Technical University Berlin

Telecommunication Networks Group

Testing the Implementation of the MOMBASA Software Environment

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

Berlin, December 2001, Version 1.4

TKN Technical Report TKN-01-018

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Abstract

This document describes the concept and method for testing the MOMBASA software environment and lists test cases. The purpose of the testing is to ensure the correct behavior of the implementation. By means of the testing it will be shown that the implementation conforms with the specification (conformance testing).

Chapter 1

General

1.1 Introduction

The implementation of the *MOMBASA Software Environment* [5, 6] is based on the specification described in [3]. The specified system consists of blocks for mobile hosts, access points, multicast routers and gateways. SDL [1] is used for protocol specification, a general-purpose specification language for communication systems. The MOMBASA system is modeled as a set of communicating processes and their environment. Each process is regarded as an Extended Finite State Machine (EFSM).

The purpose of the testing is to ensure the correct behavior of the implementation. By means of the test cases it will be shown that the implementation conforms with the specification. This kind of testing is usually referred to as *conformance testing* [4].

In the following the testing concept and environment is described, and the test cases are listed.

1.2 Testing concept and method

For testing of the *MOMBASA Software Environment*, all components – Mobile Agent (MA), Mobility-Enabling Proxy (MEP), and Gateway-Proxy (GW_P) – have to be tested. Hence, there are three distinct implementations to test. In general, the *implementation under test (IUT)* is considered as a black box with a finite set of input and output ports. For testing, the IUT is provided with sequences of input signals (messages) at given *points of control and observation (PCOs)* and the resulting behavior in terms of the signal flow crossing the PCO is observed. The tester function may be divided into *upper* and *lower tester (UT/LT)* accessing the IUT via the corresponding PCOs. The IUT passes the testing only if all outputs match those prescribed by the test cases. The test concept is illustrated in Figure 1.1.

In general, a *distributed test method* is used. In this test method the tester function is distributed among the test system containing the LT (lower tester), and the system under test (SUT) comprising the IUT and the UT (upper tester). The test system and the SUT are connected via an underlying medium service, thus allowing the LT to access the IUT via its PCO. Test coordination procedures (TCP) might be used to coordinate the testing activities of UT and LT. The LT plays the role of the protocol partner (peer) for the IUT, the UT acts as user of the service provided by the IUT. Therefore, LT and IUT are exchanging protocol data units (PDUs), while the UT is accessing the IUT by means

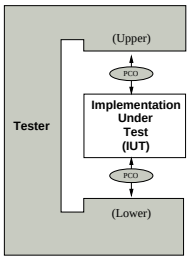


Figure 1.1: Test concept

of abstract service primitives (ASPs) of the service provided by the IUT. The distributed test method is illustrated in Figure 1.2.

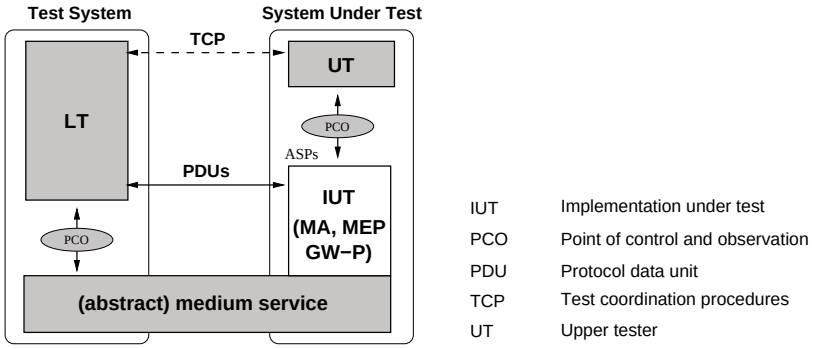


Figure 1.2: Distributed test method

For efficient testing we have instrumented the particular IUTs with management interfaces. This interface is utilized to access the state of the IUT, to retrieve the actual setting of timers, counters and to access databases. For example, when an IUT receives a *status* message, it responds with an output message that uniquely identifies its current state without changing the state. In fact, the management interface facilitates remote testing and supersedes the upper tester as depicted in Figure 1.2.

Several tests include multicast operations. Hence, these tests depend on the type of multicast management protocol. Although the test cases are generally described, the implemented test suites utilize IGMPv2 [2].

One of the challenges of testing implementations for mobility support arise from the behavior of mobile systems: When a mobile is not registered with an access points, it is not able to communicate. This pertains also the communication between the mobile agent and the tester, which is used only for testing purposes. Therefore we have instrumented the implementation of the mobile host with specific testing facilities.

For correct testing the selection of test cases is a major issue. A common method to test implementations based on finite state machines is to generate all possible input signals in all states (structural testing).[1] This method results in a test suite with a length of $4SV$ messages where S rep-

[1] In detail, for each test the implementation is reset. Then the *set* message is used to transfer the implementation into the

resents the number of states and V the number of distinct messages of the input vocabulary. We have used another method. In order to reduce the length of the test suite, we apply only messages which are expected in the particular state and assume a correct behavior of the other implementations. For example, we do not test conditions like: A mobile agent receives a *Registration Reply without having sent a* Registration Request. However, this method includes explicitly input signals which are not part of the input vocabulary of a certain state. Such conditions can be caused by race conditions. In the *MOMBASA Software Environment*, race conditions can occur in the following scenarios:

- Suppose, a mobile host is registered with MEP1. MEP1 has registered the mobile host indirectly with MEP2. Then the mobile host executes a handoff to MEP2 and registers this MEP[2]. In this case, MEP1 continues to register the mobile host indirectly with MEP2 until the mobile host entry in MEP1 times out. In this scenario it has to be ensured that the actual direct registration will not be overwritten by the indirect registration.
- Suppose, a mobile host is in the state INACTIVE. Then it registers with a MEP and starts sending packets. The MEP sends a paging update with lifetime 0 to the gateway and subscribes for the multicast channel. Two cases can occur: First, the paging update message arrives at the gateway before the multicast channel subscription. Second, the multicast channel subscription arrives the gateway before the paging update message.

1.3 Test setup and configuration

For efficient testing a test suite for each IUT has been assembled. The software for testing is included in the source code distribution [5]. It allows automatic testing of the test cases described in section 2 and results in a summary of passed tests.

The test configuration for mobile agent, mobility enabling proxy and gateway proxy are shown in Figures 1.3(a), 1.3(b) and 1.3(c). For the test configuration of the gateway, an additional NAT box is required which performs NAT address translation for source addresses [3]

The test software is executed by

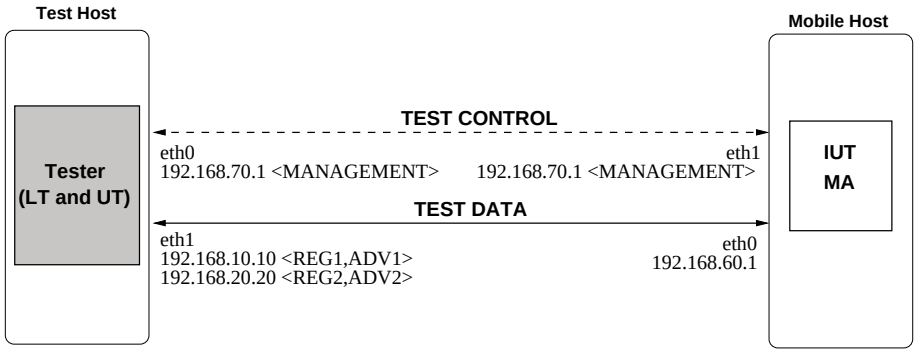
ma_test 0
ma_test 1
mep_test
gwp_test

The parameter for *ma_test* forces the usage of a particular test suite. The parameter '0' corresponds with *ma_test1.conf* (enabled de-registration), and '1' with *ma_test2.conf* (disabled de-registration). The configuration files for the MEP testing and the GW_P testing are *mep_test.conf* and *gwp_test.conf*, respectively. All configuration files are listed in the Appendix A.

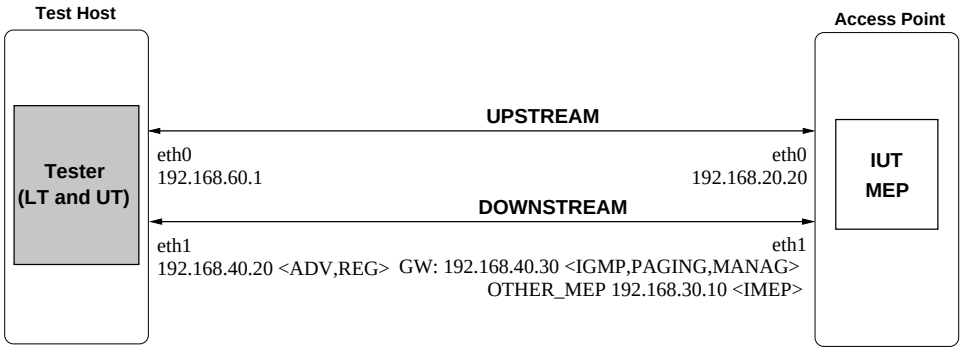
particular state and the input message is applied. After each test the implementation is forced into the initial state. These tests verify that the IUT is capable of correctly performing all state transitions.

[2]The mobile host does not de-registers with MEP1

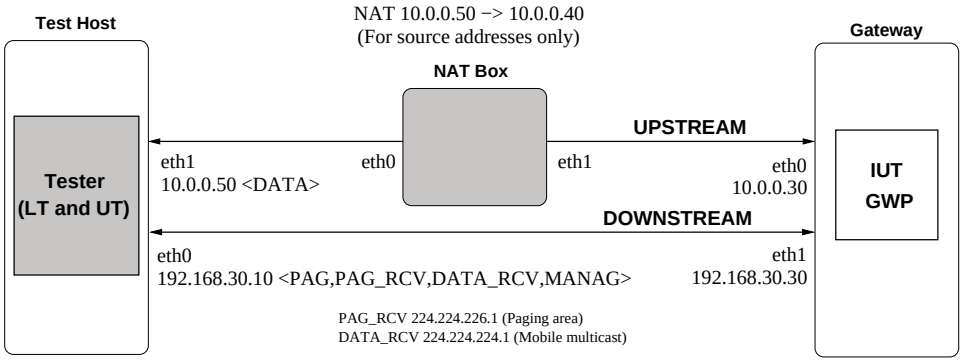
[3]Without network address translation the test host drops received multicast packets. Shortly, this is caused by the RPF (Reverse Path Forwarding) check which avoids forwarding of multicast packets on invalid interfaces.



(a) Mobile Agent



(b) MEP



(c) Gateway

Figure 1.3: Test setup

Chapter 2

Test Cases

2.1 Mobile Agent

2.1.1 State *WAIT4MEP*

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MEP Advertisement ICMP_ROUTERADVERT					
MA-1	The mobile host is in the state WAIT4MEP and no access points are known.	Reception of a valid ICMP_ROUTERADVERT message.	RESTART STATUS ICMP_ROUTERADVERT STATUS	Insertion of access point into the database. Sending a REG_REQ message to this access point.	√
MA-2	The mobile host is in the state WAIT4MEP and no access points are known.	Reception of valid ICMP_ROUTERADVERT message with BUSY flag.	RESTART STATUS ICMP_ROUTERADVERT STATUS	Insertion of the access point into database. NO registration at this access point.	√

2.1.2 State *REG_PENDING*

#	Precondition	Test Description	Test Sequence	Expected Behavior	
Registration reply REG_REPLY					
MA-3	The mobile host is in the state REG_PENDING and a pending registration exists.	Reception of a REG_REPLY message with (REPLY_CODE = REGISTRATION_ACCEPTED).	RESTART ICMP_ROUTERADVERT STATUS REG_REPLY STATUS	The database is updated (the access point is marked as registered) and the mobile host goes into the state ACTIVE.	√
MA-4	The mobile host is in the state REG_PENDING and a pending registration exists.	Reception of a REG_REPLY message, but the REG_REPLY contains the REPLY_CODE = POORLY_FORMED_REQUEST (or another reply code denying the registration).	RESTART ICMP_ROUTERADVERT STATUS REG_REPLY STATUS	The REG_REPLY message is ignored. The MEP is marked as stale.	√
Registration request timeout REGREQ_TO					
MA-5	The mobile host is in the state REG_PENDING and a pending registration exists (MAX_REG_RETRY is set to 3).	Repeated reception of a REGREQ_TO.	RESTART ICMP_ROUTERADVERT STATUS REGREQ_TO STATUS REGREQ_TO STATUS REGREQ_TO STATUS	For the first and second REGREQ_TO the mobile host sends a REG_REQ message to the selected access point. On the fourth reception of the REGREQ_TO the mobile host goes into the state WAIT4MEP without sending a REG_REQ (Giving up). The MEP entry is marked as stale.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MEP advertisement timeout MEP_TO					
MA-6	The mobile host is in the state REG_PENDING and a pending registration exists. There is only a single access point entry in the database.	Reception of a MEP_TO signal.	RESTART ICMP_ROUTERADVERT STATUS MEP_TO STATUS	The access point entry is removed from the database. The mobile host goes into the state WAIT4MEP.	√
MA-7	The mobile host is in the state REG_PENDING and a pending registration exists. There exist two access point entries.	Reception of a MEP_TO signal for the access point with the higher priority.	RESTART ICMP_ROUTERADVERT ICMP_ROUTERADVERT STATUS MEP_TO STATUS	The access point entry with the higher priority is removed from the database. The mobile host sends a REG_REQ message and goes into the state REG_PENDING.	√
MA-8	The mobile host is in the state REG_PENDING and a pending registration exists. There exist two access point entries.	Reception of a MEP_TO signal for the access point with the lower priority.	RESTART ICMP_ROUTERADVERT ICMP_ROUTERADVERT STATUS MEP_TO STATUS	The access point entry with the lower priority is removed from the database.	√
Handoff trigger signal HO_TRIG					
MA-9	The mobile host is in the state REG_PENDING and a pending registration exists.	Reception of a HO_TRIG signal.	RESTART ICMP_ROUTERADVERT STATUS HO_TRIG STATUS	The HO_TRIG signal is ignored.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MA-10	The mobile host is in the state REG_PENDING and a pending registration exists. There exist two access point entries in the database.	Reception of a HO_TRIG signal to handoff to the other access point.	RESTART ICMP_ROUTERADVERT ICMP_ROUTERADVERT STATUS HO_TRIG STATUS	The mobile host sends a REG_REQ message to the access point.	√

2.1.3 State *ACTIVE*

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MEP Advertisement ICMP_ROUTERADVERT					
MA-11	The mobile host is active and registered. A single access point is known.	Reception of a valid ICMP_ROUTERADVERT message. The new access point is less attractive than the old one, according to selection policies.	RESTART ICMP_ROUTERADVERT REG_REPLY STATUS ICMP_ROUTERADVERT STATUS	Insertion of access point into the database.	√
MA-12	The mobile host is active and registered. A single access point is known. De-registration is disabled.	Reception of a valid ICMP_ROUTERADVERT message. The new access point is more attractive than the old one, according to selection rules.	RESTART ICMP_ROUTERADVERT REG_REPLY ICMP_ROUTERADVERT STATUS REG_REPLY	Insertion of access point into database. The mobile host sends a REG_REQ message to the new access point.	√
MA-13	The mobile host is active and registered. A single access point is known. De-registration is enabled.	Reception of a valid ICMP_ROUTERADVERT message. The new access point is more attractive than the old one, according to selection rules.	RESTART ICMP_ROUTERADVERT REG_REPLY STATUS ICMP_ROUTERADVERT STATUS	Insertion of access point into database. The mobile host sends a REG_REQ message to the new access point and a REG_REQ message with lifetime of 0 for de-registration (if De-registration is configured).	√
MA-14	The mobile host is active and registered. A single access point is known.	Reception of a valid ICMP_ROUTERADVERT message for an already known access point.	RESTART ICMP_ROUTERADVERT REG_REPLY STATUS ICMP_ROUTERADVERT STATUS	Update of the database.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
Paging request PAG_REQ					
MA-15	The mobile host is active and registered. A single access point is known.	Reception of a PAG_REQ message.	RESTART ICMP_ROUTERADVERT REG_REPLY STATUS PAG_REQ STATUS	The PAG_REQ message is ignored.	√
MEP advertisement timeout MEP_TO					
MA-16	The mobile host is active and registered and there exists only a single access points in the database.	Lifetime of an ICMP_ROUTERADVERT message expires.	RESTART ICMP_ROUTERADVERT REG_REPLY STATUS MEP_TO STATUS	The access point is removed from the database and the mobile host goes into the state WAIT4MEP.	√
MA-17	The mobile host is active and registered and there exist an entry for the actual and for another access points in the database.	Lifetime of an ICMP_ROUTERADVERT message expires of that MEP the mobile host is registered with.	RESTART ICMP_ROUTERADVERT REG_REPLY ICMP_ROUTERADVERT STATUS MEP_TO STATUS	The entry of the actual access point is removed from the database. The mobile host sends a REG_REQ message to the other MEP and goes into the state REG_PENDING.	√
MA-18	The mobile host is active and registered, and there exists an entry for the actual and an another access points in the database.	Lifetime of an ICMP_ROUTERADVERT message of the other MEP expires.	RESTART ICMP_ROUTERADVERT REG_REPLY ICMP_ROUTERADVERT STATUS MEP_TO STATUS	The access point is removed from the database.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
Re-registration timeout REREG_TO					
MA-19	The mobile host is active and registered.	Reception of a REREG_TO signal.	RESTART ICMP_ROUTERADVERT REG_REPLY STATUS REG_TO STATUS	The mobile host sends a REG_REQ message to the selected access point.	√
Activity timeout ACT_TO					
MA-20	The mobile host is active and registered. De-registration is disabled.	Reception of a ACT_TO signal.	RESTART ICMP_ROUTERADVERT REG_REPLY STATUS ACT_TO STATUS	The mobile host sends a REQ_REQ message (activity flag set to 0) message via the selected access point.	√
Data transport					
MA-21	The mobile host is active and registered.	The mobile host intends to send a data packet (e.g. ICMP echo request).	RESTART ICMP_ROUTERADVERT REG_REPLY STATUS ICMP_REQUEST STATUS	The data packet is sent.	√
Handoff trigger HO_TRIG					
MA-22	The mobile host is active and registered. De-registration is disabled.	Reception of a HO_TRIG signal.	RESTART ICMP_ROUTERADVERT REG_REPLY ICMP_ROUTERADVERT STATUS HO_TRIG STATUS	A REG_REQ message is sent to the new access point. The mobile host remains in the state ACTIVE.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MA-23	The mobile host is active and registered. De-registration is enabled.	Reception of a HO_TRIG signal.	RESTART ICMP_ROUTERADVERT REG_REPLY ICMP_ROUTERADVERT STATUS HO_TRIG STATUS	The mobile host sends a REG_REQ message to the new access point. The mobile host remains in the state ACTIVE. After receiving a REG_REPLY the mobile sends a REG_REQ message with lifetime 0 to the old access point (deregistration).	√
Registration reply REG_REPLY					
MA-24	The mobile host is active and registered.	Reception of a valid REG_REPLY message.	RESTART ICMP_ROUTERADVERT REG_REPLY ICMP_ROUTERADVERT HO_TRIG STATUS REG_REPLY STATUS	The database is updated.	√
MA-25	The mobile host is active, registered and executes a handoff.	Reception of a REG_REPLY message with busy flag set and REPLY_CODE = INSUFFICIENT_RESSOURCES or another invalid reply code.	RESTART ICMP_ROUTERADVERT REG_REPLY ICMP_ROUTERADVERT HO_TRIG STATUS REG_REPLY STATUS	The MEP entry is marked as stale. The mobile host sends a REG_REPLY message to the old access point.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	

2.1.4 State *INACTIVE*

#	Precondition	Test Description	Test Sequence	Expected Behavior	
PAG_REQ					
MA-26	The mobile host is in the state INACTIVE and there exists an access point entry in the database.	Reception of a PAG_REQ message.	RESTART ICMP_ROUTERADVERT REG_REPLY ACT_TO STATUS PAG_REG STATUS	The mobile host sends a REG_REQ message to the selected access point and goes into the state REG_PENDING.	√
Wakeup signal					
MA-27	The mobile host is in the state INACTIVE and there is an access point entry in the database.	Reception of a wakeup signal (e.g. mobile host intends to send an ICMP echo request).	RESTART ICMP_ROUTERADVERT REG_REPLY ACT_TO STATUS ICMP_REQUEST STATUS	The mobile host sends a REG_REQ to the access point and goes into the state REG_PENDING.	√
MEP advertisement timeout MEP_TO					
MA-28	The mobile host is in the state INACTIVE and there is a single access point entry in the database.	Reception of a MEP_TO signal.	RESTART ICMP_ROUTERADVERT REG_REPLY ACT_TO STATUS MEP_TO STATUS	The database is updated.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
Paging update timeout REREG_TO					
MA-29	The mobile host is in the state INACTIVE and there is a single access point entry in the database.	Reception of a REREG_TO signal.	RESTART ICMP_ROUTERADVERT REG_REPLY ACT_TO ICMP_ROUTERADVERT STATUS PAG_TO STATUS	The mobile host sends a REG_REQ message (activity flag set to 0) to the gateway.	√
MA-30	The mobile host is in the state INACTIVE and there are two access point entries in the database.	Reception of a REREG_TO signal.	RESTART ICMP_ROUTERADVERT REG_REPLY ACT_TO ICMP_ROUTERADVERT ICMP_ROUTERADVERT STATUS PAG_TO STATUS	The mobile host sends a REG_REQ message (activity flag set to 0) via the selected gateway.	√

2.2 Mobility Enabling Proxy MEP

2.2.1 State *IDLE*

#	Precondition	Test Description	Test Sequence	Expected Behavior	
Registration request REG_REQ					
MEP-1	There is NO mobile host entry in the database.	Reception of a REG_REQ message with Activity flag set and lifetime larger than 0. The wakeup flag is set.	RESTART STATUS REG_REQ STATUS	The MEP updates its database and sends a PAG_UPD message to the GW_P with a lifetime of 0. Then the MEP subscribes for the multicast channel and sends a REG_REPLY to the mobile host.	√
MEP-2	There is a mobile host entry in the database and marked as directly registered.	Reception of a REG_REQ message with Activity flag set and lifetime larger than 0.	RESTART REG_REQ STATUS REG_REQ STATUS	The MEP updates its database and sends a REG_REPLY to the mobile host.	√
MEP-3	There is a mobile host entry in the database and marked as indirectly registered. A buffer thread for the mobile host exists.	Reception of a REG_REQ message with Activity flag set and lifetime larger than 0.	RESTART IMEP_ADVERT STATUS REG_REQ STATUS	The MEP updates its database and sends a REG_REPLY to the mobile host.	√
MEP-4	There is a mobile host entry in the database and marked as directly registered.	Reception of a REG_REQ message with a lifetime equals 0.	RESTART REG_REQ STATUS REG_REQ STATUS	The MEP removes the mobile host entry from its database and unscubscribes from the multicast channel.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MEP-5	There is a mobile host entry in the database and marked as indirectly registered. A buffer thread exists.	Reception of a REG_REQ message with a lifetime equals 0.	RESTART IMEP_ADVERT STATUS REG_REQ STATUS	The message is ignored.	√
MEP-6	There is NO mobile host entry in the database.	Reception of a REG_REQ message with a lifetime equals 0.	RESTART STATUS REG_REQ STATUS	The REG_REQ message is ignored.	√
MEP-7	There is a mobile host entry in the database.	Reception of a REG_REQ message with Activity flag NOT set.	RESTART REG_REQ STATUS REG_REQ STATUS	The access point removes the mobile host from its database, unsubscribes from the multicast channel and sends a PAG_UPD message to the gateway.	√
MEP-8	There is NO mobile host entry in the database.	Reception of a REG_REQ message with Activity flag NOT set.	RESTART STATUS REG_REQ STATUS	The access point sends a PAG_UPD message to the gateway.	√
Solicitation ICMP_ROUTERSOLICIT					
MEP-9	There is NO mobile host entry in the database.	Reception of an ICMP_ROUTERSOLICIT message.	RESTART STATUS ICMP_ROUTERSOLICIT STATUS	The access point sends a ICMP_ROUTERADVERT message.	√
Inter-MEP advertisement IMEP_ADVERT					
MEP-10	There is NO mobile host entry in the database.	Reception of an IMEP_ADVERT message with a single mobile host entry.	RESTART STATUS IMEP_ADVERT STATUS	The MEP insert the mobile host into the database, subscribes for the multicast channel and starts a buffer thread.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MEP-11	There is a mobile host entry in the database which is registered indirectly, the MEP is subscribed for the corresponding multicast channel and a buffer thread is running.	Reception of an IMEP_ADVERT message from the same MEP with the same mobile host entry.	RESTART IMEP_ADVERT STATUS IMEP_ADVERT STATUS	The database is updated.	√
MEP-12	There is a mobile host entry in the database which is registered directly, the MEP is subscribed for the corresponding multicast channel and a buffer thread is NOT running).	Reception of an IMEP_ADVERT message with a mobile host entry which is already registered directly.	RESTART REG_REPLY STATUS IMEP_ADVERT STATUS	The database is updated.	√
MEP-13	There is a mobile host entry in the database which was indirectly registered).	Reception of an IMEP_ADVERT message without a mobile host entry.	RESTART IMEP_ADVERT STATUS IMEP_ADVERT STATUS	The mobile host entry is removed from the database. The MEP unsubscribes from the multicast channel.	√
MEP-14	There is a mobile host entry in the database which was indirectly registered).	Reception of an IMEP_ADVERT message without mobile host entry from another MEP than before.	RESTART IMEP_ADVERT STATUS IMEP_ADVERT STATUS	The database is updated. The mobile host entry remains in the database.	√
Paging request PAG_REQ					
MEP-15	There is NO mobile host entry in the database.	Reception of a PAG_REQ message.	RESTART STATUS PAG_REQ STATUS	The PAG_REQ message is forwarded.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MEP-16	There is a mobile host entry in the database and the mobile host is marked as directly registered.	Reception of a PAG_REQ message.	RESTART REG_REQ(A_FLAG=TRUE) STATUS PAG_REQ STATUS	The PAG_REQ message is forwarded.	√
MEP-17	There is a mobile host entry in the database and the mobile host is marked as indirectly registered.	Reception of a PAG_REQ message.	RESTART IMEP_ADVERT STATUS PAG_REQ STATUS	The PAG_REQ message is forwarded.	√
MEP advertisement timeout MEP_ADV_TO					
MEP-18	The MEP is in the state IDLE	Reception of a MEP_ADV_TO signal.	RESTART REG_REQ(A_FLAG=TRUE) STATUS MEP_ADV_TO STATUS	The access point sends an ICMP_ROUTERADVERT message.	√
IMEP advertisement timeout IMEP_ADV_TO					
MEP-19	There is a mobile host entry in the database. The mobile host is marked as directly registered. The handoff policy for the mobile is *predictive handoff*.	Reception of a IMEP_ADV_TO signal.	RESTART REG_REQ(A_FLAG=TRUE) STATUS IMEP_ADV_TO STATUS	The access point sends an IMEP_ADVERT message.	√
Registration timeout REG_TO					
MEP-20	There is a mobile host entry in the database, the mobile host is directly registered and marked as *ACTIVE*. The access point is subscribed for the corresponding multicast channel. There exists no indirect registration (via IMEP_ADVERT).	Reception of a REG_TO signal.	RESTART REG_REQ(A_FLAG=TRUE) STATUS REG_TO STATUS	The mobile host entry is removed from the database and the MEP unsubscribes from the multicast channel.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
MEP-21	There is a mobile host entry in the database, the mobile host is directly registered and marked as *ACTIVE*. The access point is subscribed for the corresponding multicast channel. There exists an indirect registration (via IMEP_ADVERT).	Reception of a REG_TO signal.	RESTART REG_REQ(A_FLAG=TRUE) IMEP_ADVERT STATUS REG_TO STATUS	The mobile host entry is updated and the access point does NOT unsubscribe from the multicast channel. A buffer thread is started.	√

2.3 Gateway Proxy GW_P

2.3.1 State *IDLE*

#	Precondition	Test Description	Test Sequence	Expected Behavior	
Paging update message PAG_UPD					
GW_P-1	There is NO mobile host entry in the database and the correspondent multicast channel does not exists. There is no flush thread running.	Reception of a PAG_UPD message with a lifetime larger than 0.	RESTART STATUS PAG_UPD STATUS	The mobile host entry is inserted into the database.	√
GW_P-2	There is NO mobile host entry in the database and the correspondent multicast channel exists. There is no flush thread running.	Reception of a PAG_UPD message with a lifetime larger than 0.	RESTART SUBSCRIBE_MCC STATUS PAG_UPD STATUS	The mobile host entry is inserted into the database.	√
GW_P-3	There is a mobile host entry in the database and the correspondent multicast channel does not exists. There is no flush thread running.	Reception of a PAG_UPD message with a lifetime larger than 0.	RESTART PAG_UPD STATUS PAG_UPD STATUS	The database is updated.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
GW_P-4	There is a mobile host entry in the database and a correspondent multicast channel does not exists. The gateway buffers data for the mobile host.	Reception of a PAG_UPD message with a lifetime larger than 0 (While a paging is ongoing, the mobile host has gone into the state INACTIVE).	RESTART PAG_UPD UDP (triggers paging and starts buffering) STATUS PAG_UPD STATUS	The database is updated (and the gateway proxy still waits for a PAG_UPD with a lifetime of 0.	√
GW_P-5	There is a mobile host entry in the database and a correspondent multicast channel exists. A flush thread is running.	Reception of a PAG_UPD message with a lifetime larger than 0 (While a buffer is flushed, the mobile host has gone into the state INACTIVE).	RESTART PAG_UPD SUBSCRIBE_MCC UDP (triggers paging and starts buffering) PAG_UPD(lifetime=0) (stops buffering and starts a paging thread) STATUS PAG_UPD STATUS	The PAG_UPD message is ignored.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
GW_P-6	There is a mobile host entry in the database and a correspondent multicast channel does not exist. The gateway proxy buffers data for the mobile host.	Reception of a PAG_UPD message with a lifetime 0.	RESTART PAG_UPD UDP (triggers paging and starts buffering) STATUS PAG_UPD STATUS	The database is updated. (The gateway proxy continues buffering and waits for establishment of the multicast channel.	√
GW_P-7	There is a mobile host entry in the database and a correspondent multicast channel exists. The gateway proxy buffers data for the mobile host.	Reception of a PAG_UPD message with a lifetime 0.	RESTART PAG_UPD SUBSCRIBE_MCC UDP (triggers paging and starts buffering) STATUS PAG_UPD STATUS	The mobile host entry is marked as flushing. The gwateway proxy stops buffering and the flush thread is started.	√
Paging timeout PAG_TO					
GW_P-8	There is a mobile host entry in the database, the access point is subscribed for the corresponding multicast channel. A paging is ongoing (e.g. triggered by an UDP packet) and the gateway proxy is buffering packets.	Reception of a PAG_TO signal.	RESTART PAG_UPD SUBSCRIBE_MCC UDP (triggers paging) STATUS PAG_TO STATUS	The gateway proxy stops buffering. Note: This depends strongly on the paging policies. The standard policy does not repage the mobile host.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	
Mobile timeout MOBILE_TO					
GW_P-9	There is a mobile host entry in the database. A flush thread is not running.	Reception of a MOBILE_TO signal.	RESTART PAG_UPD STATUS MOBILE_TO STATUS	The mobile host entry is removed from the database.	√
GW_P-10	There is a mobile host entry in the database. There exists a flush thread.	Reception of a MOBILE_TO signal.	RESTART PAG_UPD SUBSCRIBE_MCC UDP (triggers paging) STATUS MOBILE_TO STATUS	The deletion of the mobile host entry is delayed.	√
GW_P-11	There is a mobile host entry in the database. The gateway proxy buffers data. No flush thread is running.	Reception of a MOBILE_TO signal.	RESTART PAG_UPD UDP (triggers paging and starts buffering) STATUS MOBILE_TO	The deletion of the mobile host entry is delayed.	√
Data transport					
GW_P12	There is NO mobile host entry in the database and a multicast channel exists. There is no flush thread running.	Reception of an UDP packet.	RESTART PAG_UPD STATUS UDP (triggers a paging and starts buffering) SUBSCRIBE_MCC PAG_UPD(lifetime=0) (stops buffering and starts the flush thread) STATUS	The data packet is sent towards the mobile host.	√

#	Precondition	Test Description	Test Sequence	Expected Behavior	

Appendix A

Configuration files

A.1 Mobile Agent

A.1.1 *ma_test1.conf* (with de-registration)

```
Interface: eth0, 0, 60, 120, 1, no, 192.168.60.1

RegPort: 434  # default is 434
IdleTimeout: 30
Predictive: yes
ManagementPort: 8888
```

A.1.2 *ma_test2.conf* (without de-registration)

```
Interface: eth0, 0, 60, 120, 1, no, 192.168.60.1

RegPort: 434  # default is 434
IdleTimeout: 30
Predictive: yes
ManagementPort: 8888
```

A.2 Mobility Enabling Proxy *mep_test.conf*

```
Upstream: eth1, 20, 40
Downstream: eth0, 20000, 40000

Gateway: 192.168.40.30
PagingPort: 434
RegPort: 434
IMEPport: 434

NATrange: 192.168.60.0/24, 224.224.224.0/24, NAT

MaxMobiles: 20, 20
MaxRegtime: 65535 # unlimited

MEPgroups: 224.224.225.2, 192.168.30.10/224.224.225.1

# PagingAreas: primary, secondary group ...
PagingAreas: 224.224.226.1, 224.224.226.2

ManagementPort: 8888
```

A.3 GW_P *gwp_test.conf*

```
Upstream: eth0
Downstream: eth1

PagingPort: 434

NATrange: 192.168.60.0/24, 224.224.224.0/24, NAT

MaxMobiles: 100

MaxCachetime: 65535 # unlimited

PagingAreas: 224.224.226.0/24

PagingSource: 172.16.0.2

ManagementPort: 8888
```

Bibliography

[1] J. Ellsberger, D. Hogrefe, and A. Sarma. *SDL Formal Object-oriented Language for Communication Systems*. Number ISBN 0-13-621384. Prentice Hall, 1997.

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

[3] A. Festag and L. Westerhoff. Protocol Specification of the MOMBASA Software Environment. Technical Report TKN-01-014, TKN, TU Berlin, Berlin, Germany, May 2001.

[4] G.J. Holzmann. *Design and Validation of Computer Protocols*. Number ISBN 0-13539925-4. Prentice Hall, 1991.

[5] L. Westerhoff and A. Festag. Implementation of the MOMBASA Software Environment. Download at `http://www-tkn.ee.tu-berlin.de/research/mombasa/mse.html`.

[6] L. Westerhoff and A. Festag. Implementation Design of the MOMBASA Software Environment. Technical Report TKN-01-017, TKN, TU Berlin, Berlin, Germany, November 2001.