0xffff indicates infinity.</td>
</tr>
<tr>
<td>RRP</td>
<td>Rerouting Policy.</td>
</tr>
<tr>
<td>W</td>
<td>The Mobile Host is just waking up (A must be also set).</td>
</tr>
<tr>
<td>Reserved</td>
<td>Zeroed.</td>
</tr>
<tr>
<td>Mobile Host Unicast IP Address</td>
<td>IP Address of the Mobile Host.</td>
</tr>
<tr>
<td>Identification</td>
<td>Used for matching registration requests with registration reply, and for protecting against replay attacks of registration messages [22].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Extension</th>
<th>Optional Mobile Host - Gateway authentication extension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Prefix Length Extension, One-byte Padding Extension).</td>
</tr>
</tbody>
</table>
4.1.4 MH Registration Reply

A registration reply is sent by the MEP to reply to a registration request.

**IP fields:**
- **Source Address** Copied from the MEP registration request destination field.
- **Destination Address** MEP IP address.

**UDP fields:**
- **Source Port** Variable.
- **Destination Port** Copied from Source Port of the MEP registration request.

The UDP header is followed by:

```
0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3
+---------------+---------------+---------------+---------------+
| Type | Code | Lifetime | Reserved |
| 00 | 00 | 00 | 00 |
+---------------+---------------+---------------+---------------+
| Identification | | | |
+---------------+---------------+---------------+---------------+
| Extensions... | | | |
```

**Type** 4 (Registration Reply)
**Code** Indicates the result of the Registration Request.
**Lifetime** Number of seconds remaining before the registration is considered expired.
**Reserved** Zeroed.
**Identification** Used for matching registration Requests with registration Reply, and for protecting against replay attacks of registration messages [22].
**Extensions** Mobile Host - Gateway authentication extension (optional)

The following Codes are defined:
- 0 Registration accepted
- 64 Reason unspecified
- 65 Administratively prohibited
- 66 Insufficient resources
- 67 Mobile Host failed authentication
- 69 Requested lifetime too long
- 70 Poorly formed Request
- 80 GW host unreachable (ICMP error received)
- 81 GW port unreachable (ICMP error received)
- 82 GW unreachable (other ICMP error received)
4.2 IMP Message Formats

4.2.1 Inter-MEP Advertisement

A inter-MEP advertisements is sent by a MEP to other MEPs to advertise its availability and to pre-register Mobile Host with other MEPs of its MEP group. Therefore the group of MEPs form a IP multicast group with a pre-defined IP multicast address whereas the currently active MEP acts as a multicast sender.

IP fields:
- Source Address: IP address from which the message is sent.
- Destination Address: Multicast IP address for the MEP group.

UDP fields:
- Source Port: Variable
- Destination Port: 434

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold-time</td>
<td>Reserved</td>
<td>Unicast IP address of Mobile Host N</td>
</tr>
</tbody>
</table>

Type: 6 (Inter-MEP Advertisement)
Length: Length of the extension excluding type and length fields.
Sequence Number: Count of Inter-MEP Advertisement sent since the MEP was initialized.
Hold-time: Number of seconds remaining until the pre-registration is considered expired.
Reserved: Zerod.
Unicast IP address: Unicast IP address of the Mobile Host to pre-register.

4.3 MEP-GWP Message Formats

4.3.1 Paging Request

A paging request is sent by the Gateway Proxy to a Paging Area which is identified by a multicast channel. The MEPs broadcast the paging request in their wireless cells or send them to the unicast IP address of the paged Mobile Host.

IP fields:
- Source Address: Address from which the message is sent.
- Destination Address: Multicast address of the paging area.

UDP fields:
- Source Port: Variable
- Destination Port: 434
The UDP header is followed by:

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Unicast IP address of Mobile Host</td>
</tr>
</tbody>
</table>

*Type* 8 (Paging Request)
*Length* Length of the extension (6 Bytes).
*Sequence Number* Count number of paging requests. Used to distinguish the paging request messages.

### 4.3.2 Paging Update

A paging update is sent by the Mobile Host to refresh the Gateway Proxy’s paging cache. It is sent frequently if the Mobile Host is in inactive state.

**IP fields:**
- **Source Address** Unicast address of the Mobile Host.
- **Destination Address** Address of the Gateway Proxy.

**UDP fields:**
- **Source Port** Variable
- **Destination Port** 434

The UDP header is followed by:

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Unicast IP address of Mobile Host</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Current Paging Area Id</td>
</tr>
</tbody>
</table>

*Type* 10 (Paging Update)
*Length* Length of the message.
*Lifetime* Number of seconds remaining before the lifetime is considered expired. Copied from registration request.

*Identification* Identification copied from registration request.
*Current Paging Area ID* Multicast Address of the Paging Area. Set by the current MEP.
4.4 IGMP Message Formats

4.4.1 IGMP

IGMP version 1 [5], version 2 [13] and version 3 [4] provide a method through which a host can join or leave a multicast group. IGMP version 3 adds support for Single Source Multicast (SSM): a host or router may report interest in receiving packets only from specific source addresses, or from all but specific source addresses sent to a specific multicast address.

IGMP messages are encapsulated in IP datagrams, with an IP protocol number of 2. The IGMP messages are sent with a TTL of 1. They carry an IP Router Alert Option [17] in its IP header.

The following types are defined:

- 0 Hello
- 0x11 Membership Query (General Query or Group-Specific Query or Group-and-Source-Specific Query)
- 0x12 Version 1 Membership Report
- 0x16 Version 2 Membership Report
- 0x22 Version 3 Membership Report
- 0x17 Leave Group

In MOMBASA II only IGMPv3 is provided.

4.4.2 IGMP Membership Query Message

Sent by Multicast Routers to query the multicast reception state of neighboring interfaces of MEPs or other Multicast Routers.

There are three variants:

- **General Query.** Sent by a multicast router to learn the complete multicast reception state of the neighboring interfaces. General Queries are periodically sent. The Group Address field and the Number of sources are zero.

- **Group-Specific Query.** Sent by a multicast router to learn the reception state for a single multicast address of the neighboring interfaces. The Group Address field contains the multicast address of interest, the Number of sources field contains zero.

- **Group-and-Source-Specific Query.** Sent by a multicast router to learn if any neighboring interface desires reception of packets for a specific multicast address from any address of the list of sources. The Group address field contains the multicast address of interest and the Source Address fields contain the source address(es) of interest.

**IP fields:**

- **Destination Address** 224.0.0.1 for General Query (All-hosts multicast group) Multicast address of interest for Group-Specific and Group-and-Source-Specific Query.
- **TTL** 1
IGMP Membership Query message format:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>0x11 (IGMP Membership Query)</td>
</tr>
<tr>
<td><strong>Max Resp Time</strong></td>
<td>Maximum allowed time before sending a responding report, in units of 1/10s.</td>
</tr>
<tr>
<td><strong>Checksum</strong></td>
<td>Checksum of the whole IGMP message (entire IP payload).</td>
</tr>
<tr>
<td><strong>Group Address</strong></td>
<td>Multicast group address. Zeroed in a General Query. Multicast address in a</td>
</tr>
<tr>
<td></td>
<td>Group-Specific or Group-and-Source-specific query.</td>
</tr>
<tr>
<td><strong>Rsv</strong></td>
<td>Zeroed.</td>
</tr>
<tr>
<td><strong>S</strong></td>
<td>Suppress Router-Side Processing (S-Flag). It indicates to a multicast</td>
</tr>
<tr>
<td></td>
<td>router to suppress the normal timer updates upon hearing a request.</td>
</tr>
<tr>
<td><strong>QRV</strong></td>
<td>Querier’s Robustness Variable. It allows tuning for the expected packet</td>
</tr>
<tr>
<td></td>
<td>loss. If a network is expected to be lossy, the Robustness Variable may be</td>
</tr>
<tr>
<td></td>
<td>increased. Default: 2</td>
</tr>
<tr>
<td><strong>QQI</strong></td>
<td>Querier’s Query Interval in seconds. If a Multicast Router is not the current</td>
</tr>
<tr>
<td></td>
<td>querier adopt the value from the most recently received Query. Default:</td>
</tr>
<tr>
<td></td>
<td>135 sec.</td>
</tr>
<tr>
<td><strong>Number of Sources</strong></td>
<td>Number of sources present in the Query. The field is zero in a General</td>
</tr>
<tr>
<td></td>
<td>Query or a Group-Specific Query.</td>
</tr>
<tr>
<td><strong>Source address</strong></td>
<td>IP unicast addresses.</td>
</tr>
</tbody>
</table>

4.4.3 IGMPv3 Membership Report Message

Sent by a MEP or Multicast Router to report the current multicast reception state of their interfaces.

IP fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Address</td>
<td>IP address of the corresponding interface in the MEP or Multicast Router.</td>
</tr>
<tr>
<td>Destination Address</td>
<td>224.0.0.13 (All IGMPv3-capable routers) (for IGMPv1 and v2 the multicast</td>
</tr>
<tr>
<td></td>
<td>group address specified in the Group Address field of the Membership Query</td>
</tr>
<tr>
<td></td>
<td>message).</td>
</tr>
<tr>
<td>TTL</td>
<td>1</td>
</tr>
</tbody>
</table>

IGMP Membership Report message format:
Each Group Record consists of

<table>
<thead>
<tr>
<th>Type</th>
<th>Record Type</th>
<th>Aux Data Len</th>
<th>Number of Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x12</td>
<td>IGMPv1 Membership Report</td>
<td>IGMPv2 Membership Report,</td>
<td>IGMPv3 Membership Report,</td>
</tr>
<tr>
<td>0x16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x22</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reserved
Checksum
Number of Group Records
Group Record
Contains information about the sender’s membership in a multicast group on the interface from which the Report is sent.

Record Type
Current state of record or change of the filter mode. In INCLUDE mode, reception of packets is requested only for those IP parameters listed in the source list. In EXCLUDE mode, reception of packets is requested from all IP source addresses except those listed in the source-list.

1 MODE_ISINCLUDE
2 MODEISEXCLUDE
3 CHANGE_TO_INCLUDE_MODE
4 CHANGE_TO_EXCLUDE_MODE
5 ALLOW_NEW_SOURCES
6 BLOCK_OLD_SOURCES

Aux Data Len
Length of the Auxiliary Data field (units of 32-bit words).

Number of sources
Number of sources present in the report.

Auxiliary Data
Not used.

4.4.4 IGMPv2 Leave Group Message

A IGMPv2 Leave Group message is sent by a MEP to un-subscribe from a multicast channel.

Source Address
IP address of the corresponding interface in the MEP.

Destination Address
224.0.0.2 (All routers).

IGMPv2 Leave Group message format:
**Type** 0x17 (IGMPv2 Leave Group)

**Reserved** Zeroed

**Checksum** Checksum of the whole IGMP message (entire IP payload).

**Group Address** Multicast group address to leave.
4.5 PIM Message Formats

All PIM control messages have protocol number 103.

The following PIM-types are defined:

- **0 Hello**
- **1 Register**
- **2 Register-Stop**
- **3 Join/Prune**
- **4 Bootstrap**
- **5 Assert**
- **6 Graft (for PIM-DM only)**
- **7 Graft-Ack (for PIM-DM only)**
- **8 Candidate-acsRP-Advertisement**

4.5.1 PIM Hello Message

A PIM Hello message is sent by periodically by the Gateway and Multicast Routers on each PIM-enabled interface. They allow to learn the neighboring PIM router.

IP fields:

- **Source Address**
  - IP address of the corresponding interface in the MEP or Multicast Router.
- **Destination Address**
  - 224.0.0.13 (All PIM routers)

PIM Hello message format:
**Version**  Version of PIM 2

**Type**  0 (Hello).

**Reserved**  Zeroed.

**Checksum**  Standard IP checksum.

**Option Type**  Type of the option given in the *Option Value* field.
- 1 Hold-time
- 2-16 reserved for future use
- 17-18 deprecated, not to use
- 19 DR Priority
- 20 Generation ID
- 21 State Refresh Option for PIM DM
- 22-65000 To be assigned by IANA
- 65001-65535 Reserved for private use.

**Option Length**  Length of the *Option Value* field in bytes. For

<table>
<thead>
<tr>
<th><strong>Option Type</strong></th>
<th><strong>Option Length</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
</tr>
</tbody>
</table>

**Option Value**  See *Option Type* field

### 4.5.2 PIM Register Message

Sent by a DR to the RP when a multicast packet needs to be transmitted on the multicast distribution tree.

**IP fields:**
- **Source Address**  IP Address of the DR.
- **Destination Address**  IP address of the RP.

**PIM Register message format:**

```plaintext
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Version   Type   Reserved   Checksum
```

Multicast data packet
**4.5.3 PIM Register-Stop Message**

Sent by a RP to the DR (sender of the Register-message).

**IP fields:**

- **Source Address**: Address to which the Register message was addressed.
- **Destination Address**: Source address of the Register message.

**PIM Register message format:**

| 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 |
| Version | Type | Reserved | Reserved | Encoded-Group Address | Encoded-Unicast-Source Address |

- **Version**: Version of PIM 2
- **Type**: 2 (Register-Stop).
- **Reserved**: Zeroed.
- **Checksum**: IP checksum.

**Encoded-Group Address**: Encoded multicast group address

**Encoded-Unicast-Source Address**: Address of source from multicast data packet in Register-message

**4.5.4 PIM Join/Prune Message**

Sent by Multicast Routers towards upstream sources and RPs.

**IP fields:**

- **Source Address**: Address of the Multicast Router.
- **Destination Address**: Address of upstream router or RP.

**PIM Join/Prune message format:**

| 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 |
| Version | Type | Reserved | Reserved | Encoded-Group Address | Encoded-Unicast-Source Address |

- **Version**: Version of PIM 2
- **Type**: 2 (Join/Prune).
- **Reserved**: Zeroed.
- **Checksum**: Encoded multicast group address

**Encoded-Unicast-Source Address**: Address of source from multicast data packet in Register-message
Version
Type
Encoded-Unicast-Upstream Neighbor Address
Hold-time
Number of Groups
Encoded-Multicast Group Address
Number of Joined Sources
Join Source Address-1..n
Number of Pruned Sources
Prune Source Address-1..n

Version of PIM 2
3 (Register).
Encoded IP address of the upstream neighbor.
Amount of time in seconds a receiver must keep the Join/Prune state alive.
Number of multicast group sets contained in this message.
A wild card group (*,*,acsRP) join is represented by a 224.0.0.0 in the group address and '4' in the mask length field.
Number of Joined Sources listed for a specific multicast group.
List with addresses of sources. The sending router will forward multicast datagrams for these sources.
Number of prune source addresses listed for a group.
List with addresses of sources. The sending router does not want to forward multicast datagrams for these sources.

4.5.5 Other PIM Messages

Bootstrap-, Assert-, Graft-, Graft-Ack-, and Candidate-acsRP-Advertisement messages are beyond the scope of this document.
Chapter 5

Protocol specification

The MOMBASA system is formally specified using SDL [10], a general-purpose specification language for communication systems. The system is modeled as communicating processes of the components described in section 2 and their environment. Each process is regarded as an Extended Finite State Machine (EFSM). Communication is represented by signals and can take place between the processes and the environment of the system. The environment generates signals (such as handover) which are considered as external events of the MOMBASA system.

In the following, the major parts of the specification are listed. At first, an overview of the SDL blocks, definitions of signals, message types, and data types used in the particular processes (registration tables, paging tables, etc.) is given. Then, the state machine of each process is shown. Due to space limitations, the procedure specifications are not included in this report.
SIGNAL
    env_MA_Restart,
    env_MA_Stop,
    env_MEP_Restart,
    env_MEP_Stop,
    env_MCR_Restart,
    env_MCR_Stop,
    env_GW_Restart,
    env_GW_Stop,
    env_Wake_up,
    env_Act_TO,
    env_Pag_Trig,
    env_HC_Trig(uint8_t),
    env_Warning(CharString),
    MEP_Advert(MEP_Advert_Packet),
    MEP_Advert(IMEP_Advert_Packet),
    IMEP_Advert(IMEP_Advert_Packet),
    MEP_Solicit(MEP_Solicit_Packet),
    MEP_Subscribe_MCC(MB_MC_Channel),
    MEP_Unsubscribe_MCC(MB_MC_Channel),
    MEP_stopBuffer(MB_Buffer),
    MEP_buffer_stopped(uint32_t),
    MEP_startFlush,
    MEP_stopFlush,
    MEP_flush_stopped(uint32_t),
    MCR_Subscribe_MCC(MB_MC_Channel),
    MCR_Unsubscribe_MCC(MB_MC_Channel),
    MA_RegReq(MA_RegReq_Packet),
    MA_RegReply(MA_RegReply_Packet),
    GW_PagReq(GW_PagReq_Packet),
    MCR_PagReq(GW_PagReq_Packet),
    MEP_PagReq(IMEP_PagReq_Packet),
    MEP_PagUpd(MA_PagUpd_Packet),
    MCR_PagUpd(MA_PagUpd_Packet),
    GW_stopBuffer(MB_Buffer),
    GW_buffer_stopped(uint32_t),
    GW_startFlush,
    GW_stopFlush,
    GW_flush_stopped(uint32_t);
System Mombasa

Signal_List_Definitions(13)

SIGNALLIST
MA_from_ENV = env_MA_Restart,env_MA_Stop,env_WakeUp,env_Act_TO,env_HO_Trig;
SIGNALLIST
MA_from_MEP = MEP_Advert,MA_RegReply,MEP_PagReq;
SIGNALLIST
MA_to_MEP = MEP_Solicit,MA_RegReq;
SIGNALLIST
MEP_from_ENV = env_MEP_Restart,env_MEP_Stop;
SIGNALLIST
MEP_from_MCR = MCR_PagReq,MCR_IMEP_Advert;
SIGNALLIST
MEP_to_MCR = MEP_Subscribe_MCC,MEP_Unsubscribe_MCC,IMEP_Advert,MEP_PagUpd;
SIGNALLIST
MCR_from_ENV = env_MCR_Restart,env_MCR_Stop;
SIGNALLIST
GW_from_ENV = env_GW_Restart,env_GW_Stop,env_Pag_Trig;
SIGNALLIST
GW_to_MCR = GW_PagReq;
SIGNALLIST
GW_from_MCR = MCR_Subscribe_MCC,MCR_Unsubscribe_MCC,MCR_PagUpd;
SIGNALLIST
to_ENV = env_Warning;
newtype MB_IPHdr struct
   proto uint16_t;
   src IP_Addr_t;
   dest IP_Addr_t;
endnewtype MB_IPHdr;

newtype IP_Addr_t
   Array(uint2_t, uint8_t)
endnewtype IP_Addr_t;

newtype MA_RegIdent struct
   high uint32_t;
   low uint32_t;
endnewtype MA_RegIdent;

newtype MB_Interface struct
   name CharString;
   index uint8_t;
   addr  IP_Addr_t;
   netmask IP_Addr_t;
   broadcast IP_Addr_t;
endnewtype MB_Interface;

newtype RRP_t
   literals HARD,SOFT,PREDICTIVE,NO_RRP;
endnewtype RRP_t;

newtype MB_Tunnel_t
   literals MC_NAT,MC_ENCAPS;
endnewtype MB_Tunnel_t;

newtype MB_Route struct
   dest IP_Addr_t;
   src IP_Addr_t;
   iif uint8_t; /* Input interface index */
   oif uint8_t; /* Output interface index */
   gateway IP_Addr_t;
endnewtype MB_Route;

newtype MB_Buffer struct
   mb_size size_t;
   empty Boolean;
endnewtype MB_Buffer;

/* MB_Data_Types II */
newtype MB_MC_Channel struct
   s_addr IP_Addr_t;
   g_addr IP_Addr_t;
endnewtype MB_MC_Channel;

newtype MRT_Entry_t struct
   ch MB_MC_Channel;
   subscribed Boolean;
endnewtype MRT_Entry_t;

newtype MB_MRT_t
   Array(uint2_t,MRT_Entry_t)
/* Array(uint32_t,MRT_Entry_t) */
endnewtype MB_MRT_t;

newtype MB_PAT_Entry_t struct
   g_addr IP_Addr_t;
   mep_pid1 PiD;
   mep_pid2 PiD;
   mep_pid3 PiD;
endnewtype MB_PAT_Entry_t;

newtype MB_PAT_t
   Array(uint2_t,MB_PAT_Entry_t)
endnewtype MB_PAT_t;

calcChecksum
MB_Warning
/* MEP_Advert */
newtype MEP_Advert struct
   icmp_type uint8_t;
   icmp_code uint8_t;
   icmp_cksum uint16_t; /* must be 1 */
   icmp_wsa uint8_t;
   icmp_lifetime duration;
   ira_addr IP_Addr_t;
   ira_preference uint32_t;
   mb_type uint8_t;
   mb_length uint8_t;
   mb_seq_no uint8_t;
   mb_maxregtime Duration;
   B_flag Boolean; /* Busy */
   P_flag Boolean; /* Paging */
   extension Character;
endnewtype MEP_Advert;

newtype MEP_Advert_Packet struct
   ip_header MB_IPHdr;
   adv MEP_Advert;
endnewtype MEP_Advert_Packet;

newtype MEP_Advert_Info struct
   pkt MEP_Advert_Packet;
   iface MB_Interface;
   quality uint8_t;
   from_pid PiD;
endnewtype MEP_Advert_Info;

/* IMEP_Advert */
newtype IMEP_Advert struct
   mb_type uint8_t;
   length uint8_t;
   seqno uint16_t;
   holdtime duration;
   reserved uint16_t;
   mobile1 IP_Addr_t;
   mobile2 IP_Addr_t;
   mobile3 IP_Addr_t;
endnewtype IMEP_Advert;

newtype IMEP_Advert_Packet struct
   ip_header MB_IPHdr;
   iadv IMEP_Advert;
endnewtype IMEP_Advert_Packet;

newtype IMEP_Advert_Info struct
   from_addr IP_Addr_t;
   pkt IMEP_Advert_Packet;
   from_pid PiD;
endnewtype IMEP_Advert_Info;

/* MEP_Solicit */
newtype MEP_Solicit struct
   icmp_code uint8_t; /* type sub code */
   icmp_cksum uint16_t; /* ones complement checksum of struct */
   icmp_reserved uint32_t;
endnewtype MEP_Solicit;

newtype MEP_Solicit_Packet struct
   ip_header MB_IPHdr;
   sol MEP_Solicit;
endnewtype MEP_Solicit_Packet;

newtype MEP_Solicit_Info struct
   pkt MEP_Solicit_Packet;
   mep_iface MEP_Interface;
   quality uint8_t;
   from_pid PiD;
endnewtype MEP_Solicit_Info;
newtype MA_RegReq struct                 /* MA_RegReq */
   mb_type uint8_t;
   send_PagUpd Boolean; /* Paging update */
      is sent when the mobile wakes up */
   B_flag Boolean; /* Broadcast datagrams */
   P_flag Boolean; /* Mobile supports paging */
   A_flag Boolean; /* Mobile is active */
   lifetime duration;
   RRP RRP_t;
   reserved uint32_t;
   mh_addr IP_Addr_t;
   id uint32_t;
   extension Character;
endnewtype MA_RegReq;

newtype MA_RegReq_Packet struct
   ip_header MB_IPHdr;
   regreq MA_RegReq;
endnewtype MA_RegReq_Packet;

newtype MA_RegReq_Info struct
   pkt MA_RegReq_Packet;
   mep_iface MEP_Interface;
   quality uint8_t;
   bs MA_BaseStationEntry;
   port uint16_t;
   from_pid PiD;
endnewtype MA_RegReq_Info;

newtype MA_RegReply struct      /* MA_RegReply */
   mb_type uint8_t;
   code uint8_t;
   lifetime DURATION;
   id uint32_t;
   extension Character;
endnewtype MA_RegReply;

newtype MA_RegReply_Packet struct
   ip_header MB_IPHdr;
   regreply MA_RegReply;
endnewtype MA_RegReply_Packet;

newtype MA_RegReply_Info struct
   pkt MA_RegReply_Packet;
   MA_iface MA_Interface;
   quality uint8_t; /* quality of received advertisement */
   addr IP_Addr_t; /* quality of received advertisement */
   addr IP_Addr_t;
   port uint16_t;
   pending_reg_info MA_RegReq_Info;
   from_pid PiD;
endnewtype MA_RegReply_Info;
/* MA_PagUpd */
newtype MA_PagUpd struct
   mb_type uint8_t;
   length uint8_t;
   lifetime duration;
   mh_addr IP_Addr_t;
   paging_area IP_Addr_t;
   id uint32_t;
endnewtype MA_PagUpd;

newtype MA_PagUpd_Packet struct
   ip_header MB_IPHdr;
   pagupd MA_PagUpd;
endnewtype MA_PagUpd_Packet;

newtype MA_PagUpd_Info struct
   pkt MA_PagUpd_Packet;
   mb_iface MB_Interface;
   from_pid PID;
endnewtype MA_PagUpd_Info;

/* GW_PagReq */
newtype GW_PagReq struct
   mb_type uint8_t;
   length uint8_t;
   seqno uint8_t;
   mh_addr IP_Addr_t;
endnewtype GW_PagReq;

newtype GW_PagReq_Packet struct
   ip_header MB_IPHdr;
   pagreq GW_PagReq;
endnewtype GW_PagReq_Packet;

newtype GW_PagReq_Info struct
   pkt GW_PagReq_Packet;
   mb_iface MB_Interface;
   from_pid PID;
endnewtype GW_PagReq_Info;
/* Gateway type definitions */

newtype GW_Config struct
   upstream GW_Interface;
   downstream GW_InterfaceMap;
   mobile_uc uint32_t; /* mobile unicast address range */
   mobile_mc uint32_t; /* mobile multicast address range */
   mobile_mask uint32_t;
   mobile_mask_len uint8_t;
   tunnel_type MB_Tunnel_t;
   paging_port uint16_t; /* PAGING disabled if paging_port = 0 */
   paging_source IP_Addr_t;
   paging_areas uint32_t;
   paging_areas_mask uint32_t;
   paging_areas_masklen uint8_t;
   max_mobiles uint8_t;
   max_cachetime Duration;
   mobile_buffer_size uint32_t;
endnewtype GW_Config;

newtype GW_Interface struct
   mb_iface MB_Interface;
   force_addr IP_Addr_t;
endnewtype GW_Interface;

newtype GW_InterfaceMap
   Array(iface_index,GW_Interface);
endnewtype GW_InterfaceMap;

newtype GW_MobileEntry struct
   addr IP_Addr_t; /* Unicast IP address of MH */
   location IP_Addr_t; /* Last known paging area */
   gw_iface GW_Interface; /* Interface from which */
   /* last paging update was heard */
   lifetime Duration;
   lastid uint32_t;
   paging Boolean;
   flushing Boolean;
endnewtype GW_MobileEntry;

newtype GW_MobileMap
   /* Array(uint32_t,GW_MobileEntry); */
   Array(uint2_t,GW_MobileEntry);
endnewtype GW_MobileMap;
/* MEP type definitions 1 */
newtype MEP_Config struct
   upstream US_Interface;
   downstream MEP_InterfaceMap;
   send_paging_updates Boolean;
   mobile_uc uint32_t; /* mobile unicast address range */
   mobile_mask uint32_t;
   mobile_mc uint32_t; /* mobile multicast address range */
   mobile_mask_len uint32_t;
   tunnel_type MB_Tunnel_t;
   reg_port uint16_t;
   imep_port uint16_t;
   paging_port uint16_t;
   primary_paging_area MB_MC_Channel;
   secondary_paging_area MB_MC_Channel;
   primary_mep_group MB_MC_Channel;
   secondary_mep_group MB_MC_Channel;
   gw_addr IP_Addr_t;
   max_direct_mobiles uint32_t;
   max_indirect_mobiles uint32_t; /* number above which to send shortage warnings */
   max_regtime Duration;
   mobile_buffer_size uint32_t;
   sent_mepadv Boolean;
   sent_imepadv Boolean;
   pseudo_seed Duration;
   /* for pseudo random number */
endnewtype MEP_Config;
/* MEP type definitions 2 */
newtype MEP_Interface struct
   mb_iface MB_Interface;
   force_addr IP_Addr_t;
   adv_interval duration; /* in msec */
   adv_lifetime duration; /* in msec */
   adv_seqno uint8_t;
endnewtype MEP_Interface;

newtype MEP_InterfaceMap
   Array(iface_index,MEP_Interface);
endnewtype MEP_InterfaceMap;

newtype US_Interface struct
   mb_iface MB_Interface;
   force_addr IP_Addr_t;
   imep_interval duration; /* in msec */
   imep_lifetime duration; /* in msec */
   imep_seqno uint32_t;
endnewtype US_Interface;

newtype MEP_BaseStationEntry struct
   addr IP_Addr_t; /*IP address of other base station */
   seqno uint8_t;
   holdtime duration;
endnewtype MEP_BaseStationEntry;

newtype MEP_MobileEntry struct
   addr IP_Addr_t; /*Unicast IP address of MH*/
   is_direct Boolean;
   mb_iface MB_Interface;
   lifetime duration;
   quality uint8_t;
   B_flag Boolean; /* Broadcast datagrams */
   P_flag Boolean; /* Paging */
   A_flag Boolean; /* Active */
   RRP RRP_t;
   mb_channel MB_MC_Channel;
endnewtype MEP_MobileEntry;

newtype MEP_MobileMap
   Array(uint2_t,MEP_MobileEntry);
endnewtype MEP_MobileMap;

newtype MEP_BaseStationMap
   Array(uint32_t,MEP_BaseStationEntry);
endnewtype MEP_BaseStationMap;
/* Mobile Agent type definitions */

newtype MA_Config struct
   ma_iface MA_InterfaceMap;
   reg_id uint32_t;
   paging_enabled Boolean;
   rrp RRP_t;
   broadcast Boolean;
   paging_port uint16_t;
   idle_timeout Duration;
   config_file CharString;
endnewtype MA_Config;

newtype MA_Interface struct
   mb_iface MB_Interface;
   force_addr IP_Addr_t;
   preference uint32_t;
   active_regtime duration;
   idle_regtime duration;
   regreq_timeout duration;
   send_solicit Boolean;
   send_dereg Boolean;
endnewtype MA_Interface;

newtype MA_InterfaceMap
   Array(iface_index,MA_Interface);
endnewtype MA_InterfaceMap;

newtype MA_BaseStationEntry struct
   addr IP_Addr_t;
   registered Boolean;
   busy Boolean;
   stale Boolean;
   quality uint8_t;
   ma_iface MA_Interface;
   adv_seqno uint16_t;
   adv_lifetime duration;
   maxregtime duration;
   P_flag Boolean;
   mep_pid PId;
endnewtype MA_BaseStationEntry;

newtype MA_BaseStationMap
   Array(uint32_t,MA_BaseStationEntry);
endnewtype MA_BaseStationMap;
/* General data types */
syntype uint2_t = Integer constants 0:3 /* 2^2 */ endsyntype;
syntype uint8_t = Integer constants 0:256 /* 2^8 */ endsyntype;
syntype uint16_t = Integer constants 0:65535 /* 2^16 */ endsyntype;
syntype uint32_t = Integer constants 0:4294967296 /* 2^32 */ endsyntype;
syntype size_t = Integer constants 0:1024 endsyntype;
syntype iface_index = Integer constants 0:1 endsyntype;
syntype reg_retry_counter_t = Integer constants 0:DEFAULT_REGRETRY endsyntype;

/* MEP message types */
synonym MB_MEP_ADVERT uint8_t = 17;
synonym MB_REG_REQ uint8_t = 2;
synonym MB_REG_REPLY uint8_t = 4;
synonym MB_IMEP_ADVERT uint8_t = 6;
synonym MB_PAGING_REQ uint8_t = 8;
synonym MB_PAGING_UPDATE uint8_t = 10;

/* ICMP message types RFC1256 */
synonym ICMP_ROUTERADVVERT uint8_t = 9;
synonym ICMP_ROUTERSOLICIT uint8_t = 10;

/* Protocol types RFC1060 */
synonym ICMP_PROT uint16_t = 1;
synonym UDP_PROT uint16_t = 2;

/* Replay codes for registration requests */
synonym REGISTRATION_ACCEPTED Integer = 0;
synonym REASON_UNSPECIFIED Integer = 64;
synonym ADMINISTRATIVELY_PROHIBITED Integer = 65;
synonym INSUFFICIENT_RESOURCES Integer = 66;
synonym MOBILE_HOST_FAILED_AUTHENTICATION Integer = 67;
synonym MOBILE_HOST_REQUESTED_LIFETIME_TOO_LONG Integer = 69;
synonym REQUESTED_LIFETIME_TOO_LONG Integer = 69;
synonym POORLY_FORMED_REQUEST Integer = 70;
synonym GW_HOST_UNREACHABLE Integer = 80; /* ICMP error received */
synonym GW_PORT_UNREACHABLE Integer = 81; /* ICMP error received */
synonym GW_UNREACHABLE Integer = 82; /* other ICMP error received */
/* Defaults */

synonym DEFAULT_PAGING_PORT Integer = 434;
synonym DEFAULT_REG_PORT Integer = 434;
synonym DEFAULT_IMEP_PORT Integer = 434;
synonym DEFAULT_MAX_MOBILES Integer = 2;
synonym DEFAULT_MAX_GATEWAY_MOBILES Integer = 2;
synonym DEFAULT_MOBILE_THRESHOLD Integer = 2;
synonym DEFAULT_MOBILE_TIMEOUT Duration = 3600;
synonym DEFAULT_MAX_PAGINGCACHE_TIME Duration = 3600;
synonym DEFAULT_FLUSH_TIME Duration = 3600;
synonym DEFAULT_REGRETRY Integer = 3;
synonym DEFAULT_MOBILE_BUFFER_SIZE Integer = 262144; /* 256 * 1024 */
synonym DEFAULT_GATEWAY_BUFFER_SIZE Integer = 262144; /* 256 * 1024 */
synonym DEFAULT_ACTIVE_REG_TIMEOUT Duration = 60;
synonym DEFAULT_IDLE_TIMEOUT Duration = 300;
synonym DEFAULT_ADV_INTERVAL Duration = 10;
synonym DEFAULT_ADV_LIFETIME Duration = 20;
synonym DEFAULTGLOBALS_TIMEOUT Duration = 10;
synonym DEFAULT_IMEP_LIFETIME Duration = 20;
synonym DEFAULT_RRP RRP_t = PREDICTIVE;
synonym DEFAULT_MEP_CONFIG_FILE CharString = 'mep_conf_file';
synonym DEFAULT_MA_CONFIG_FILE CharString = 'MA_conf_file';
synonym DEFAULT_GW_CONFIG_FILE CharString = 'gw_conf_file';
5.1 Mobile Agent State Machine

Process MA_SM

MobileAgent_DCL(11)

DCL
bs,oldbs MA_BaseStationEntry,
basestations MA_BaseStationMap,
basestation index uint32_t,
force bs IP_Addr_t,
ho_trigger uint8_t,
default_route MB_Route,
reg_reply_count reg_retry_counter_t,
idletimeout Duration,
mep_adv_pkt MEP_Advert_Packet,
mep_adv_info MEP_Advert_Info,
mep_solicit_pkt MEP_Solicit_Packet,
reg_req_pkt MA_RegReq_Packet,
reg_reply_pkt MA_RegReply_Packet,
pag_req_pkt GW_PagReq_Packet,
pag_reply_pkt GW_PagReply_Packet,
config MA_Config,
success Boolean;
TIMER
Reg_Req_T,
Reg_T,
Pag_T,
MEP_T(uint32_t);
Process MA_SM

MobileAgent_2(11)

- (Wait4MEP)

MEP_T
(basestation_index)

MA_remove_MEP_entry
(basestation_index,basestations)

basestation_index = bs!addr(3)

MA_delete_MEP_entry
(basestation)

MA_selectBS
(basestation,basestations)

(bs!addr(0) = 0) AND
(bs!addr(1) = 0) AND
(bs!addr(2) = 0) AND
(bs!addr(3) = 0)

Wait4MEP

reg_retry_count := DEFAULT_REGRETRY

MA_send_MA_RegReq
(config,bs,TRUE,FALSE,
reg_req_pkt)

pending_reg_info!pkt := reg_req_pkt,
pending_reg_info!bs := bs

pending_reg_info!mep_iface

pending_reg_info!quality

pending_reg_info!port

are not used

reg_retry_count :=
reg_retry_count − 1

Reg_Pending

Adv lifetime
of actual basestation
is timed out!

Copyright at Technical University Berlin. All Rights reserved. TKN-01-014 Page 50
Process MA_SM

Wait4MEP

MEP_Advert ([mep_adv_pkt])

MA_set_mep_adv_info (config, mep_adv_pkt, mep_adv_info)

MA_handle_mep_advert (config, basestations, mep_adv_info, success)

success = TRUE

Wait4MEP

MA_selectBS (bs, basestations)

reg_retry_count > 0

MA_send_MA_RegReq (config, bs, TRUE, FALSE, reg_req_pkt)

pending_reg_info!pkt := reg_req_pkt,
pending_reg_info!bs := bs,
pending_reg_info!mep_iface,
pending_reg_info!quality,
pending_reg_info!port are not used

reg_retry_count := reg_retry_count−1

Reg_Pending

MB_Warning ('MH: state Wait4MEP, reg_retry_count <= 0')

No more retries,
give it up!

Wait4MEP

FALSE TRUE

TRUE FALSE

FALSE

TRUE

Copyright at Technical University Berlin. All Rights reserved. TKN-01-014 Page 51
Process MA_SM

MobileAgent_4(11)

Copyright at Technical University Berlin. All Rights reserved.
Process MA_SM

Reg_Pending

Reg_Req_T

MA_selectBS
(bs,basestations)

Selected BS has not changed

Selected BS has changed

reg_retry_count > 0

reg_retry_count = 0

reg_req_pkt := pending_reg_info!pkt

MA_send_MA_RegReq(config,bs,TRUE,FALSE,
reg_req_pkt)

set(now+(config!ma_iface(0)!regreq_timeout),
Reg_Req_T)

Reg_Pending

TRUE

FALSE

MH: state Reg_Pending, reg_retry_count <=0

No more retries, give it up!

reg_retry_count := DEFAULT_REGRETRY

MA_send_MA_RegReq(config,bs,TRUE,FALSE,
reg_req_pkt)

set(now+(config!ma_iface(0)!regreq_timeout),
Reg_Req_T)

Reg_Pending
Process MA_SM

MobileAgent_7(11)

MA_Active

env_Act_TO

config!paging_enabled = TRUE

MB_Warning (MH: state MA_Active: env_Act_TO received, but Paging is disabled. Ignored.)

MA_Active

reset (Reg_T)

config!ma_iface(0)!send_dereg = TRUE

MA_send_dereg(config!ma_iface(0)!send_dereg)

idletimeout := (config!ma_iface(0)!idle_regtime)/3

set (now + idletimeout, Pag_T)

MA_Inactive

env_HO_Trig (ho_trigger)

force_bs(0) := 192, force_bs(1) := 168, force_bs(2) := 10, force_bs(3) := 10

MA_Handle_HO_Trig (basestations, config, reg_req_pkt, force_bs, success)

success = TRUE

reset (Reg_T)

pending_reg_info!pkt := reg_req_pkt, pending_reg_info!bs := bs

reg_retry_count := DEFAULT_REGRETRY−1

bs := basestations(force bs(3))

MA_Active

Copyright at Technical University Berlin. All Rights reserved.
Process MA_SM

MA_Inactive

env_Wakeup
reset (Pag_T)
MA_selectBS (bs,basestations)
reg_retry_count := DEFAULT_REGRETRY
MA_send_MA_RegReq (config,bs,TRUE,TRUE,reg_req_pkt)
pending_reg_info!pkt := reg_req_pkt, pending_reg_info!bs := bs
reg_retry_count := DEFAULT_REGRETRY
set (now+(config!ma_iface(0)!regreq_timeout), Reg_Req_T)
Reg_Pending

MEP_PagReq (pag_req_pkt)
reset (Pag_T)
MA_selectBS (bs,basestations)
MA_send_MA_RegReq (config,bs,TRUE,TRUE,reg_req_pkt)
pending_reg_info!pkt := reg_req_pkt, pending_reg_info!bs := bs
reg_retry_count := DEFAULT_REGRETRY
set (now+(config!ma_iface(0)!regreq_timeout), Reg_Req_T)
Reg_Pending

pag_T

MEP_Advert (mep_adv_pkt)
MA_selectBS (bs,basestations)
MA_send_MA_RegReq (config,bs,FALSE,FALSE,reg_req_pkt)
idletimeout := (config!ma_iface(0)!idle_regtime)/3
set (now + idletimeout, Pag_T)
Reg_Pending

MA_inactive

MobileAgent_8(11)

Copyright at Technical University Berlin. All Rights reserved.
TKN-01-014
Page 56
5.2 MEP State Machine
Process MEP_SM

```c
FPAR
    mep_id uint2_t;
```

MEP_DCL(5)

```c
DCL
   config MEP_Config,
   mobiles MEP_MobileMap,
   other_basestations MEP_BaseStationMap,
   mep_buffer MB_Buffer,
   buffer_pid PId,
   flush_pid PId,
   reg_req_pkt MA_RegReq_Packet,
   reg_reply_pkt MA_RegReply_Packet,
   pag_upd_pkt MA_PagUpd_Packet,
   pag_req_pkt GW_PagReq_Packet,
   mep_adv_pkt MEP_Advert_Packet,
   imep_adv_pkt IMEP_Advert_Packet,
   imep_adv_info IMEP_Advert_Info,
   mep_solicit_pkt MEP_Solicit_Packet,
   imep_solicit_info IMEP_Solicit_Info,
   mobile_index uint32_t,
   mc_channel MB_MC_Channel;
```

TIMER

```c
    MEP_Adv_T,
    IMEP_Adv_T,
    Reg_T(uint32_t);
```
Process MEP_SM

MEP_init
(DEFAULT MEP_CONFIG_FILE, mobiles, config, mep_adv_pkt, imep_adv_pkt, other_BaseStations)

MEP_Cleanup(config)

MEP_init
(DEFAULT MEP_CONFIG_FILE, mobiles, config, mep_adv_pkt, imep_adv_pkt, other_BaseStations)

Idle

env MEP_Restart

MEP_Cleanup(config)

MEP_flush_stopped(mobile_index)

flush_pid := NULL

MEP_buffer_stopped(mobile_index)

buffer_pid := NULL
Process MEP_SM

FPAR
mep_id uint2_t;

Message handler:

- MEP_handle_MA_RegReq
- MEP_handle_MEP_Solicit
- MEP_handle IMEP_Advert
- MEP_handle_PagingReq
- MEP_handle_dereg

Send procedures:

- MEP_send_MEP_Adv
- MEP_send_IMEP_Adv
- MEP_send_MA_RegReply
- MEP_send_PagUpd
- MEP_send_zeroPagUpd

Utilities:

- MEP_get_MC_Addr
- MEP_remove_mobile_entry
- MEP_get_indirect_mobiles
- MEP_get_random_duration
- MEP_startBufferThread
- MEP_stopBufferThread
- MEP_startFlushThread
- MEP_stopFlushThread

'Get_info' procedures:

- MEP_get_regreq_info
- MEP_get_iadv_info
- MEP_get_solicit_info

Copyright at Technical University Berlin. All Rights reserved.
5.3 Gateway Proxy State Machine
Process GW_P_SM

DCL
    config GW_Config,
    mobiles GW_MobileMap,
    mrt MB_MRT_t,
    mc_channel MB_MC_Channel,
    gw_buffer MB_Buffer,
    buffer_pid PId,
    flush_pid PId,
    last_paging_seqno uint32_t,
    mobile_index uint32_t,
    success Boolean,
    env_pag_req GW_PagReq,
    pag_req_pkt GW_PagReq_Packet,
    pag_upd_pkt MA_PagUpd_Packet,
    pag_upd_info MA_PagUpd_Info;

TIMER
    Mobile_T(uint32_t),
    Pag_T(uint32_t);
Process GW_P_SM

GW_init (GW_CONFIG_FILE, config, mobiles, last_paging_seqno)

Idle

GW_Cleanup (mobiles, gw_buffer, buffer_pid, flush_pid)

GW_stop

env GW_ReInit

GW_init (GW_CONFIG_FILE, config, mobiles, last_paging_seqno)

Idle

GW_stop

MCR_Subscribe_MCC (mc_channel)

GW_handle_Subscribe_MCC (mc_channel, mrt, mobiles, gw_buffer, buffer_pid, flush_pid)

Idle

MCR_Unsubscribe_MCC (mc_channel)

GW_handle_Unsubscribe_MCC (mc_channel, mrt, mobiles, flush_pid)

Idle

env_Pag_Trig

GW_completePagTrig (env_pag_req)

config!paging_port > 0

GW_handle_env_PagTrig (config, mobiles, gw_buffer, buffer_pid, flush_pid, env_pag_req, pag_req_pkt, success)

success

mobile_index := env_pag_req!mh_addr(3)

GW_send_MA_PagReq (mobile_index, last_paging_seqno, mobiles, pag_req_pkt)

Idle

MB_Warning ('GWP: State Idle, env_Pag_Trig received, but PAGING disabled. Ignored.')
Process GW_P_SM

- GW_init
- GW_initConfig
- GW_readConfig
- GW_checkConfig
- GW_openSockets
- GW_initRouting
- GW_initSignalHandler
- GW_init_mobiles

GW_P_Procedures_1(5)

- GW_cleanup
- GW_restoreRouting
- MEP_cleanup
- GW_closeSockets
Process GW_P_BufferThread

```
FPAR
   gw_buffer MB_Buffer,
   mobileindex uint32_t;

GW_initBuffer
(gw_buffer)

mobileindex_to_buffer :=
  mobileindex

GW_stopBuffer
(gw_buffer)
```

Copyright at Technical University Berlin. All Rights reserved.

TKN-01-014
Process GW_P_FlushThread

DCL
   flush_duration Duration,
   mobileindex_to_flush uint32_t,
   success Boolean;

TIMER
   flush_ready_T;

GW_checkBuffer
   (gw_buffer,success)

success TRUE FALSE

mobileindex_to_flush := mobileindex

flush_duration := DEFAULT_FLUSH_TIME

set (now + flush_duration, flush_ready_T)

Forwarding

GW_stopFlush

GW_flush_stopped (mobileindex_to_flush)

MB_Warning ('GW_FlushThread: Forwarding stopped, but not ready yet.')
5.4 Multicast Router State Machine

Process MCR_SM

MCR_SM

DCL

mrt MB_MRT_t,
paging_areas MB_PAT_t,
mc_channel MB_MC_Channel,
nul_addr IP_Addr_t,
me p1_pid, mep2_pid, mep3_pid PID,
pag_upd_pkt MA_PagUpd Packet,
pag_req_pkt GW_PagReq Packet,
iadv_pkt MEP_Advert Packet;

MCR_init_MRT
MCR_init_PA
MCR_start_MEPs
MCR_handle_GW_PagReq
MCR_handle_MA_PagUpd
Process MCR_SM

MCR_init_MRT
(mrt,null_addr)

MCR_start_MEPs
(mep1_pid,mep2_pid,mep3_pid)

MCR_init_PA
(paging_areas,mep1_pid,mep2_pid,mep3_pid)

Idle

MEP_Subscribe_MCC
(mc_channel)
mrt(mc_channel!g_addr(3))!ch := mc_channel
mrt(mc_channel!g_addr(3))!subscribed := TRUE

MCR_Subscribe_MCC
(mc_channel)

MEP_Unsubscribe_MCC
(mc_channel)
mrt(mc_channel!g_addr(3))!ch!g_addr := null_addr
mrt(mc_channel!g_addr(3))!subscribed := FALSE

MCR_Unsubscribe_MCC
(mc_channel)

MEP_PagUpd
(pag_upd_pkt)

MCR_handle_MA_PagUpd
(mrt,pag_upd_pkt)

Idle

GW_PagReq
(pag_req_pkt)

MCR_handle_GW_PagReq
(paging_areas,pag_req_pkt)

Idle

Copyright at Technical University Berlin. All Rights reserved.
Chapter 6

Simulation Trace

In order to give an intuitive understanding of the system behavior, the following figures show typical protocol operations by means of a simulation trace (Message Sequence Charts), i.e.

1. Initial registration: The mobile host is in its INIT state and registers with a MEP.
2. Activity Timeout: The mobile host goes into the state INACTIVE,
3. Wakeup: The mobile host wakes up and goes into state ACTIVE,
4. Handover: The mobile host executes a handover,
5. Activity Timeout: The mobile host goes into the state,
6. Paging: The mobile host is paged by the gateway proxy.

To generate the MSCs, the SDL code was transformed into a number of C source files that are compiled and linked with a SDT runtime library. This process has resulted in an executable program which is used to simulate the system.

In the simulated scenario, the system consists of

- a single mobile host which executes a mobile agent process,
- an access network, which is composed of three access points executing mobility-enabling proxies (MEPs) and a multicast router which executes a simplified multicast routing demon,
- a gateway proxy (GWP)

The three MEPs in the access network are configured that only MEP1 and MEP2 are actively involved in handover. MEP3 plays only a passive role, i.e. it does not send any MEP advertisements and IMEP advertisements. To simulate different cases of grouping access points to MEP groups, we differentiate between primary and secondary MEP groups: A MEP sends IMEP advertisements to the multicast channel of the primary MEP group and receives the IMEP advertisements from the multicast channel of the secondary MEP group. Moreover, in the simulated scenario MEP2 and MEP3 form a paging area, where MEP1 is the only access point in another paging area.

To improve the readability of the MSC, some minor details are not shown, such as IMEP Advertisement, when no mobile host is registered with the particular MEP and subscription to static multicast channels (e.g. MEP groups).
Step 3: Wakeup
Mobile Host goes into the state ACTIVE
env_Act_TO

Step 5: Activity Timeout
Mobile Host goes into the state INACTIVE
Step 5: Activity Timeout

Mobile Host goes into the state INACTIVE

env_0 GW_P_SM_1_1 process GW_P_SM
MCR_SM_1_2 process MCR_SM
MEP_SM_1_4 process MEP_SM
MEP_SM_2_5 process MEP_SM
MEP_SM_3_6 process MEP_SM
MEP_BufferThread_4_11 process MEP_BufferThread
MA_SM_1_3 process MA_SM
Step 6: Paging Trigger
Mobile Host is paged by the gateway proxy.

Copyright at Technical University Berlin. All Rights reserved.

TKN-01-014
Bibliography


Chapter 7

Acronyms

**AP** Access Point

**CH** Correspondent Host

**DR** Designated Router

**GW** Gateway

**GWP** Gateway Proxy

**IGMP** Internet Group Management Protocol

**IMP** Inter-MEP Protocol

**LHP** Last-Hop Protocol

**MEP** Mobility-Enabling Proxy

**MFC** Multicast Forwarding Cache

**MOMBASA** MObility support - a Multicast-BAsed Approach

**MH** Mobile Host

**MU** Mobile User

**MR** Multicast Router

**MRIB** Multicast Routing Information Base

**NAT** Network Address Translation

**PA** Paging Area

**PIM** Protocol-Independent Multicast

**PIM-DM** Protocol-Independent Multicast - Dense Mode

**PIM-SSM** Protocol-Independent Multicast - Single Source Mode
PIM-SM  Protocol-Independent Multicast - Sparse Mode

PPP  Point-to-Point Protocol

RIB  Routing Information Base

RP  Rendezvous Point