3GPP2 C.S0091-0

Version 1.0

Version Date: August 29, 2008

Signaling Conformance Specification for Ultra Mobile Broadband Air Interface

No Text.

CONTENTS

CONTENTS

CONTENTS

CONTENTS

(This foreword is not part of this Standard)

This standard was prepared by Technical Specification Group C of the Third Generation Partnership Project 2 (3GPP2). This standard is a companion to the Ultra Mobile Broadband™ (UMB™)[1] air interface standards. This specification provides a set of procedures that the access terminal and the access network can use to conduct the signaling conformance tests in a laboratory environment.

[1] Ultra Mobile Broadband™ and (UMB™) are trade and service marks owned by the CDMA Development Group (CDG).

The following standards and documents contain provisions, which, through reference in this text, constitute provisions of this specification. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

Normative References

[1] *C.S0084-000-0, Overview for Ultra Mobile Broadband (UMB) Air Interface Specification.*

[2] *C.S0084-001-0, Physical Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.*

[3] *C.S0084-002-0, MAC Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.*

[4] C.S0084-003-0, *Radio Link Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.*

[5] *C.S0084-004-0, Application Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.*

[6] *C.S0084-005-0, Security Functions for Ultra Mobile Broadband (UMB) Air Interface Specification.*

[7] *C.S0084-006-0, Connection Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification.*

[8] C.S0084-007-0, *Session Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification.*

[9] C.S0084-008-0, *Route Control Plane for Ultra Mobile Broadband (UMB) Air Interface Specification.*

REFERENCES

Informative References

[10] C.S0084-009-0, Broadcast-Multicast Upper Layer for Ultra Mobile Broadband (UMB) Air Interface Specification.

[11] C.R1001, Administration of Parameter Value Assignments for cdma2000 Spread Spectrum Standards, Release E.

No Text.

1 OVERVIEW

1.1 Scope of This Document

These technical requirements form a standard for signaling conformance in UMB data systems. These requirements ensure that compliant access terminals and compliant access networks can execute tests in meeting the objectives stated in 1.2.

1.2 Objectives

The objective of the test procedures contained herein is to demonstrate that access terminal/access network implementation of signaling functionality in a cabled test environment at nominal levels is in conformance with [1]-[9].

1.3 Requirements Language

"Shall" and "shall not" identify requirements to be followed strictly to conform to the standard and from which no deviation is permitted. "Should" and "should not" indicate that one of several possibilities is recommended as particularly suitable, without mentioning or excluding others, that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. "May" and "need not" indicate a course of action permissible within the limits of the standard. "Can" and "cannot" are used for statements of possibility and capability, whether material, physical, or causal.

1.4 Document Organization

This document is organized into the following sections:

- Section 1 Overview: This section describes the document scope and objectives as well as document organization, list of acronyms and notations.
- Section 2 Basic QoS Management Tests: This section describes the test setups, procedures and minimum requirements for the Basic QoS Management Protocol (BQMP) tests.
- Section 3 Basic Radio Link Protocol Tests: This section describes the test setups, procedures and minimum requirements for the Basic Radio Link Protocol (BRLP) tests.
- Section 4 Basic Route Protocol Tests: This section describes the test setups, procedures and minimum requirements for the Basic Route Protocol tests.
- Section 5 Basic Session Control Protocol Tests: This section describes the test setups, procedures and minimum requirements for the Basic Session Control Protocol (BSCP) tests.
- Section 6 Basic Route Control Protocol Tests: This section describes the test setups, procedures and minimum requirements for the Basic Route Control Protocol (BRCP) tests.

- Section 7 Basic Connection Control Protocol Tests: This section describes the test setups, procedures and minimum requirements for the Basic Air-Link Management Protocol, Basic Initialization State Protocol, Basic Idle State Protocol, Basic Connected State Protocol, Basic Active Set Management Protocol, and Overhead Message Protocol (BCCP) tests.
- Section 8 Basic Security Functions Tests: This section describes the test setups, procedures and minimum requirements for the Basic Key Exchange Protocol (BKEP) tests.
- Section 9 Figures: This section includes various figures used in the document.

Protocols belonging to [5] are tested implicitly through various tests in this specification. Some tests for protocols specified in [2], [3] and [10] may be included in future versions of this document.

1.5 Abbreviations, Acronyms and Terms

Access Network (AN). The network equipment providing data connectivity between a packet switched data network (typically the Internet) and the access terminals.

ASP. Active Set Pilot.

Access Terminal (AT). A device providing data connectivity to a user. An access terminal may be connected to a computing device such as a laptop personal computer or it may be a self-contained data device such as a personal digital assistant.

BCMCS. Broadcast and Multicast services.

Channel. The set of channels transmitted between the access network and the access terminals within a given frequency assignment. A Channel consists of a Forward Link and a Reverse Link.

Forward Channel. The portion of the Channel consisting of those Physical Layer Channels transmitted from the access network to the access terminal.

Forward Control Channel. The channel that carries data to be received by all access terminals monitoring the Forward Channel.

Forward MAC Channel. The portion of the Forward Channel dedicated to Medium Access Control activities. The Forward MAC Channel consists of the RPC, DRCLock, and RA Channels.

Forward Traffic Channel. The portion of the Forward Channel that carries information for a specific access terminal. The Forward Traffic Channel can be used as either a Dedicated Resource or a non-Dedicated Resource. Prior to successful access terminal authentication, the Forward Traffic Channel serves as a non-Dedicated Resource. Only after successful access terminal authentication can the Forward Traffic Channel be used as a Dedicated Resource for the specific access terminal.

FTP. File Transfer Protocol.

MAC Layer. The MAC Layer defines the procedures used to receive and to transmit over the Physical Layer.

NA. Not Applicable.

Physical Layer Protocol. The Physical Layer Protocol provides the channel structure, frequency, power output, modulation, and encoding specifications for the forward and reverse links.

RATI. Random Access Terminal Identifier.

Reservation. Air interface resources set up by the access network to carry a higher layer flow. A Reservation is identified by its ReservationLabel. ReservationLabels are bound to RLP Flows that carry higher layer flows. A Reservation can be either in the Open or Close state.

Reverse Access Channel. The portion of the Reverse Channel that is used by access terminals to communicate with the access network when they do not have a traffic channel assigned. There is a separate Reverse Access Channel for each sector of the access network.

Reverse Channel. The portion of the Channel consisting of those Physical Layer Channels transmitted from the access terminal to the access network.

Reverse Traffic Ack Channel. The portion of the Reverse Traffic Channel that indicates the success or failure of the Forward Traffic Channel reception.

Reverse Traffic Channel. The portion of the Reverse Channel that carries information from a specific access terminal to the access network. The Reverse Traffic Channel can be used as either a Dedicated Resource or a non-Dedicated Resource. Prior to successful access terminal authentication, the Reverse Traffic Channel serves as a non-Dedicated Resource. Only after successful access terminal authentication can the Reverse Traffic Channel be used as a Dedicated Resource for the specific access terminal.

Reverse Traffic Data Channel. The portion of the Reverse Traffic Channel that carries user data.

Reverse Traffic MAC Channel. The portion of the Reverse Traffic Channel dedicated to Medium Access Control activities. The Reverse Traffic MAC Channel consists of the RRI and DRC Channels.

RLP. Radio Link Protocol provides retransmission and duplicate detection for an octet-aligned data stream.

Rx. Receive.

Sector. The part of the access network that is identified by (SectorID, CDMA Channel).

Security Layer. The Security Layer provides authentication and encryption services.
Session Layer. The Session Layer provides protocol negotiation, protocol configuration, and state maintenance services.

Subnet Mask (of length *n*). A 128-bit value whose binary representation consists of n consecutive '1's followed by 128-n consecutive '0's.

Tx. Transmit.

UATI. Unicast Access Terminal Identifier.

Universal Coordinated Time (UTC). An internationally agreed-upon time scale maintained by the Bureau International de l'Heure (BIH) used as the time reference by nearly all commonly available time and frequency distribution systems.

UTC. Universal Temps Coordine. See Universal Coordinated Time.

ms. Millisecond

s. Second

1.6 Notation

$\lfloor x \rfloor$	Indicates the largest integer less than or equal to x: $\lfloor 1.1 \rfloor = 1, \lfloor 1.0 \rfloor = 1$.
x mod y	Indicates the remainder after dividing x by y: x mod y = $x - (y \times \lfloor x/y \rfloor)$.

2 BASIC QOS MANAGEMENT PROTOCOL TESTS

The Basic QoS Management Protocol (BQMP) of the Radio Link Layer (RLL) of UMB provides packet filtering and QoS for IP packets. It also provides a mapping of reservations to streams. Applications that require QoS may reside at the AT or the TE. The interface between the applications and the RLL is implementation dependent and is beyond the scope of this specification[2]. The application should be able to connect with the server/peer and receive and transmit data. This may require password authentication when using commercial applications residing on the AT.

Unless otherwise specified, for tests in this section, ProfileType field of Reservation*KK*QoSRequestFwd or Reservation*KK*QoSRequestRev attributes shall be set to a value other than 0x00.

For all test procedures in this section, ReservationLabel 0xXY refers to any acceptable ReservationLabel other than 0xfe. Once negotiated, the ReservationLabel 0xXY will remain constant for the duration of the test. The AT and AN negotiate ReservationLabel 0xXY to be bound to stream S on both the reverse and forward links through Reservation*XY*StreamRev and Reservation*XY*StreamFwd attributes respectively, where S is different from the stream to which ReservationLabel 0xfe is bound. Stream S is activated on the forward link if the value of the attribute ActivatedFwd is 0x01. Similarly, Stream S is activated on the reverse link if the value of the attribute ActivatedRev is 0x01.

Unless otherwise stated,

a) Test procedure steps should be executed in such way that the AT does not transition to idle.

b) When turning a reservation off, it should be ensured that the AN or the AT do not negotiate Reservation*XY*StreamFwd, Reservation*XY*StreamRev, Reservation*XY*PacketFiltersFwd or Reservation*XY*PacketFiltersRev to their default values.

2.1 Basic QoS Management Protocol Access Terminal Tests

[2]Typically, the interface allows the application to start and stop QoS. Starting of application will cause the AT to transmit QoS requests for resource allocation from the AN. Similarly, stopping of the application will cause the AT to transmit request for releasing the resources reserved for the application. The application will provide packet filter information to the AT. The AT will inform the application of any changes in the QoS. It is the responsibility of the application to ensure that the reservation is in open state before it generates any data. Otherwise, the data may not achieve QoS at the Air Interface.

2.1.1 AT Protocol Initialization for InUse Instance

2.1.1.1 Definition

When the InUse instance of the BQMP is initialized, except for the ReservationLabel 0xfe, the AT initializes the reservation state for all reservations to close state. The ReservationLabel 0xfe is initialized to open state. This is verified in step c in part 1 of the test. If the reservation is open then the AT transmits data for this reservation using the stream to which the reservation is bound or through stream 0xfe. This is verified in step e and e in part 2 of the test. Further, when the AT goes dormant, the reservation state for all reservations except ReservationLabel 0xfe transitions to close state. This is verified in step g in part 3 of the test.

QoS Application:

The QoS application needs to generate data before the AT configures the QoS at the air-interface. This may not be the typical behavior for the application.

Traceability:

[4]

Section 2.3.1	Definitions
Section 2.3.2.1	Initialization procedures for the InUse Instance of the Protocol
Section 2.3.2.4.3.1	Reservation Open State requirements for AT
Section 2.3.2.4.3.2	Reservation Open State requirements for AN

2.1.1.2 Test Procedure

a. Instruct the AN to terminate existing session and cause the AT to negotiate a new session with the AN. Cause the AT and AN to negotiate ReservationLabel 0xXY to be bound to stream S.

Part 1

b. Cause the AT to establish a connection with the AN and cause a data source to generate data that does not match any open reservation, i.e. data for ReservationLabel 0xfe, for transmission from the AT. Note, the data source need not be the same as the one that generated the QoS Request.

c. Verify that the AT transmits data using stream bound to ReservationLabel 0xfe without transmitting *ReservationOnRequest* message(s).

Part 2

d. Cause the AT to transmit a *ReservationOnRequest* message(s) requesting forward and reverse link reservation for on ReservationLabel 0xXY. Instruct the AN to accept the request and transmits a *ReservationOnAccept* message(s).

e. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT transmits the packet using the stream S.

Part 3

f. Allow the connection to go dormant.

g. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT either does not transmit this packet using any stream other than the one bound to ReservationLabel 0xfe or transmits the packet using stream S after opening the reservation for ReservationLabel 0xXY.

2.1.1.3 Minimum Standard

AT shall comply with step c, e and g.

2.1.2 Stream binding at the AT

2.1.2.1 Definition

This test verifies the ability of the AT transmits data for a Reservation using appropriate stream under various conditions. Step d ensures that the AT transmits data for reservation using the stream to which the reservation is mapped or through the stream bound to reservation 0xfe. Steps f and g in part 1 of test verify the AT behavior when the stream associated with the reservation for which packet is to be transmitted is deactivated. Steps j and k in part 2 of the test verify the AT behavior when the reservation for which data needs to be transmitted is not mapped to any stream.

AN Control Requirement:

This test requires the AN to deactivate a stream while data is being transferred using this stream. It also requires to set the value of Reservation*XY*StreamRev to the default, i.e. remove the reservation to stream mapping while the data is being transferred using this stream.

QoS Application Requirement:

This test requires the application to continue to generate the data even when the AT indicates that it no longer has the QoS granted on the air-interface. This may not be the typical application behavior.

Traceability:

[4]

Section 2.3.1 Definitions

Section 2.3.2.4.1 Reservation State Maintenance

2.1.2.2 Test Procedure

a. Instruct the AN to terminate existing session and cause the AT to negotiate a new session with the AN. Cause the AT and AN to negotiate ReservationLabel 0xXY to be bound to stream S.

b. Cause the AT to establish a connection with the AN.

c. Cause the AT to request forward and reverse reservations for ReservationLabel 0xXY by transmitting *ReservationOnRequest* message(s). Ensure that the stream S is activated for forward and reverse links.

d. Ensure that no *XonRequest* was transmitted by the AT and that the AT transmits the packets belonging to ReservationLabel 0xXY using the stream S or using stream bound to ReservationLabel 0xfe.

Part 1

e. Cause the AN to negotiate a value of 0x00 for the attributes ActivatedFwd and ActivatedRev for the stream S.

f. Verify that the AT transmits data for ReservationLabel 0xXY using either the stream bound to ReservationLabel 0xfe or the stream S after activating it.

g. Verify that the AT does not transmit data for ReservationLabel 0xXY using any other stream than the one bound to ReservationLabel 0xfe or stream S after activating it.

Part 2

h. If the stream S is not activated, cause the AN to activate stream S in the forward and reverse directions.

i. Cause the AN to negotiate a value of 0xff for Reservation*XY*StreamRev.

j. Verify that the AT transmits data for ReservationLabel 0xXY either using the stream bound to ReservationLabel 0xfe or the steam S after negotiating ReservationLabel 0xXY to be bound to stream S through Reservation*XY*StreamRev attribute.

k. Verify that the AT does not transmit data for ReservationLabel 0xXY using any other stream than the one bound to ReservationLabel 0xfe or the steam S after negotiating ReservationLabel 0xXY to be bound to stream S.

2.1.2.3 Minimum Standard

AT shall comply with step d, g and k.

AT should comply with steps f and j.

2.1.3 Reservation*KK*RoRAllowedFwd/Rev Negotiation be AT

2.1.3.1 Definition

When Reservation*KK*RoRAllowedFwd is set to 0x00, then the AT is not allowed to include forward link reservation request for this reservation. Similarly, when Reservation*KK*RoRAllowedRev is set to 0x00, then the AT is not allowed to transmit reverse link reservation request for this reservation. The application used in the test procedure is assumed to generate forward and reverse link QoS requests, and the AT will transmit these unless prohibited by the Reservation*KK*RoRAllowedFwd/Reservation*KK*RoRAllowedRev attribute.

AN Control Requirement:

The AN needs to control the negotiation of Reservation*KK*RoRAllowedFwd and Reservation*KK*RoRAllowedRev attribute.

Traceability:

[4]

Section 2.3.2.4.1.1.2 State-Independent Requirements for AT

Section 2.5.1 Simple Attributes

2.1.3.2 Test Procedure

a. Configure the AN to use default values for Reservation*KK*RoRAllowedFwd and Reservation*KK*RoRAllowedRev for all reservations.

b. Start an application that requires QoS on forward and reverse link. Ensure that the AT transmits a *ReservationOnRequest* message requesting forward and reverse link reservations for the application.

c. Configure the AN to negotiate Reservation*KK*RoRAllowedFwd to 0x00 for all reservation requests. Note, Reservation*KK*RoRAllowedRev is set to its default value of 0x01.

d. Instruct the AN to terminate existing session and cause the AT to negotiate a new session with the AN.

e. Cause the AT to establish a connection with the AN.

f. Start the application that was started in step b above.

g. Verify that the AT does not transmit *ReservationOnRequest* message requesting forward link reservation. Note, the AT may transmit *ReservationOnRequest* requesting reverse link reservation.

h. Cause the AN to negotiate Reservation*KK*RoRAllowedRev to 0x00 for all reservation requests. Note, Reservation*KK*RoRAllowedFwd is set to its default value of 0x01.

i. Cause the AT to establish a connection with the AN.

j. Start the application that was started in step b above.

k. Verify that the AT does not transmit *ReservationOnRequest* message requesting reverse link reservation. Note, the AT may transmit *ReservationOnRequest* requesting forward link reservation.

2.1.3.3 Minimum Standard

AT shall comply with steps g and k.

2.1.4 Reservation Closed State Requirements for the AT

2.1.4.1 Definition

If the reservation is closed, then the AT will not transmit packets for this reservation using any other stream than the one bound to ReservationLabel 0xfe. Upon receiving a

ReservationOnAccept for a reverse link reservation the AT will transition the reservation to open state. Upon receiving a *RevReservationOn* message, the AT will transition the reservation contained in the message to open state.

Part 1 of the test verifies processing of *ReservationAccept* message at the AT.

Part 2 of the test verifies the processing of *RevReservationOn* message at the AT.

AN Control Requirement:

This test requires the AN to generate a *RevReservationOn* message.

QoS Application Requirement:

The test requires the application generating data at the AT to submit packets to the AT without waiting for the AT to activate the reservation on the air-interface. This may not be the typical behavior.

Traceability:

[4]

Section 2.3.2.4.2.1	Closed State Requirements for AT
Section 2.3.3.1	*ReservationOnRequest*
Section 2.3.3.3	*ReservationOnAccept*
Section 2.3.3.6	*RevReservationOn*

2.1.4.2 Test Procedure

a. Instruct the AN to terminate existing session and cause the AT to negotiate a new session with the AN.

b. Cause the AT to establish a connection with the AN.

c. Ensure that the AT transmits a *ConfigurationRequest* message(s) negotiating Reservation*XY*QoSRequestFwd and Reservation*XY*QoSRequestRev attributes for ReservationLabel 0xXY.

d. Ensure that the AN and AT negotiate Reservation*XY*StreamFwd and Reservation*XY*StreamRev attributes of ReservationLabel 0xXY to a stream S. Ensure that the AN and AT negotiate Reservation*XY*PacketFiltersFwd and Reservation*XY*PacketFiltersRev attributes for the ReservationLabel 0xXY.

Part 1

e. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT.

f. Verify that the AT does not transmit the packet using any other stream than the stream bound to ReservationLabel 0xfe.

g. If the AT has not transmitted a *ReservationOnRequest* message due to step e, cause the AT to transmit a *ReservationOnRequest* message for ReservationLabel 0xXY.

Ensure that the AN transmits a *ReservationAccept message* in response to the *ReservationOnRequest* message.

h. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT transmits the packet using the stream S.

Part 2

i. If the reservation is in open state, cause the AT to transmit *ReservationOffRequest* message for this reservation to the AN. Ensure that the AT receives a *ReservationAccept message* in response.

j. Cause the AN to transmit a *RevReservationOn* message for the reservation 0xXY to the AT.

k. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT transmits the packet using the stream S without transmitting a *ReservationOnRequest* message for ReservationLabel 0xXY.

2.1.4.3 Minimum Standard

AT shall comply with steps h and k.

AT should comply with f.

2.1.5 Reservation Open State Requirements for the AT

2.1.5.1 Definition

If the reservation is open state the AT will transition to closed state upon receiving *ReservationOffAccept* message in response to a *ReservationOffRequest* message or *RevReservationOff* message. The AT may retransmit a *ReservationOffRequest* message if it does not receive a response within $T_{QMPResponse}$ of sending the *ReservationOffRequest*. The AT will continue to maintain the reservation in open state if it receives a *ReservationReject* message.

AT will transition all reservations other than 0xfe to close state upon entering idle state.

The AT may send *ReservationOnRequest* message to request change in the granted QoS of a Reservation. If the AT does not receive a response within $T_{QMPResponse}$ of sending the *ReservationOnRequest*, it may retransmit the request. If the *ReservationOnRequest* message contains a Reverse Reservation, then the access terminal will transition the Reservation to the Close state when the access terminal receives a *ReservationReject* message.

Part 1 of the test verifies processing of a *ReservationOffAccept* message at the AT.

Part 2 of the test verifies processing of a *RevReservationOff* message at the AT.

Part 3 of the test verifies that the AT does not retransmit *ReservationOffRequest* message within $T_{QMPResponse}$ of sending a *ReservationOffRequest*.

Part 4 of the test verifies that the AT turns a reservation (≠ 0xfe) off when the connection goes idle.

Part 5 of the test verifies that the AT can successfully change the QoS_ATTRIBUTE_SET_ID for a reservation in open state.

Part 6 of the test verifies that if the AT's request for change in QoS_ATTRIBUTE_SET_ID is denied by the AN, the AT closes the reservation.

AN Control Requirement:

This test requires the AN to generate a *RevReservationOff and ReservationReject* messages. It is also required to transmit *ReservationReject* message in response to certain *ReservationOnRequest* message.

QoS Application Requirements:

The test requires the application generating data at the AT to submit packets to the AT even though the reservation is turned off. It also requires the Application to trigger *ReservationOffRequest* at the air-interface, while it is still generating data. The application is also required to cause the AT to transmit ReservationOnRequest with a different QoS_ATTRIBUTE_SET_ID. This may not be the typical behavior.

Traceability:

[4]

Section 2.3.2.4.3.1	Open State Requirements for AT
Section 2.3.3.2	*ReservationOffRequest*
Section 2.3.3.4	*ReservationOffAccept*
Section 2.3.3.5	*ReservationReject*
Section 2.3.3.7	*RevReservationOff*

2.1.5.2 Test Procedure

a. Instruct the AN to terminate existing session and cause the AT to negotiate a new session with the AN.

b. Cause the AT to establish a connection with the AN.

c. Ensure that the AT transmits a *ConfigurationRequest* message(s) negotiating Reservation*XY*QoSRequestFwd and Reservation*XY*QoSRequestRev with multiple QoS_ATTRIBUTE_SET_ID attributes for ReservationLabel 0xXY.

d. Ensure that the AN and AT negotiate Reservation*XY*StreamFwd and Reservation*XY*StreamRev attributes of ReservationLabel 0xXY to stream S. Ensure that the AN and AT negotiate Reservation*XY*PacketFiltersFwd and Reservation*XY*PacketFiltersRev attributes for the ReservationLabel 0xXY.

e. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Ensure that the AT does not

transmit the packet using any other stream than the one bound to ReservationLabel.

f. If the reservation 0xXY is in closed state, cause the AT to transmit a *ReservationOnRequest* message for the ReservationLabel 0xXY. Ensure that the AN responds with a *ReservationOnAccept* message.

g. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Ensure that the AT transmits the packet using stream S.

Part 1

h. Cause the AT to transmit a *ReservationOffRequest* message for ReservationLabel 0xXY. Ensure that the AN transmits a *ReservationOffAccept* message in response.

i. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT does not transmit the packet using any other stream than the one bound to ReservationLabel 0xfe.

Part 2

j. If the reservation 0xXY is in a closed state, cause the AT to transmit a *ReservationOnRequest* message for the ReservationLabel 0xXY. Ensure that the AN responds with a *ReservationOnAccept* message.

k. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Ensure that the AT transmits the packet using stream S.

l. Cause the AN to transmit a *RevReservationOff* message for the ReservationLabel 0xXY.

m. After the *RevReservationOff* message is received at the AT, cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT does not transmit the packet using any other stream than the one bound to ReservationLabel 0xfe or using stream S after turning on the reservation 0xXY.

Part 3

n. If the reservation 0xXY is in a closed state, cause the AT to transmit a *ReservationOnRequest* message for the ReservationLabel 0xXY. Ensure that the AN responds with a *ReservationOnAccept* message.

o. Cause the AT to transmit a *ReservationOffRequest* message for ReservationLabel 0xXY. Cause the AN to not respond to the *ReservationOffRequest* message.

p. Verify that the AT does not retransmit *ReservationOffRequest* message within $T_{QMPResponse}$ of sending the *ReservationOffRequest*.

q. Before the AT receives a *ReservationOffAccept* message, Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT transmits the packet using stream S.

Part 4

r. If the reservation 0xXY is in a closed state, cause the AT to transmit a *ReservationOnRequest* message for the ReservationLabel 0xXY. Ensure that the AN responds with a *ReservationOnAccept* message.

s. Allow the connection to go Idle.

t. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT does not transmit the packet using any other stream than the one bound to ReservationLabel 0xfe.

Part 5

u. Allow the connection to go Idle.

v. Cause the AT to transmit a *ReservationOnRequest* message for the ReservationLabel 0xXY. Ensure that the AN responds with a *ReservationOnAccept* message.

w. Cause the AT to transmit a *ReservationOnRequest* message for the ReservationLabel 0xXY requesting a different QoS_ATTRIBUTE_SET_ID for the reverse reservation. Instruct the AN to accept the new value of QoS_ATTRIBUTE_SET_ID and respond with a *ReservationOnAccept* message.

x. Cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT and verify that the AT transmits the packet using stream S.

Part 6

y. Cause the AT to transmit a *ReservationOnRequest* message for the ReservationLabel 0xXY requesting a different QoS_ATTRIBUTE_SET_ID for the reverse reservation. Instruct the AN to respond to the *ReservationOnRequest* with a *ReservationReject* message.

z. After the AT receives the *ReservationReject* message, cause a data source to generate a packet that matches the packet filter for ReservationLabel 0xXY for transmission at the AT. Verify that the AT does not transmit the packet using any other stream than the one bound to ReservationLabel 0xfe.

2.1.5.3 Minimum Standard

AT shall comply with steps q, t, x and z.

AT should comply with steps i, m, and p.

2.2 Basic QoS Management Protocol Access Network Tests

2.2.1 ANSupportedQoSProfiles Negotiation

2.2.1.1 Definition

During session configuration, if necessary, the AN may initiate the negotiation of ANSupportedQoSProfiles and includes ProfileValue with ProfileType equal to 0x04. The test verifies this functionality.

Traceability:

[4]

Section 2.2.2 Procedures for the InConfiguration Instance of the Protocol

Section 2.5.2.1 ANSupportedQoSProfiles Attribute

[11]

Chapter 13 Flow Profile Identifier Assignments

2.2.1.2 Test Procedure

a. Configure the AN and the AT to support multiple QoS profiles with ProfileType equal to 0x04.

b. Instruct the AN to terminate existing session and cause the AT to negotiate a new session with the AN.

c. If the AN initiates the negotiation of ANSupportedQoSProfiles attribute, verify that the AN sets the fields of the ANSupportedQoSProfiles attribute as follows

    1. QoSProfileCount set to the number of ProfileValues included in this message

    QoSProfileCount occurrences of the following fields

    2. ProfileType set to 0x04

    3. ProfileLength set to length of ProfileValue in octets

    4. ProfileValue is set to FlowProfileID

2.2.1.3 Minimum Standard

AN shall comply with step c.

2.2.2 AN Protocol Initialization for InUse Instance

2.2.2.1 Definition

This test verifies the initialization of the reservation at the AN and is similar to test 2.1.1.

Traceability:

[4]

Section 2.3.1 Definitions

Section 2.3.2.1 Initialization procedures for the InUse Instance of the Protocol

Section 2.3.2.4.3.1 Reservation Open State requirements for AT

Section 2.3.2.4.3.2 Reservation Open State requirements for AN

2.2.2.2 Test Procedure

a. Instruct the AT or the AN to terminate existing session and cause the AT to negotiate a new session with the AN. Cause the AT and AN to negotiate ReservationLabel 0xXY to be bound to stream S.

b. Cause the AT to establish a connection with the AN.

Part 1

c. Cause a data source to generate a packet for transmission from the AN that does not match any open reservation, i.e. data for ReservationLabel 0xfe.

d. Verify that the AN transmits data using stream bound to ReservationLabel 0xfe without transmitting *FwdReservationOn* message to the AT.

Part 2

e. Cause the AT to transmit a *ReservationOnRequest* message(s) requesting forward and reverse link reservation for on ReservationLabel 0xXY. Cause the AN to accept the request and transmits a *ReservationOnAccept* message(s).

f. Cause a data source to generate a packet for transmission from the AN that matches the packet filter for ReservationLabel 0xXY. Verify that the AN transmits the packet using the stream S or the stream bound to ReservationLabel 0xfe.

g. Verify that the AN does not transmit the packet generated in step e using any stream other than the stream S or the stream bound to ReservationLabel 0xfe.

Part 3

h. Allow the connection to go dormant.

i. Cause a data source to generate a packet for transmission from the AN that matches the packet filter for ReservationLabel 0xXY. Verify that the AN either does not transmit this packet using any stream other than the one bound to ReservationLabel 0xfe or transmits the packet using stream S after opening the reservation for ReservationLabel 0xXY.

2.2.2.3 Minimum Standard

AN shall comply with steps d, g, and i.

AN should comply with step f.

2.2.3 Stream binding at the AN

2.2.3.1 Definition

This test verifies that the AN transmits data for a reservation using the stream to which the reservation is mapped or through the stream bound to reservation 0xfe.

Traceability:

[4]

Section 2.3.1	Definitions
Section 2.3.2.4.1	Reservation State Maintenance
Section 2.3.2.4.1.1.3	AN State-Independent Requirements

2.2.3.2 Test Procedure

a. Instruct the AN to terminate existing session and cause the AT to negotiate a new session with the AN. Cause the AT and AN to negotiate ReservationLabel 0xXY to be bound to stream S.

b. Cause the AT to establish a connection with the AN.

c. Cause the AT to request forward and reverse reservations for ReservationLabel 0xXY by transmitting *ReservationOnRequest* message(s). Ensure that the stream S is activated for forward and reverse links.

d. Ensure that no *XonRequest* was transmitted by the AT and verify that the AN transmits the packets belonging to ReservationLabel 0xXY using the stream S or using stream bound to ReservationLabel 0xfe.

2.2.3.3 Minimum Standard

AN shall comply with step d.

2.2.4 Reservation Closed State Requirements for the AN

2.2.4.1 Definition

This test verifies that the AN turns on a reservation after transmitting *ReservationOnAccept* message.

Traceability:

[4]

Section 2.3.2.4.1	Reservation State Maintenance
Section 2.3.2.4.1.1.1	Packet Filter Processing Requirements
Section 2.3.2.4.2.2	Closed State Requirements for AN
Section 2.3.3.1	*ReservationOnRequest*
Section 2.3.3.3	*ReservationOnAccept*

[11]

Chapter 13 Flow Profile Identifier Assignments

2.2.4.2 Test Procedure

a. Instruct the AN to terminate existing session and cause the AT to negotiate a new session with the AN.

b. Cause the AT to establish a connection with the AN.

c. Cause the AT to transmit a *ReservationOnRequest* message for reservation label 0xXY.

d. Ensure that the AT or AN transmits a *ConfigurationRequest* message(s) negotiating Reservation*XY*QoSRequestFwd and Reservation*XY*QoSRequestRev attributes for ReservationLabel 0xXY.

e. Ensure that the AN and AT negotiate Reservation*XY*StreamFwd and Reservation*XY*StreamRev attributes of ReservationLabel 0xfe to stream S. Ensure that the AN and AT negotiate Reservation*XY*PacketFiltersFwd and Reservation*XY*PacketFiltersRev attributes for the ReservationLabel 0xXY.

f. Verify that the AN responds to the *ReservationOnRequest* message and transmits a *ReservationOnAccept* message to the AT.

g. Verify that the AN sets the fields of the *ReservationOnAccept* message as follows

    1. MessageID is set tot 0x02

    2. TransactionID is set to the same value as TransactionID field of the *ReservationOnAccept* message to which the AN is responding.

    3. ReservationCount is set to the number of reservations carried in the *ReservationOnRequest* message in response to which this message is being sent

    ReservationCount occurrences of the following variables

    4. ReservationLabel is set to 0xXY for a ReservationLabel.

    5. Link is set to ‘1’ for forward link reservation and ‘0’ to reverse link reservation

    6. ProfileType is set to 0x03 for all occurrences

    7. ProfileLength is set to 0x0001

    8. ProfileValue is set to the QoS_ATTRIBUTE_SET_ID

h. Cause a data source to generate a packet for transmission from the AN that matches the packet filter for ReservationLabel 0xXY. Verify that the AN transmits the packet using the stream S.

2.2.4.3 Minimum Standard

AN shall comply with step f and g.

AN should comply with step h.

2.2.5 Reservation Open State Requirements for the AN

2.2.5.1 Definition

If the AN transmits a *ReservationOffAccept* message for a forward link reservation or *ReservationReject* message in response to a *ReservationOnRequest* message, it shall transition the reservation to close state.

AN will transition all reservations other than 0xfe to close state upon receiving an *AirLinkManagement.IdleInitiated* indication.

Part 1 of the test verifies the change in reservation state with the transmission of *ReservationOffAccept* message.

Part 2 of the test verifies that the AN turns off a reservation (≠ 0xfe) when the connection with the AT is lost.

QoS Application Requirements:

The test requires the application generating data for transmission at the AN to submit packets to the AN even though the reservation is turned off.

Traceability:

[4]

Section 2.3.2.4.1	Reservation State Maintenance
Section 2.3.2.4.1.1.1	Packet Filter Processing Requirements
Section 2.3.2.4.3.2	Open State Requirements for AN
Section 2.3.3.2	*ReservationOffRequest*
Section 2.3.3.4	*ReservationOffAccept*
Section 2.3.3.5	*ReservationReject*

2.2.5.2 Test Procedure

a. Instruct the AT or AN to terminate existing session and cause the AT to negotiate a new session with the AN.

b. Cause the AT to establish a connection with the AN.

c. Cause the AT transmits a *ConfigurationRequest* message(s) negotiating Reservation*XY*QoSRequestFwd and Reservation*XY*QoSRequestRev attributes for ReservationLabel 0xXY.

d. Cause the AN and AT negotiate Reservation*XY*StreamFwd and Reservation*XY*StreamRev attributes of ReservationLabel 0xfe to stream S. Cause the AN and AT negotiate Reservation*XY*PacketFiltersFwd and Reservation*XY*PacketFiltersRev attributes for the ReservationLabel 0xXY.

e. Cause the AT to transmit a *ReservationOnRequest* for forward ReservationLabel 0xXY. Ensure that the AN transmits a *ReservationOnAccept* message in response to the *ReservationOnRequest* message.

f. Cause a data source to generate a packet for transmission from the AN that matches the packet filter for ReservationLabel 0xXY. Ensure that the AN transmits the packet using the stream S.

Part 1

g. Cause the AT to transmit *ReservationOffRequest* message for ReservationLabel 0xXY to the AN. Ensure that the AN transmits a *ReservationOffAccept message* in response.

h. Verify that the AN sets the fields of the *ReservationOffAccept* message as follows

1. MessageID field is set to 0x03.

2. TransactionID field of the *ReservationOffAccept* message is the same as the *ReservationOffRequest* message in response to which this message is being transmitted.

i. Generate a packet for transmission from the AN that matches the packet filter for ReservationLabel 0xXY. Verify that the AN does not transmit the packet using any other stream than the stream bound to ReservationLabel 0xfe.

Part 2

j. If the reservation 0xXY is in a closed state, cause the AT to transmit a *ReservationOnRequest* message for reservation label 0xXY. Ensure that the AN responds to the that the AN transmits a *ReservationOnAccept* message to the AT.

k. Allow or cause the connection to go idle.

l. Generate a packet for transmission from the AN that matches the packet filter for ReservationLabel 0xXY. Ensure that the AN transmits the packet using the stream S.

m. Generate a packet for transmission from the AN that matches the packet filter for ReservationLabel 0xXY. Verify that the AN does not transmit the packet using any other stream than the stream bound to ReservationLabel 0xfe.

2.2.5.3 Minimum Standard

AN shall comply with steps h, i and m.

3 RADIO LINK PROTOCOL TESTS

The Basic Radio Link Protocol (BRLP) of the Radio Link Layer (RLL) of UMB provides a packet stream between the AT and AN. Please see [4] for details.

In the test procedures in this section AT and AN transmit IP data with Best Effort QoS using single carrier. This stream is denoted as S. Note, typically stream 0x08 will be used for this transfer, but other streams can be used. See section 4.1 of [4] for details. Unless otherwise stated, the default values for stream 0x08 as specified in [4] is used for various attributes for stream S. It is expected that the AN will also deliver data for this stream in order to the higher layer. Similarly, stream S' is used for carrying IP data with DataUnitFwd and DataUnitRev being set to 0x01. For stream S' OutofOrderDeliveryFwd attribute should be set to 0x01 and NakAckEnableFwd should be set to 0x00. It is expected that the AN will also deliver data for this stream out of order to the higher layer. Unless otherwise stated, test procedure steps should be executed in such way that the AT does not transition to idle.

Certain tests in this section require that a packet or message be lost (or dropped). Unless the precise location of the loss is specified, the packet or message can be dropped anywhere before it reaches the intended receiver. This includes the option of suppressing the message generation, or dropping the packet / message at a buffer or causing the loss due to erasures during over the air transmission. When a SAR packet loss is specified in a test procedure, multiple packets may be dropped as long as they carry contiguous SAR data units.

When a message or packet transmission time needs to be verified, a processing delay of *p* ms should be allowed from the time of the trigger event. For example, if reception of a SAR packet causes an *ATReceiverStatus* message to be generated, then the *ATReceiverStatus* message should be generated within *p* ms of the reception of the SAR packet.

Testing Requirements:

Tests in this section require the AT and the AN to log various RLP messages used in the test. Unless otherwise specified, AT minimum standard should be verified by logging messages at the AN. Similarly, AN minimum standard should be verified by logging messages at the AT. When required to ensure that signaling has occurred, logging should be done at the AT as well as the AN. Besides signaling, logging of SAR packets is also needed. Any other requirements are mentioned in the definition of the test.

3.1 Basic Radio Link Protocol Access Terminal Tests

3.1.1 AT Transmitter Initiated SAR Reset

3.1.1.1 Definition

This test verifies the SAR transmitter initiated reset at the AT.

Part 1 of the test (step e) verifies that the SAR transmitter does not start transmission of new data units in reset state. It also verifies that the SAR transmitter ignores the ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed indication in the in reset state.

Part 2 of the test (step f) verifies that if the SAR transmitter receives a retransmission request for data that it did not transmit, it initiates a SAR reset. It also verifies that the SAR transmitter increments the SARResetCounter upon reset.

Part 3 of the test (step h) verifies that the SAR transmitter ignores the *ANReceiverStatus* message in reset state.

Part 4 of the test (step j) verifies that the SAR transmitter initializes V(S) upon reset.

Part 5 of the test (step l) verifies that the SAR transmitter ignores the *ResetTxComplete* message when it is not in reset state.

AN Control Requirements:

The AN should be able to:

a. Generate *ANReceiverStatus* message requesting retransmission of data unit that have not been transmitted by the AT.

b. Generate *ANReceiverStatus* message requesting retransmission of data unit that it successfully received in the last AbortTimer interval.

c. Generate *ResetTxComplete* message to the AT, while the AT is not in SAR Reset state.

Traceability:

[4]

Section 3.3.1	Protocol Initialization for the InUse Protocol Instance
Section 3.4.3.1.1.2.1	Reset Procedure for the Initiating Side when it is a SAR Transmitter
Section 3.4.3.1.1.2.5	SAR Reset Message Flows

[3]

Section 8.5.5.1.6	Access Terminal Transmission Logic for Persistent Assignments

3.1.1.2 Test Procedure

a. Configure the AN to negotiate PhysicalLayerNAKEnableRev to 0x01 for stream S.

b. Cause the AT to establish a connection with the AN.

c. Start bi-directional transfer for IP data with Best Effort QoS using stream S.

d. After the data transfer has started for stream S, instruct the AN to transmit an *ANReceiverStatus* message requesting retransmission of data unit 2^{P-1} - 1, i.e. the TrailingEdge in the *ANReceiverStatus* message is equal to 2^{P-1}, where P is the SARSequenceLength for stream S. Note, the *ANReceiverStatus* message should be

received before the AT transmits 2^{P-1} data units to the AN. Optionally, *ANReceiverStatus* message can carry any TrailingEdge value greater than V(S) of the SAR transmitter at the time the message is received at the transmitter.

Part 1

e. Verify that after receiving the *ANReceiverStatus* message the SAR transmitter for stream S the AT does not start transmission of SAR packets until *ResetTxComplete* message is received from the AN. Note, it should be verified that the AT does not retransmit any packets for which it receives ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed indication.

Part 2

f. Verify that the SAR transmitter at the AT transmits a *ResetTxIndication* message after the transmission of SAR packets is complete and sets the ResetCounter field of the *ResetTxIndication* message to 0x02. Verify that the AT transmits the *ResetTxIndication* message using stream 0x02.

Part 3

g. Before the AN responds to the *ResetTxIndication* message, instruct the AN to transmit an *ANReceiverStatus* message requesting retransmission of any data unit received in the last AbortTimer interval at the AN. Note, AbortTimer is part of the TimersRev attribute.

h. Verify that the AT does not transmit any data upon receiving the *ANReceiverStatus* message transmitted in step g.

Part 4

i. Ensure that the AN transmits *ResetTxComplete* message to the AT and that the ResetCounter field is set to 0x02.

j. Verify that the first SAR packet for stream S transmitted by the AT after receiving *ResetTxComplete* message from the AN has SARSEQ set to 0.

Part 5

k. Instruct the SAR receiver for stream S at the AN to transmit a *ResetTxComplete* message to the AT.

l. Verify that the AT ignores the *ResetTxComplete* message and continues transmitting data for stream S with SARSEQ field in the RLP header set to a non-zero value.

3.1.1.3 Minimum Standard

AT shall comply with steps e, f, h, j, and l.

3.1.2 AT SAR Receiver response to AN Transmitter Initiated Reset

3.1.2.1 Definition

When the SAR receiver receives a *ResetTxIndication* message it performs SAR initialization and transmits a *ResetTxComplete* message.

Part 1 of the test (step e) verifies that the AT transmits a *ResetTxComplete* message upon receiving a *ResetTxIndication* message.

Part 2 of the test (step g) verifies that the AT initializes the SAR variables V(R) and V(N) (see [4]) upon receiving *ResetTxIndication* message.

AN Control Requirements:

The AN should be able to:

a. Generate a *ResetTxIndication* message without receiving ATReceiverStatus message.

b. Generate data starting with SARSEQ greater than 0 after a SAR Reset has occurred.

Traceability:

[4]

Section 3.4.3.1.1.2.3 Reset Procedure for the Responding Side when it is a SAR Receiver

3.1.2.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Start bi-directional transfer for IP data with Best Effort QoS using stream S.

c. Stop the data transmission on stream S.

Part 1

d. Cause the SAR transmitter at the AN for stream S to transmit a *ResetTxIndication* message to the AT with ResetCounter field set to 0x02.

e. Verify that the AT transmits a *ResetTxComplete* message to the AN with ResetCounter field set to 0x02 using stream 0x02.

Part 2

f. After the AN receives the *ResetTxComplete* message, instruct the SAR transmitter for stream S to transmit a SAR packet with SARSEQ greater than 0. Optionally, the first few packets can be dropped before being delivered to the SAR receiver.

g. Verify that the AT transmits an *ATReceiverStatus* message requesting retransmission of SARSEQ 0.

3.1.2.3 Minimum Standard

AT shall comply with steps e and g.

3.1.3 AT Receiver Initiated SAR Reset

3.1.3.1 Definition

SAR receiver may initiate a reset and transmit a *ResetRxIndication* message to the transmitter. Note, the conditions under which the SAR Receiver will reset are not specified in [4]. These conditions are implementation dependent. For example, AT SAR Receiver may reset if there are large number of *ATReceiverStatus* messages generated. The exact condition for SAR reset should be known beforehand. Otherwise, this test should not be executed. This test is not mandatory for the compliance to this specification.

Part 1 of the test verifies that the AT increments ResetCounter upon transmitting *ResetRxIndication* message.

Part 2 of the test verifies that SAR receiver performs SAR initialization and ignores all data until it receives a *ResetRxComplete* message.

AN Control Requirements:

The AN should be able to drop SAR packets after the SAR Reset.

Traceability:

[4]

Section 3.4.3.1.1.2.2	Reset Procedure for Initiating Side when it is a SAR Receiver
Section 3.4.3.1.1.2.4	Reset Procedure for the Responding Side when it is a SAR transmitter

3.1.3.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Start bi-directional transfer for IP data with Best Effort QoS using stream S.

Part 1

c. Cause the SAR receiver for stream S to reset at the AT. Note the SARSEQ of the last packet received at the SAR receiver before SAR reset.

d. Verify that the AT transmits *ResetRxIndication* message to the AN with ResetCounter set to 0x02.

e. Ensure that the AN responds to the *ResetRxIndication* message by transmitting a *ResetRxComplete* message with ResetCounter set to 0x02 after the transmission of currently in-transmission SAR data units is complete.

Part 2

f. Ensure that after transmitting the *ResetRxComplete* message, the SAR transmitter for stream S transmits the first data unit with SARSEQ set to 0.

g. Cause a SAR packet for stream S to be dropped before reaching the SAR receiver at the AT. Ensure that the SARSEQ of the lost packet is less than the SARSEQ of the last packet received before *ResetRxIndication* message was transmitted by the AT.

h. Verify that the AT generates an *ATReceiverStatus* message for the data unit lost in step g.

3.1.3.3 Minimum Standard

AT shall comply with steps d and h.

3.1.4 AT Response to AN Receiver Initiated SAR Reset

3.1.4.1 Definition

This test verifies SAR Receiver initiated reset response at the AT

Part 1 of the test verifies that the SAR transmitter will respond to the *ResetRxIndication* message by transmitting a *ResetRxComplete* message.

Part 2 of the test verifies that the SAR transmitter will perform initialization after transmitting *ResetRxIndication* message.

AN Control Requirements:

The AN should be able to reset the SAR Receiver.

Traceability:

[4]

Section 3.4.3.1.1.2.2	Reset Procedure for Initiating Side when it is a SAR Receiver
Section 3.4.3.1.1.2.4	Reset Procedure for the Responding Side when it is a SAR transmitter

3.1.4.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Start bi-directional transfer for IP data with Best Effort QoS using stream S.

Part 1

c. Cause the SAR receiver for stream S at the AN to reset.

d. Ensure that the AN transmits *ResetRxIndication* message to the AN with ResetCounter set to 0x02.

e. Verify that the AT responds to the *ResetRxIndication* message by transmitting a *ResetRxComplete* message with ResetCounter set to 0x02 after the transmission of currently in-transmission SAR data units is complete.

Part 2

f. Verify that after transmitting the *ResetRxComplete* message, the SAR transmitter for stream S at the AT transmits the first data unit with SARSEQ set to 0.

3.1.4.3 Minimum Standard

AT shall comply with steps e and f.

3.1.5 StreamEarlyOpenRev Test

3.1.5.1 Definition

This test verifies that the AT does not start transmission of data for a stream if StreamealryOpenRev is set to 0x00, unless the AT has an open route with the AN and has been assigned a LinkID.

AN Control Requirements:

The AN should be able delay LinkID assignment to the AT.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

[9]

Section 2.5.1.5 LinkID Procedures

3.1.5.2 Test Procedure

a. Configure the AN to delay the assignment of LinkID to the AT.
b. Ensure that the stream S uses StreamEarlyOpenRev is set to 0x00.
c. Cause the AT to open a route and establish a connection with the AN.
d. Generate data for stream S.
e. Verify that the AT does not transmit data for stream S until it receives a LinkID assignment.

3.1.5.3 Minimum Standard

AT shall comply with step e.

3.1.6 Retransmissions when PhysicalLayerNAKEnableRev is set to 0x01

3.1.6.1 Definition

This test verifies the retransmission at the SAR transmitter at the AT when PhysicalLayerNAKEnableRev is set to 0x01 and a packet transmission is unsuccessful after the maximum number of harq attempts.

AN Control Requirements:

This test requires the AN to generate a NAK on the harq channel after the data transfer has started on the reverse link.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

3.1.6.2 Test Procedure

a. Configure the AN to negotiate NakAckEnableRev to 0x00 and PhysicalLayerNAKEnableRev to 0x01 for stream S.

b. Cause the AT to open a route and establish a connection with the AN.

c. Generate data for stream S.

d. Adjust the channel conditions such that a packet transmission fails after maximum number of harq attempts on the reverse link or configure the AN to NACK on the HARQ channel.

e. Verify that the AT retransmits the SAR data lost in step d.

3.1.6.3 Minimum Standard

AT shall comply with step e.

3.1.7 NakAckEnableFwd set to 0x01 and NakDelayTimerSingleCarrier Usage

3.1.7.1 Definition

This test verifies that the SAR receiver at the AT requests retransmission of missing SAR packets (step e).

Part 1 of the test verifies that the AT will delay the transmission of *ATReceiverStatus* message for NakDelayTimerSingleCarrier duration. This is verified in step f and g.

Part 2 of the test (step k) verifies that if the AT receives missing data units within this duration, the AT will not transmit the *ATReceiverStatus* message.

AN Control Requirements:

This test requires the AN to drop packets on the forward link. Some dropped packets need to be transmitted within NAKDelayTimerSingleCarrier to the AT.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

Section 3.4.3.2.2 SAR Receive Procedures

3.1.7.2 Test Procedure

a. Configure the AT or AN to negotiate NakDelayTimerSingleCarrier to 0x0064 either during Session configuration or through FastConfiguration.

b. Cause the AT to establish a connection with the AN.

c. Start bi-directional transfer for IP data with Best Effort QoS using stream S.

Part 1:

d. Cause a SAR packet for forward link stream S to be dropped.

e. Verify that the AT transmits an *ATReceiverStatus* message requesting retransmission of all the missing data units.

f. Verify that the *ATReceiverStatus* message is not transmitted within NakDelayTimerSingleCarrier ms after the AT received the first SAR packet with SARSEQ number greater than that of the lost packet(s).

g. Verify that the *ATReceiverStatus* message is transmitted within NakDelayTimerSingleCarrier ms to NakDelayTimerSingleCarrier + *p* ms after the AT received the first SAR packet with SARSEQ number greater than that of the lost packet(s).

Part 2:

h. Continue bi-directional transfer for IP data with Best Effort QoS using stream S.

i. Cause a SAR packet for forward link stream S to be dropped.

j. Ensure that the AN transmits the dropped packets to the AT and the AT receives the lost packet within NakDelayTimerSingleCarrier of the time the AT received the first SAR packet with SARSEQ number greater than that of the packet(s) lost in step i. Note, such a transmission of dropped packets may occur if the AN transmits packets using ForwardPacketsMissidIndication without waiting for ATReceiverStatus to be received from the AT.

k. Verify that the AT does not transmit *ATReceiverStatus* message requesting retransmission of the packet(s) lost in step i.

3.1.7.3 Minimum Standard

AT shall comply with step e, f and k.

AT should comply with step g.

3.1.8 ATReceiverStatus transmission when NakAckEnableFwd is set to 0x02

3.1.8.1 Definition

When NakAckEnableFwd attribute is set to 0x02 for a stream, the AT transmits an *ATReceiverStatus* message acknowledging the reception of data unit within AckTimer time interval of receiving the data unit. This is verified in step e.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

Section 3.4.3.2.2 SAR Receive Procedures

3.1.8.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Ensure that the AN and AT use a value of 0x02 for NakAckEnableFwd attribute for stream 0x02.

c. Cause the AN to transmit a signaling message to the AT using stream 0x02. This can be achieved, for example, by triggering a SAR reset at the AN.

d. Ensure that the message is received at the AT.

e. Verify that the AT transmits an *ATReceiverStatus* message acknowledging the receipt of the data units transmitted in step c within AckTimer of receiving the data units.

3.1.8.3 Minimum Standard

AT shall comply with step e.

3.1.9 AT RLP retransmission when NakAckEnableRev attribute is set to 0x02

3.1.9.1 Definition

When NakAckEnableRev attribute is set to 0x02 for a stream, the AT will retransmit a data unit for this stream if it does not receive an acknowledgement for the data unit within an implementation dependent time. This is verified in step e. The AT will transmit the message MaxTransmissionsRev times until it receives an acknowledgement. This is verified in step g. Step g also verifies that the AT closes the connection after an implementation dependent retransmission attempts fail for a reliable transmission stream.

AN Control Requirements:

This test requires the AN to not drop packets transmitted by the AT using reliable RLP stream. Alternatively, the AN may prevent the acknowledgement for the reliable stream from being transmitted.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

Section 3.4.3.2.2 SAR Receive Procedures

3.1.9.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Ensure that the AN and AT use a value of 0x02 for NakAckEnableRev attribute for stream 0x02.

c. Cause the AT to transmit a signaling message to the AN using stream 0x02. This can be achieved, for example, by triggering a SAR Reset at the AT.

d. Either drop the SAR packets carrying the signaling message or drop the *ANReceiverStatus* message transmitted by the AN in response to data transmitted in step c.

e. Verify that the AT retransmits the signaling message that it sent in step c.

f. Continue to either drop the SAR packets carrying the signaling message or drop the *ANReceiverStatus* message transmitted by the AT in response to the signaling message.

g. Verify that the AT transmits the message MaxTransmissionsRev times and then closes the connection.

3.1.9.3 Minimum Standard

AT should comply with steps e and g.

3.1.10 Flush procedure at the AT

3.1.10.1 Definition

When NakAckEnableRev is set to 0x01, the SAR transmitter at the AT transmits a SAR packet containing the data unit with the greatest SAR Sequence after FlushTimer duration has elapsed since the last time it transmitted a packet. This is verified in the test procedure e.

Traceability:

[4]

Section 3.4.3.2.1	SAR Transmit Procedures
Section 3.4.3.2.2	SAR Receive Procedures

3.1.10.2 Test Procedure

a. Ensure that the AN and AT use a value of 0x01 for NakAckEnableRev attribute for stream S.

b. Ensure that AT uses a value of 0x00 for PhysicalLayerNakEnableRev for stream S.

c. Cause the AT to establish a connection with the AN and to transmit data to the AN using stream S.

d. Cause the data transmission at the AT to stop and note the time (t0) of the first transmission of the SAR packet containing the data unit with the greatest SARSEQ (S).

e. Verify that the AT transmits a SAR packet containing the data unit with the greatest SARSEQ (S) after FlushTimer interval from t0, where FlushTimer is a field in the TimersRev attribute. Note a processing delay of *p* ms should be allowed.

3.1.10.3 Minimum Standard

AT should comply with step e.

3.1.11 AbortTimer and OutOfOrderDeliveryFwd Usage at AT

3.1.11.1 Definition

This test verifies the AT performance w.r.t. delivery of data to higher layers. If the SAR receiver at the AT determines that a SAR packet(s) is lost then it may request retransmission of the lost packet(s). If during the recovery period other packets are received completely, then the AT passes these packets either in-order or out-of-order to the higher layers. If the OutOfOrderDeliveryFwd is set to 0x01, then SAR receiver at the AT will pass the packets out-of-order to the higher layer. This is verified in step r. If the OutOfOrderDeliveryFwd is set to 0x00, then SAR receiver at the AT will pass the packets in-order to the higher layer. The release of packets can occur either when the recovery of all missing data is complete or if the abort timer expires. This is verified in steps l and h respectively. If the missing data unit is received before the corresponding abort timer expires, then the data unit is passed to the higher layer. This is verified in step x. If a data unit is received after the abort timer corresponding to it has expired, then the data unit is discarded. This is verified in step aa.

Part 1 of the test verifies the abort functionality at the AT for stream S.

Part 2 of the test verifies the recovery of data within abort timer.

Part 3 of the test verifies the out-of-order delivery of data to the higher layers for stream S'.

Part 4 of the test verifies the recovery of data for stream S'.

AN Control Requirements:

This test requires the AN to create gaps in the SARSEQ of the packets received by the AT. This can be achieved, for example, by dropping some packets or by skipping certain SARSEQ numbers while generating the SAR packets. In some cases, it requires the AN to transmit dropped packets to the AT before the abort timer expires.

This test requires the AT to log the arrival of the SAR packets. Logging of release of higher layers packets is recommended, but not necessary as this can be done at the higher layer.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

Section 3.4.3.2.2 SAR Receive Procedures

3.1.11.2 Test Procedure

a. Configure the AN to negotiate NakAckEnableFwd for stream S and S' to 0x00 either during Session configuration or through FastConfiguration.

b. Start bi-directional transfer for IP data with Best Effort QoS using stream S.

c. During data transfer, cause SAR packet(s) for forward link stream S to be dropped.

Part 1:

d. Log the time of reception (t0) of the first packet carrying SAR data unit with SARSEQ greater than the lost packets.

e. Ensure that the AN does not transmit the dropped packet using ForwardPacketsMissidIndication and continues transmitting data to the AT.

f. Verify that the AT does not transmit an *ATReceiverStatus* message for the packet(s) lost on stream S to the AN.

g. Verify that the SAR receiver for stream S at the AT does not release the packets with SARSEQ greater than that of the first lost packet till t0 + AbortTimer, where AbortTimer is a parameter of the TimersFwd attribute.

h. Verify that the SAR receiver at the AT release all completed packets with SARSEQ greater than that of the first lost packet in order to the higher layer. Verify that the first in order packet is released within *p* ms of the expiration of the abort timer. Note, the logging for the release of higher layer packet can be done outside the air-interface.

Part 2:

i. Cause the AN to negotiate NakAckEnableFwd for stream S to 0x01 using FastConfiguration or a new Session Configuration.

j. Repeat steps b-c.

k. Ensure that the AT receives a SAR data unit before the AN retransmits the lost packet(s). Ensure that the AN retransmits the lost packets before the AbortTimer for the lost packet expires at the SAR receiver at the AT.

l. Verify that the SAR receiver at the AT releases all completed packets in order to the higher layers. Verify that the first of the recovered packets is released within *p* ms of the delivery of the last SAR data unit recovered through retransmission.

Part 3:

m. Start bi-directional transfer for IP data using stream S'.

n. During data transfer, cause SAR packet(s) for forward link stream S' to be dropped.

o. Log the time of reception (t1) of the first packet carrying SAR data unit with SARSEQ greater than the lost packets.

p. Ensure that the AN does not retransmit the packet and continues transmitting data to the AT.

q. Verify that the AT does not transmit an *ATReceiverStatus* message for the packet(s) lost on stream S' to the AN.

r. Verify that the SAR receiver for stream S' at the AT releases the packets to the higher layer with SARSEQ greater than that of the first lost packet within *p* ms of

receiving the completed data units. Note, the delivery times of the higher layer packets may be observed after they are released from the AT.

s. Stop data transmission on stream S'.

Part 4:

t. Cause the AN to negotiate NakAckEnableFwd for stream S' to 0x01 using FastConfiguration or a new Session Configuration.

u. Start bi-directional transfer for IP data using stream S'.

v. During data transfer, cause SAR packet(s) for forward link stream S' to be dropped.

w. Ensure that the AT receives a complete SAR data unit with SARSEQ greater than the lost packet before the AN retransmits the lost packet(s). Ensure that the AN retransmits the lost packets before the AbortTimer for the lost packet expires at the SAR receiver at the AT.

x. Verify that the SAR receiver for stream S' at the AT releases the lost packets to the higher layer when they are received completely. Note, the delivery of the packets to the higher layer will be out of order.

y. Repeat step v.

z. Cause the AN to retransmit the lost packets after the AbortTimer has expired at the AT.

aa. Verify that the SAR receiver at the AT does not pass the retransmitted packets to the higher layer. Note, this may be verified by counting the number of packet delivered by the AT to the higher layer.

3.1.11.3 Minimum Standard

AT shall comply with steps f, g, h, l, q, r, x and aa.

3.1.12 TransmitterStatus Message

3.1.12.1 Definition

When a *TransmitterStatus* message is received at the SAR receiver, the receiver will move the V(R) and V(N) pointers and perform an abort.

AN Control Requirements:

This test requires the AN to control the transmission of *TransmitterStatus* message to the AT.

Traceability:

[4]

Section 3.4.3.2.1	SAR Transmit Procedures
Section 3.4.3.2.2	SAR Receive Procedures
Section 3.4.3.3.8	*TransmitterStatus* message

3.1.12.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Cause the AN to establish a connection with the AT and to transmit data to the AT using stream S.

c. Cause the AN drop packet(s) during transmission to the AT using stream S.

d. Ensure that the AT transmits an *ATReceiverStatus* message requesting retransmission of lost data.

e. Instruct the AN not to retransmit the data requested in the *ATReceiverStatus* message and continue to transmit at least 1 higher layer packet to the AT.

f. Ensure that the AT performs an abort and passes the completed packets to the higher layer.

g. Allow the connection to go idle.

h. Repeat steps b-d.

i. Instruct the AN to transmit at least 1 higher layer packet to the AT and to transmit a *TransmitterStatus* message to the AT indicating that it does not have the data units that AT requested in step *ATReceiverStatus* message.

j. Verify that upon receiving the *TransmitterStatus* message the AT passes the higher layer packet with SARSEQ greater than the lost data units. Note, for verification, it may be possible to note the time taken to deliver the packet to the higher layer as this will be smaller than step f.

3.1.12.3 Minimum Standard

AT shall comply with steps j.

3.2 Basic Radio Link Protocol Access Network Tests

3.2.1 AN Response to ATReceiverStatus message

3.2.1.1 Definition

This test verifies that the AN retransmits the missing data unit requested in the *ATReceiverStatus* message.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

Section 3.4.3.2.2 SAR Receive Procedures

3.2.1.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Start bi-directional transfer for IP data with Best Effort QoS using stream S.

c. Ensure that the AT transmits an *ATReceiverStatus* message requesting retransmission of the missing data units. This can be caused for example by dropping a SAR packet on for stream S of the forward link.

d. Verify that the AN retransmits the packets requested in the *ATReceiverStatus* message.

3.2.1.3 Minimum Standard

AN shall comply with step d.

3.2.2 ANReceiverStatus transmission when NakAckEnableRev is set to 0x02

3.2.2.1 Definition

When NakAckEnableRev attribute is set to 0x02 for a stream, the AN transmits an *ANReceiverStatus* message acknowledging the reception of data unit within AckTimer time interval of receiving the data unit. This is verified in step e.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

Section 3.4.3.2.2 SAR Receive Procedures

3.2.2.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Ensure that the AN and AT use a value of 0x02 for NakAckEnableRev attribute for stream 0x02.

c. Cause the AT to transmit a signaling message to the AN using stream 0x02.

d. Ensure that the message is received at the AN.

e. Verify that the AN transmits an *ANReceiverStatus* message acknowledging the receipt of the data units transmitted in step c within AckTimer of receiving the data units in step c.

3.2.2.3 Minimum Standard

AN shall comply with step e.

3.2.3 Flush procedure at the AN

3.2.3.1 Definition

When NakAckEnableFwd is set to 0x01, the AN transmits a Flush packet or a message depending on NakDelayTimerMultiCarrier or NakDelayTimerSingleCarrier fields of the TimersFwd attribute.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

Section 3.4.3.2.2 SAR Receive Procedures

3.2.3.2 Test Procedure

a. Ensure that the AN and AT use a value of 0x01 for NakAckEnableFwd attribute for stream S.

b. Configure the AN or the AT to negotiate NakDelayTimerMultiCarrier and NakDelayTimerSingleCarrier fields of the TimersFwd attribute to 0x0000.

c. Cause the AN to establish a connection with the AT and to transmit data to the AT using stream S.

d. Instruct the AN to stop transmitting data to the AT and note the time (t0) of the first transmission of the SAR packet containing the data unit with the greatest SARSEQ (S).

e. Verify that the AN transmits a SAR packet or a *Flush* message containing the data unit with the greatest SARSEQ after FlushTimer interval from t0, where FlushTimer is a field in the TimersFwd attribute.

f. Configure the AN or the AT to negotiate NakDelayTimerSingleCarrier field of the TimersFwd attribute to a value other than 0x0000.

g. Repeat steps c-d.

h. Verify that the AN transmits a *Flush* message containing the data unit with the greatest SARSEQ after FlushTimer interval from t0, where FlushTimer is a field in the TimersFwd attribute.

3.2.3.3 Minimum Standard

AN shall comply with steps e and h.

3.2.4 AbortTimer Usage at AN

3.2.4.1 Definition

This test verifies the abort timer functionality at the AN and is similar to test 3.1.11.

Traceability:

[4]

Section 3.4.3.2.1 SAR Transmit Procedures

Section 3.4.3.2.2 SAR Receive Procedures

3.2.4.2 Test Procedure

a. Configure the AT or AN to negotiate NakAckEnableRev for stream S to 0x00, and PhysicalLayerNAKEnableRev is set to 0x00 during session configuration or through FastConfiguration.

b. Cause the AT to establish a connection with the AN.

c. Start bi-directional transfer for IP data with Best Effort QoS using stream S.

d. During data transfer, cause SAR packet(s) for reverse link stream S to be dropped.

e. Log the time of reception (t0) of the first packet carrying SAR data unit with SARSEQ greater than the lost packet(s).

f. Verify that the AN does not transmit an *ANReceiverStatus* message for the packet(s) lost on stream S to the AT.

g. Verify that the SAR receiver for stream S at the AN does not release the packets with SARSEQ greater than that of the first lost packet till t0 + AbortTimer, where AbortTimer is a parameter of the TimersFwd attribute.

h. Verify that the SAR receiver at the AN releases all completed packets with SARSEQ greater than that of the first lost packet in order to the higher layer.

i. Cause the AN to negotiate NakAckEnableRev for stream S to 0x01 and PhysicalLayerNAKEnableRev is set to 0x01 using FastConfiguration or a new Session Configuration.

j. Repeat steps b-c.

k. Ensure that the AN receives a complete SAR data unit before the AT retransmits the lost packet(s). Ensure that the AT retransmits the lost packets before the AbortTimer for the lost packet expires at the SAR receiver at the AN. Such a transmission can be triggered using the ReverseTrafficChannelMAC.ReverseTrafficPacketsMissed indication.

l. Verify that the SAR receiver at the AN releases all completed packets in order to the higher layer. Verify that the first recovered packet is released within *p* ms of the delivery of the last SAR data unit recovered through retransmission.

3.2.4.3 Minimum Standard

AN shall comply with steps f, h, and l.

AN should comply with g.

4 ROUTE PROTOCOL TESTS

This section verifies the route creation headers and error code handling functionality of Route Protocol.

4.1 Basic RouteProtocol Access Terminal Tests

4.1.1 AT Initial Route Establishment Test

4.1.1.1 Definition

This test verifies the initial access and route setup for the AT. Step c verifies that the AT transmits a RouteCreation header in the initial access with the AN. Step d verifies that the AT transmits ATI header in the in the initial access with the AN. Step h verifies that the AT stops including extended headers in the Route Protocol packets once the route is open.

Traceability:

[4]

Section 5.5.1.6.1	Access Terminal Procedure (Transmit)
Section 5.5.1.4	Route Protocol Header

4.1.1.2 Test Procedure

a. If the AT has a session with the AN, instruct the test equipment to close the session.

b. Cause the AT to establish a session with the AN.

c. Verify that AT's access the AN and adds a RouteCreation Header with

   1. ExtendedHeaderIncluded field set to '1'
   2. HeaderType set to '000'
   3. PSIIncluded set to '0'
   4. IPSI set to one of the IPSI advertised by the AN
   5. PersonalityIndex set to 0xf.

d. Verify that the AT access the AN and adds a RouteCreation Header with

   1. ExtendedHeaderIncluded field set to '1'
   2. HeaderType set to '001'
   3. ATIType field set to '11'

e. Ensure that the AN transmits a *RouteOpenAccept* message to the AT.

f. Ensure that the AN assigns a UATI to the AT.

g. After the AT receives the *RouteOpenAccept* message from the AN, cause the AT to transmit data or signaling packets to the AN.

h. Verify that the AT does not include any extended headers in the Route Protocol packets that it transmits to the AN.

4.1.1.3 Minimum Standard

AT shall comply with steps c, d and h.

4.1.2 RouteCreation Header transmission in WaitingToOpen state

4.1.2.1 Definition

This test verifies that the AT continues to add RouteCreation headers in the packets sent to the AN while the route is in WaitingToOpen state and no FL packets are received on this route. Note, the test procedure requires the AT to transmit multiple MAC packets. This can be achieved by restricting the number of bytes allocated for transmission on the reverse link.

AN Control Requirements:

This test requires the AN to withhold packet transmission to the AT. It also requires the AN to control the RLAB to the AT such that the AT transmits data using multiple MAC packets.

Traceability:

[4]

Section 5.5.1.6.1	Access Terminal Procedure (Transmit)
Section 5.5.1.4	Route Protocol Header

[3]

Section 5.5.4.1.1.3.3	RLAB

4.1.2.2 Test Procedure

a. If the AT has an established connection with the AN, cause the AT to go dormant.

b. Instruct the AN to withhold packet transmission to the AT.

c. Cause the AT open a connection with the AN such that the AT transmits multiple MAC packets.

d. Verify that the AT includes the RouteCreation Header in all MAC packets that it transmits.

4.1.2.3 Minimum Standard

AT shall comply with step d.

4.1.3 RouteCreation with Decentralized SRNC

4.1.3.1 Definition

This test verifies that the AT does not add RouteCreation header to packets sent to the AN, if the local AN is the SRNC and the AT has an open route to it.

AN Control Requirement:

This test requires the local AN to be the SRNC.

Traceability:

[4]

Section 5.5.1.6.1 Access Terminal Procedure (Transmit)

Section 5.5.1.4 Route Protocol Header

4.1.3.2 Test Procedure

a. Configure the test equipment such that local AN contains the SRNC.

b. If the AT does not have a session with the AN, cause the AT to establish a session with the AN.

c. Ensure that the AT has a route with the SRNC.

d. Allow the connection to do dormant.

e. Cause the AT to establish a connection with the AN.

f. Verify that the AT does not include RouteCreation Header while accessing the AN.

4.1.3.3 Minimum Standard

AT shall comply with step f.

4.1.4 Access ATI with Session Active

4.1.4.1 Definition

This test verifies that the AT accesses the AN with UATI when it has a session. This is tested for connection set-up as well as power-up scenario. Note, in step g below, the AT should be powered off gracefully using the power-off button.

Traceability:

[4]

Section 5.5.1.6.1 Access Terminal Procedure (Transmit)

4.1.4.2 Test Procedure

a. If the AT does not have a session with the AN, instruct the AT to establish a session with the AN.

b. Ensure that the AN has assigned a UATI to the AT.

c. Allow the connection to become dormant.

d. Cause the AT to establish a connection.

e. Verify that the AT access the AN and adds a ATI Header with

1. ExtendedHeaderIncluded field set to '1'
2. HeaderType set to '001'
3. ATIType field set to '10'

f. Allow the connection to become dormant.

g. Power cycle the AT.

h. Cause the AT to establish a connection.

i. Verify that the AT access the AN and adds a ATI Header with

1. ExtendedHeaderIncluded field set to '1'
2. HeaderType set to '001'
3. ANType field set to '10'

4.1.4.3 Minimum Standard

AT shall comply with steps e and i.

4.1.5 Extended Headers with multiple PCP packets in MAC packet

4.1.5.1 Definition

If multiple Packet Consolidation Protocol packets are transmitted in a single MAC packet, the AT will include ExtendedHeader only for the first PCP packet. Such a scenario may occur when the AT has a negotiated session and the MAC packet carries request from multiple protocols. For example, *KeyRequest* message, *RouteOpenRequest* message may be transmitted in the same MAC packet.

AN Control Requirements:

This test requires RLAB assignment to be controlled such that the MAC packet transmitted by the AT can carry multiple PCP packets.

Traceability:

[4]

Section 5.5.1.6.1 Access Terminal Procedure (Transmit)

[3]

Section 5.5.4.1.1.3.3 RLAB

4.1.5.2 Test Procedure

a. If the AT does not have a session with the AN, cause the AT to establish the session with the AN.

b. After the session configuration is over allow the connection to go dormant.

c. Ensure that the AN allocates sufficient bytes to the AT on the RL such that the AT is able to transmit multiple PCP packets in a MAC packet.

d. Cause the AT to establish a connection with the AN and to transmit multiple PCP packets in single MAC packet. For example, AT could transmit *RouteOpenRequest*, *KeyRequest* message in different PCP packets.

e. Verify that the AT includes the Route Protocol Extended Headers only in the first PCP packet.

4.1.5.3 Minimum Standard

AT shall comply with step e.

4.1.6 ATI Header Transmission and Reception in BindATI State

4.1.6.1 Definition

This test verifies that AT continues to include ATI header in Route Protocol packets as long as it does not receive ATI echo from the AN. It also verifies the AT behavior during BindATI state of Connected State Protocol when AT receives an incorrect value of ATI in a Route Protocol packet or if ATI is not included in a Route Protocol Packet.

AN Control Requirements:

This test requires AN to withhold transmission to the AT in the BindATI state. It also requires the AN to skip the ATI header, place an incorrect ATI header in the Route Protocol packet while in BindATI state.

Traceability:

[4]

Section 5.5.1.6.1 Access Terminal Procedure (Transmit)

[7]

Section 5.4.3.1.2 Cleanup procedures for the BindATI State

[3]

Section 5.5.4.1.1.3.3 RLAB

4.1.6.2 Test Procedure

a. Instruct the AN to withhold from transmitting any packets to the AT until it receives multiple MAC packets from the AT.

b. Cause the AT to establish a connection and to transmit multiple MAC packets. This can be achieved by granting RLAB to the AT.

c. Verify that the AT includes ATI Header with HeaderType set to ‘001’ in all MAC packets transmitted.

d. After the connection is established, allow the connection to go idle.

e. Cause the AT to attempt a connection with the AN.

f. Instruct the AN to echo incorrect value of ATI in response to the Route Protocol packet transmitted by the AT, while ensuring that the MAC assignments are maintained.

g. Verify that the AT closes the connection and does not transmit on the traffic channel upon receiving RLAB.

h. Cause the AT to attempt a connection with the AN.

i. After the AN receives a packet from the AT which includes ATI Header with HeaderType set to '001', instruct the AN to transmit a Route Protocol packet without HeaderType '001'.

j. Verify that the AT closes the connection and does not transmit on the traffic channel upon receiving RLAB.

4.1.6.3 Minimum Standard

AN shall comply with steps c, g, and j.

4.1.7 AT Behavior Upon Receiving various Error Codes

4.1.7.1 Definition

This test verifies AT behavior upon receiving various error codes. Error codes '0000' (Personality not supported), '0010' (Route exists), and '0011' (Route does not exist) are tested here.

Part 1 verifies that the AT accesses the AN with PersonalityIndex 0xf, i.e. IPSI based access, when it receives an error code specifying that Personality is not supported.

Part 2 verifies that the AT closes the session when it receives an error code '0010'.

Part 3 verifies that the AT creates a route with the AN when it receives an error code '0011' (Route does not exist) from this AN.

AN Control Requirements:

For first 2 parts, the local AN should not be the SRNC for the AT. For the last part, local AN should also be the SRNC for the AT. This test also requires AN to transmit various ErrorCode headers in response to the RouteCreation headers received from the AT.

AT Requirements:

The AT should support

Traceability:

[4]

Section 5.5.1.7.1 Access Terminal Procedure (Reception)

[9]

Section 2.5.1.8.1 Access Terminal Requirements (WaitingToOpen State)

[8]

Section 2.5.1.6 Processing the Deactivate Command

4.1.7.2 Test Procedure

a. If the AT has an established session with the AN, instruct the AN to close the session.

b. During Session negotiation, configure the AN to negotiate a personality with the AT.

c. Ensure that the AN transmits a *SwitchPersonality* message to the AT to switch the InUse personality to the negotiated personality.

d. Allow the connection to go idle.

Part 1

e. Cause AT to establish a connection.

f. Instruct the AN to transmit a Route Protocol packet with HeaderType field set to '010' and ErrorCode field set to '0000' (Personality not supported).

g. Verify that the AT does not reaccess the AN with the same personality as step e.

h. Allow the AT to become dormant.

Part 2

i. Cause the AT to establish a connection with the AN.

j. In response to the Route Creation header received from the access terminal, instruct the AN to transmit a Route protocol packet with HeaderType '010', ErrorCode field set to '0010' (Close session) and CRCErrorDetectPattern field is set to '1010101010101010'.

k. Verify that the AT deactivates the session and transmits a *SessionClose* message to the AN.

l. Verify that the AT accesses the AN using an ATIType set to '11' (RATI).

Part 3: To be done when local AN serves as SRNC.

m. Ensure that the AT has an established session with the AN and that the connection is dormant.

n. Cause the AT to establish a connection with the AN.

o. In response to the Route Creation header received from the access terminal, instruct the AN to transmit a Route protocol packet with HeaderType '010', ErrorCode field set to '0011' (Route does not exist).

p. Verify that the AT transmits a *RouteOpenRequest* message to the AN along with a RouteCreation header.

4.1.7.3 Minimum Standard

AT shall comply with steps k, l, and p.

AT should comply with step g.

4.2 Basic Route Protocol Access Network Tests

4.2.1 ATI Header Transmission by the AN

4.2.1.1 Definition

This test verifies that the AN includes the ATI header in BindATI state.

Traceability:

[4]

Section 5.5.1.6.2 Access Network Procedures (Transmission)

4.2.1.2 Test Procedure

a. Cause the AT to establish a connection with the AN.

b. Ensure that the AT transmits the Route Protocol packets with ATI header.

c. Verify that includes the ATI header in the Route Protocol packets that it transmits and sets the value of the ATIType and ATI to the same as that received in the Route Protocol packet received from the AT.

4.2.1.3 Minimum Standard

AN shall comply with step c.

5 SESSION CONTROL PROTOCOL TESTS

This section provides SCP signaling conformance tests for the AT and the AN.

Testing Requirements:

Tests in this section require the AT and the AN to log various SCP messages used in the test. Unless otherwise specified, AT minimum standard should be verified by logging messages at the AN. Similarly, unless otherwise specified, AN minimum standard should be verified by logging messages at the AT. When required to ensure that signaling has occurred, logging should be done at the AT as well as the AN. Any other requirements are mentioned in the definition of the test.

5.1 Basic Session Control Protocol Access Terminal Tests

5.1.1 Initial Session Configuration

5.1.1.1 Definition

This test verifies the initiated session configuration for the AT. The test is divided into 3 parts.

Part 1: AT initiated state

Step d verifies that in the AT initiated state, the AT negotiates HardwareID, ATSupportedProtocolSetIDs and ATSupportedApplicationProtocolIDs when these attributes are set to their default values and the AT supports a non-default value for these attributes. Step e, verifies that when transmitting a *ConfigurationRequest* in response to *ConfigurationStart* message the AT uses Personality index that it received in the *ConfigurationStart* message. Step g, verifies that the AT transmits a *ConfigurationComplete* message when it has completed the negotiation of various attributes in the AT initiates state.

Part 2: AN Initiated state

Step i verifies that the AT responds to the AN *ConfigurationRequest* message by transmitting *ConfiguirationResponse* message. Step k, verifies that the AT transmits a *SessionUpdated* message when it receives a *ConfigurationComplete* message from the AN.

Part 3: Open State

Steps m and o verify that the AT changes the personality when it receives a *SwitchPersonality* message from the AN.

Traceability:

[8]

Section 2.4.1.1	Procedures and Messages for the InConfiguration Instance of the Protocol, Other Procedures
Section 2.5.1.1	Generic Configuration Procedures

Section 2.5.2.13	ConfigurationStart Message
Section 2.5.2.2	ConfigurationRequest Message
Section 2.5.2.12	ConfigurationComplete Message
Section 2.5.1.15.1	Access Terminal Requirements (AT Initiated State)
Section 2.5.1.9	Processing the Reconfigured Indication
Section 2.5.1.16.1	Access Terminal Requirements (AN Initiated State)
Section 2.5.1.14.1	Access Terminal Requirements (Open State)

5.1.1.2 Test Procedure

a. If the AT has a session with the AN, instruct the AN to close the session.

b. Cause the AT to negotiate a new session with the AN.

c. Ensure that the AN transmits a *ConfigurationStart* message with PersonalityIndex field set to 0xf and SessionSignature set to 0x0000.

Part 1

d. Verify that if the AT supports non-default values for HardwareID, ATSupportedProtocolSetIDs and ATSupportedApplicationProtocolIDs attributes, then the AT transmits a *ConfigurationRequest* to the AN message and negotiates these attributes.

e. Verify that the AT uses a value of 0xf for the PersonalityIndex field of the *ConfigurationRequest* message.

f. Ensure that the AN transmits a *ConfigurationResponse* message accepting the proposed value for the values for HardwareID, ATSupportedProtocolSetIDs and ATSupportedApplicationProtocolIDs attributes.

g. Verify that the AT transmits a *ConfigurationComplete* message after the AT initiated session negotiation is complete.

Part 2

h. Ensure that the AN transmits a *ConfigurationRequest* message negotiating a value of 0x0001 for ATInitiatedConfigurationAllowed attribute.

i. Verify that the AT transmits a *ConfigurationResponse* message accepting the value of 0x0001 for ATInitiatedConfigurationAllowed attribute.

j. Ensure that the AT receives a ConfigurationComplete message from the AN. Ensure that the AN includes the updated value of SessionSignature and a PersonalityIndexStore value in the range 0 to PersonalityCount-1 in the *ConfigurationComplete* message.

k. Verify that the AT transmits a *SessionUpdated* message to the AN and includes the same SessionSignature that it received in the *ConfigurationComplete* message from the AN.

Part 3

l. Ensure that the AN transmits a *SwitchPersonality* message to the AT with the same value for PersonalityIndex field as that of the PersonalityIndexStore field in the *ConfigurationComplete* message sent by the AN. Ensure that the AN sets the Hard-Switch field value correctly.

m. Verify that the AT transmits a *SwitchPersonalityAccept* message to the AN.

n. Cause the AT to transmit a *ConfigurationRequest* message. This may be achieved, for example, by causing the AT to negotiate ReservationKKPacketFiltersFwd for a reservation.

o. If the AT transmits a *ConfigurationRequest* message, verify that the sets the PersonalityIndex field in the message to the same value that it received in the *SwitchPersonality* message from the AN.

5.1.1.3 Minimum Standard

AT shall comply with step d, e, g, i, k, m, and o.

5.1.2 Negotiation and Deletion of Personalities

5.1.2.1 Definition

This test verifies the negotiation and deletion of personalities. It is divided into 2 parts.

Part 1: Negotiation of Personalities

This test verifies that the AT can store and use upto PersonalityCount different personalities.

Part 2: Deletion of Personalities

This test also verifies personality deletion behavior of the access terminal.

AN Control Requirements:

The AN should be able to support negotiation of at least 4 personalities. It should be able to transmit *SwitchPersonalityRequest* message to the AT and cause the AT to switch between all the negotiated personalities. The AN should be able to transmit a *DeletePersonalityRequest* message to the AT requesting deletion of personalities and should be able to order the list of indices of personalities that are required to be deleted.

Traceability:

[8]

Section 2.7.1	Simple Attributes
Section 2.5.2.18	SwitchPersonality Message
Section 2.5.2.19	SwitchPersonalityAccept Message
Section 2.5.2.13	ConfigurationStart Message

Section 2.5.2.2	ConfigurationRequest Message
Section 2.5.2.12	ConfigurationComplete Message
Section 2.5.2.15	DeletePersonalityRequest
Section 2.5.2.16	DeletePersonalityAccept
Section 2.5.2.17	DeletePersonalityReject
Section 2.5.1.16.1	Access Terminal Requirements (AN Initiated State)

5.1.2.2 Test Procedure

a. Configure the AN to negotiate maximum number of personalities supported by the AT. Note, maximum number of personalities supported by the AT will be determined by PersonalityCount attribute that may be negotiated by the AT during AT initiated state.

b. If the AT has a session with the AN, instruct the AN to close the session.

c. Cause the AT to negotiate a new session with the AN.

d. Verify that the AT uses a value of PersonalityCount that is greater than or equal to 0x0004.

e. Instruct the AN to negotiate ATInitiatedConfigurationAllowed attribute value to 0x0001 and ensure that the negotiation is successful.

f. Instruct the AN to negotiate a new personality.

g. Ensure that at the end of the AN initiated state the AN transmits a *ConfigurationComplete* message and the AT receives the message.

h. Ensure that in the *ConfigurationComplete* message the AN includes the updated value of SessionSignature and a previously unused PersonalityIndexStore value that is in the range 0 to PersonalityCount-1.

i. Ensure that the AT transmits a *SessionUpdated* message to the AN.

j. Repeat steps f-i such that the AT has negotiated PersonalityCount personalities.

Part 1

k. Instruct the AN to transmit a *SwitchPersonality* message to the AT. Ensure that the AN sets the Hard-Switch field value correctly.

l. Verify that the AT transmits a *SwitchPersonalityAccept* message to the AN.

m. Cause the AT to transmit a *ConfigurationRequest* message. This may be achieved, for example, by causing the AT to negotiate ReservationKKPacketFiltersFwd for a reservation.

n. If the AT transmits a *ConfigurationRequest* message, verify that the sets the PersonalityIndex field in the message to the same value as it received in the *SwitchPersonality* message from the AN.

o. Repeat steps k-n for a value of 0-PersonalityCount-1 for PersonalityIndex field of the *SwitchPersonality* message in step k.

Part 2

p. Cause the AN to transmit a *ConfigurationStart* message to the AT.

q. During the AN initiated state, instruct the AN to transmit a *DeletePersonalityRequest* message to the AT requesting deletion of all stored personalities except the personality with personality Index 0xf. Instruct the AN to set the current InUse personality index as the first entry in the list of proposed personality indices.

r. Verify that the AT transmits a *DeletePersonalityAccept* message and includes the indices of all stored personalities except the current InUse personality and the personality index 0xf.

s. Ensure that the AN transmits a *ConfigurationComplete* message to the AT.

t. Repeat step p-q.

u. Verify that the AT transmits a *DeletePersonalityReject* message to the AN and sets the RejectReason to 'Personality currently InUse'.

v. Instruct the AN to transmit a *SwitchPersonality* message to the AT requesting the AT to switch to one of the deleted personalities.

w. Verify that the AT rejects the SwitchPersonality message and transmits a *SwitchPersonalityReject* message.

5.1.2.3 Minimum Standard

AT shall comply with steps d, l, n, r, u and w.

5.1.3 Soft-Switch and Hard-Switch across Personalities

5.1.3.1 Definition

This test verifies AT behavior for various switch across personalities. The test is divided into 4 parts.

Part 1:

Steps k and m verify that the AT switches the personality when it receives a SwitchPersonality message with HardSwitch field set to '0' and the personalities are soft-switcheable.

Part 2:

Step o verifies that the AT rejects a *SwitchPersonality* message if the SessionSignature carried in the message is out of sync.

Part 3:

Steps q, r, t, and v verify that the AT performs a PersonalitySwitch between soft-switcheable personalities when the HardSwitch field is set to '1'.

Part 4:

Steps x, y, aa, and cc verify that the AT performs a hard-switch correctly.

AN Control Requirements:

The test also requires the AN to negotiate 3 personalities with the AT and cause the AT to switch across these by transmitting a *SwitchPersonality* message. It also requires the AN to set the HardSwitch field of the *SwitchPersonality* message to '0', even though the difference between the personalities contains hard-committable attributes.

Traceability:

[8]

Section 2.5.2.18	SwitchPersonality Message
Section 2.5.2.19	SwitchPersonalityAccept Message
Section 2.5.2.2	ConfigurationRequest Message
Section 2.5.1.16.1	Access Terminal Requirements (AN Initiated State)

[4]

Section 5.5.1.6.1	Access Terminal Transmission Procedure

5.1.3.2 Test Procedure

a. Configure the AN to negotiate three personalities P_a, P_b and P_c. P_a and P_b are soft-switcheable, while is P_c is not soft-switcheable w.r.t P_a and P_b.

b. If the AT has a session with the AN, instruct the AN to close the session.

c. Cause the AT to negotiate a new session with the AN.

d. Ensure that the AT and AN negotiate P_a, P_b and P_c.

e. Start data transfer from the AT to the AN and ensure that AT has data to transmit to the AN for the remainder of the test.

f. If the current InUse personality is not P_a, instruct the AN to transmit a *SwitchPersonality* message to the AT requesting it to switch to P_a.

g. Ensure that the AT transmits a *SwitchPersonalityAccept* message to the AN.

h. Cause the AT to transmit a *ConfigurationRequest* message. This may be achieved, for example, by causing the AT to negotiate ReservationKKPacketFiltersFwd for a reservation.

i. If the AT transmits a *ConfigurationRequest* message, ensure that the AT sets the PersonalityIndex field in the message to the PersonalityIndex of personality P_a.

Part 1: Soft switch with HardSwitch field set to '0'

j. Instruct the AN to transmit a *SwitchPersonality* message to the AT requesting it to switch to personality P_b. Note, HardSwitch field in the *SwitchPersonality* message should be set to '0'.

k. Verify that the AT transmits a *SwitchPersonalityAccept* message to the AN.

l. Cause the AT to transmit a *ConfigurationRequest* message. This may be achieved, for example, by causing the AT to negotiate ReservationKKPacketFiltersFwd for a reservation.

m. If the AT transmits a *ConfigurationRequest* message, verify that the AT sets the PersonalityIndex field in the message to the PersonalityIndex of personality P_b.

Part 2: Soft Switch Reject due to SessionSignature out of sync

n. Instruct the AN to transmit a *SwitchPersonality* message to the AT requesting it to switch to P_a with the SessionSignature field set to an incorrect value.

o. Verify that the AT transmits a *SwitchPersonalityReject* message with RejectReason set to 'SessionSignature out of sync' and includes the correct SessionSignature value in the message.

Part 3: Soft Switch with HardSwitch field set to '1'

p. Instruct the AN to transmit a *SwitchPersonality* message to the AT requesting it to switch to P_a. Instruct the AN to set the HardSwitch field of the *SwitchPersonality* message to '1'.

q. Verify that the AT transmits a *SwitchPersonalityAccept* message to the AN.

r. Verify that the AT closes the connection.

s. Cause the AT to establish a connection with the AN.

t. Verify that the AT establishes the connection with the AN and includes PersonalityIndex for personality P_a in the Route Protocol Header.

u. Cause the AT to transmit a *ConfigurationRequest* message. This may be achieved, for example, by causing the AT to negotiate ReservationKKPacketFiltersFwd for a reservation.

v. If the AT transmits a *ConfigurationRequest* message, verify that the AT sets the PersonalityIndex field in the message to the PersonalityIndex of personality P_a.

Part 4: Hard-Switch with Hard-Switch Set to '1'

w. Instruct the AN to a *SwitchPersonality* message to the AT requesting it to switch to P_c. Ensure that the HardSwitch field of the *SwitchPersonality* message is set to '1'.

x. Verify that the AT transmits a *SwitchPersonalityAccept* message to the AN.

y. Verify that the AT closes the connection.

z. Cause the AT to establish a connection with the AN.

aa. Verify that the AT establishes the connection with the AN and includes PersonalityIndex for personality P_c in the Route Protocol Header.

bb. Cause the AT to transmit a *ConfigurationRequest* message. This may be achieved, for example, by causing the AT to negotiate ReservationKKPacketFiltersFwd for a reservation.

cc. If the AT transmits a *ConfigurationRequest* message, verify that the AT sets the PersonalityIndex field in the message to the PersonalityIndex of personality P_c.

5.1.3.3 Minimum Standard

AT shall comply with steps k, m, o, q, r, t, v, x, y, aa, and cc.

5.1.4 Soft-Commit of Static Attributes at Personality Negotiation

5.1.4.1 Definition

This test verifies that when the negotiation of a new personality updates a static attribute, the AT updates these across all stored personalities. This is verified in part 2. The test also validates the Keep-Alive functionality. This is verified in part 2. Note, in order to reduce the test time, the smallest valid value should be chosen in step a for TimerBasedRegPeriod for personality P_a.

AN Control Requirements:

This test requires the AN to negotiate two personalities with different values of TimerBasedRegPeriod. The AN needs to log the Registration messages transmitted by the AT.

Traceability:

[8]

Section 2.7.1	Simple Attributes
Section 2.5.2.18	SwitchPersonality Message
Section 2.5.2.19	SwitchPersonalityAccept Message
Section 2.5.1.10	Keep Alive Functions
Section 2.5.1.16.1	Access Terminal Requirements (AN Initiated State)

5.1.4.2 Test Procedure

a. Configure the AN to support negotiation of two personalities, P_a and P_b. Personality P_a should negotiate the TimerBasedRegPeriod to a value (I) different than 0x0000 and P_b should negotiate the TimerBasedRegPeriod to 0x0000.

b. If the AT has a session with the AN, instruct the AN to close the session.

c. Cause the AT to negotiate a new session with the AN.

d. Ensure that the local route is also the SessionAnchor route.

e. Configure the AN negotiates the personality P_a.

f. Instruct the AN to transmit a *SwitchPersonality* message instructing the AT to switch to P_a.

g. Ensure that the AT transmits a *SwitchPersonalityAccept* message.

Part 1: Verification of Keep-Alive functionality

h. Verify that the AT transmits a Registration message at every I minute interval.

i. Instruct the AN to negotiate personality P_b and store it in an index different than the PersonalityIndex of P_a. ensure that the AN does not switch to P_b.

Part 2: Verification of update of static attributes at personality negotiation

j. Verify that after the negotiation the AT does not transmit a Registration message in the next 2I minute interval.

5.1.4.3 Minimum Standard

AT shall comply with steps h and j.

5.1.5 Configuration Message Processing

5.1.5.1 Definition

This test verifies that the AT responds to *ConfigurationRequest*, *FastConfigurationRequest* and *CopyConfigurationRequest* messages within $T_{Turnaround}$ time. The 3 parts of the test verify the 3 different messages.

AN Control Requirements:

This test requires the AN to generate *ConfigurationRequest*, *FastConfigurationRequest* and *ConfigurationCopyRequest* messages and to measure the time taken by the AT to respond these messages.

Traceability:

[8]

Section 2.5.1.1 Generic Configuration Procedures

5.1.5.2 Test Procedure

a. If the AT has a session with the AN, instruct the AN to close the session.

b. Cause the AT to negotiate a new session with the AN.

Part 1: *ConfigurationRequest* message response

c. During AN initiated state, instruct the AN to transmit a *ConfigurationRequest* message to the AT.

d. Verify that the AT responds to the *ConfigurationRequest* message by transmitting a *ConfigurationResponse* message within $T_{TurnAround}$ time and sets the TransactionID field in the *ConfigurationResponse* message to the value of TransactionID in the corresponding *ConfigurationRequest* message.

Part 2: *FastConfigurationRequest* message response

e. After the session configuration is over, cause the AN to transmit a *FastConfigurationRequest* message to the AT.

f. Verify that the AT responds to the *FastConfigurationRequest* message by transmitting a *FastConfigurationAccept* or *FastConfigurationReject* message within $T_{TurnAround}$ time and sets the TransactionID field to the value of TransactionID in the corresponding *FastConfigurationRequest* message.

Part 3: *ConfigurationCopyRequest* message response

g. Instruct the AN to start the negotiation of a new personality.

h. During AN initiated state instruct the AN to transmit a *ConfigurationCopyRequest* message to the AT.

i. Verify that the AT responds to the *ConfigurationCopyRequest* message by transmitting *ConfigurationCopyAccept* or *ConfigurationCopyReject* message within a within $T_{TurnAround}$ time and sets the TransactionID field to the value of TransactionID in the corresponding *ConfigurationCopyRequest* message.

5.1.5.3 Minimum Standard

AT shall comply with steps d, f, and i.

5.1.6 SessionSignature Validation

5.1.6.1 Definition

If the SessionSignature field of the message is out of sync, then the AT shall reject *ConfigurationStart* or *FastConfigurationRequest* messages by transmitting a *ConfigurationReject* message with RejectReason set to 'SessionSignature out of sync'.

AN Control Requirements:

This test requires the AN to generate *ConfigurationStart* or *FastConfigurationRequest* message with incorrect value of the SessionSignature field.

Traceability:

[8]

Section 2.4.1.1 2.5.1.1 Generic Configuration Procedures

5.1.6.2 Test Procedure

a. If the AT has a session with the AN, instruct the AN to close the session.

b. Cause the AT to negotiate a new session with the AN.

c. After session configuration has been completed, instruct the AN to either transmit a *FastConfigurationRequest* message or *ConfigurationStart* message with incorrect value of SessionSignature.

d. Verify that the AT responds to the message in step c, by transmitting a *ConfigurationReject* message with RejectReason set to '00000000' (SessionSignature out of sync) and includes the correct value of SessionSignature.

5.1.6.3 Minimum Standard

AT shall comply with step d.

5.1.7 TransactionID Increment

5.1.7.1 Definition

This test verifies that the AT increments TransactionID for each new each new *ConfigurationRequest* message that it transmits.

AN Control Requirements:

The AN is required to log the *ConfigurationRequest* message, *FastConfigurationRequest* message, *ConfigurationCopyRequest* or *AttributeReset* messages transmitted by the AT.

Traceability:

[8]

Section 2.5.1.1	Generic Configuration Procedures
Section 2.5.2.2	ConfigurationRequest
Section 2.5.2.9	FastConfigurationRequest
Section 2.5.2.4	ConfigurationCopyRequest
Section 2.5.2.7	AttributeReset

5.1.7.2 Test Procedure

a. If the AT has a session with the AN, instruct the AN to close the session.

b. Cause the AT to negotiate a new session with the AN.

c. After the session configuration is over, cause the AT to transmit a *FastConfigurationRequest* or *ConfigurationRequest* message.

d. Verify that the AT uses a value of 0 for the TransactionID field of the first *ConfigurationRequest* message, *FastConfigurationRequest* message, *ConfigurationCopyRequest* or *AttributeReset* message.

e. Verify that the AT increments the value of TransactionID field by 1 for each *ConfigurationRequest* message, *FastConfigurationRequest* message, *ConfigurationCopyRequest* or *AttributeReset* message that it transmits.

5.1.7.3 Minimum Standard

AT shall comply with steps d and e.

5.1.8 ATInitiatedConfigurationAllowed set to 0x0000

5.1.8.1 Definition

This test verifies that in open state the AT does not initiate session configuration if ATInitiatedConfigurationAllowed is set to 0x0000.

AN Control Requirements:

The AN should control the negotiation of ATInitiatedConfigurationAllowed.

Traceability:

[8]

Section 2.5.1.14.1 AT Requirements in Open State

5.1.8.2 Test Procedure

a. If the AT has a session with the AN, instruct the AN to close the session.

b. Cause the AT to negotiate a new session with the AN.

c. Ensure that the AN negotiates ATInitiatedConfigurationAllowed to 0x0001.

d. After the session configuration is over, cause the AT to transmit a *FastConfigurationRequest* or ConfigurationRequest message.

e. Ensure that the AT and AN negotiate the session.

f. Instruct the AN not to negotiate ATInitiatedConfigurationAllowed and ensure that the AT and AN continue to use the default value of 0x0000.

g. Create similar conditions that caused the AT to transmit *FastConfigurationRequest* or ConfigurationRequest message in step d.

h. Verify that the AT does not transmit a *FastConfigurationRequest* or *ConfigurationRequest* message.

5.1.8.3 Minimum Standard

AT shall comply with steps h.

5.1.9 AttributeReset Message Response

5.1.9.1 Definition

This test verifies that the AT responds to the *AttributeReset* message by transmitting an *AttributeResetAccept* message. Note, in step b in the test procedure any attribute can be used. The attribute that is reset using the *AttributeReset* message should be chosen such that step is f verifiable. For example, if the ReservationKKStreamRev is reset, data for the reservation will not be transmitted using the stream to which it was previously mapped.

AN Control Requirements:

This test requires the AN to transmit *AttributeReset* message. It is also required to log packets/messages in order to determine whether the *AttributeReset* message has been processed correctly by the AT.

Traceability:

[8]

Section 2.5.1.1	Generic Configuration Procedures
Section 2.5.2.7	AttributeReset
Section 2.5.2.8	AttributeResetAccept

5.1.9.2 Test Procedure

a. If the AT does not have a session with the AN, cause the AT to negotiate a new session with the AN.

b. If required for verification in step f, cause the AT to start using the attribute being reset in test procedure below. For example, AT could start transmitting data for reservation KK through stream mapped to the reservation KK through ReservationKKStreamRev.

c. Instruct the AN to transmit a *ConfigurationStart* message.

d. During the AN initiated state instruct the AN to transmit an *AttributeReset* message to the AT.

e. Verify that the AT responds to the *AttributeReset* message by transmitting an *AttributeResetAccept* message.

f. Verify that the AT starts using the default value for the attribute that was reset in step e.

5.1.9.3 Minimum Standard

AT shall comply with step e, and f.

5.1.10 SessionClose Message Response

5.1.10.1 Definition

This test verifies that the AT responds to the *SessionClose* message by transmitting a *SessionClose* message.

AN Control Requirements:

AN Control should be able to generate *SessionClose* message.

Traceability:

[8]

Section 2.5.1.8	Processing the SessionClose Message

5.1.10.2 Test Procedure

a. If the AT does not have a session with the AN, cause the AT to negotiate a new session with the AN.

b. After the session has been negotiated, instruct the AN to transmit a *SessionClose* message to the AT.

c. Verify that the AT responds to the *SessionClose* message by transmitting a *SessionClose* message.

d. Cause the AT to establish a connection with the AN.

e. Verify that the AT accesses the AN using a RATI.

5.1.10.3 Minimum Standard

AT shall comply with step c, and e.

5.1.11 ConfigurationReject Message Transmission

5.1.11.1 Definition

This test verifies that the AT responds to *ConfigurationStart* or *FastConfigurationRequest* message from the AN by transmitting a *ConfigurationReject* message if it is already configuring a session.

AN Control Requirements:

The AN needs to delay the response to *ConfigurationRequest* message or a *FastConfigurationRequest* message and transmit *ConfigurationStart* message to the AT.

Traceability:

[8]

Section 2.5.1.14.1	Access Terminal Requirements (Open State)
Section 2.7.1	Simple Attributes
Section 2.5.2.2	ConfigurationRequest Message
Section 2.5.2.9	FastConfigurationRequest Message
Section 2.5.2.14	ConfigurationReject Message

5.1.11.2 Test Procedure

a. If the AT does not have a session with the AN, cause the AT to negotiate a new session with the AN.

b. Ensure that the AN negotiates a value of 0x0001 for ATInitiatedConfigurationAllowed with the AT

c. After the session has been negotiated, cause the AT to either transmit a *ConfigurationRequest* message or a *FastConfigurationRequest* message.

d. Instruct the AN to transmit a *FastConfigurationRequest* or a *ConfigurationStart* message to the AT before responding to the message transmitted by the AT in step c.

e. Verify that the AT responds to the message in step d by transmitting a *ConfigurationReject* message with RejectReason set to 'Session Configuration in progress'.

5.1.11.3 Minimum Standard

AT shall comply with step e.

5.1.12 Connection Closed during Session Configuration

5.1.12.1 Definition

This test verifies that the AT discards the InConfiguration instances if connection is closed during session configuration.

AN Control Requirements:

During AN initiated state AN needs to close the connection without transmitting *ConfigurationComplete* message to the AT.

Traceability:

[8]

Section 2.4.1.1	Protocol Initialization for the InConfiguration Protocol Instance
Section 2.5.2.18	SwitchPersonality Message
Section 2.5.2.19	SwitchPersonalityAccept Message
Section 2.5.2.13	ConfigurationStart Message
Section 2.5.2.12	ConfigurationComplete Message
Section 2.5.2.15	DeletePersonalityRequest
Section 2.5.2.16	DeletePersonalityAccept

5.1.12.2 Test Procedure

a. Configure the AN to negotiate two personalities with the AT.

b. If the AT does not have a session with the AN, cause the AT to negotiate a new session with the AN.

c. Ensure that the AN negotiates two personalities with the AT.

d. After the session configuration is complete, instruct the AN to transmit a *ConfiguratioStart* message to the AT.

e. In the AN initiated state, instruct the AN to request deletion of one personality by transmitting a *DeletePersonalityRequest* message. Note, the AN should list the personality currently not in use in the *DeletePersonalityRequest* message.

f. After the AN receives a *DeletePersonalityAccept* message from the AT, instruct the AN to close the connection such that the AT does not receive the *ConfigurationComplete* message.

g. Cause the AT to establish the connection with the AN.

h. Instruct the AN to transmit a *SwitchPersonalityRequest* message with the HardSwitch field set appropriately.

i. Verify that the AT transmits a *SwitchPersonalityAccept* message in response to the *SwitchPersonalityRequest* message.

j. Cause the AT to transmit a *ConfigurationRequest* message. This may be achieved, for example, by causing the AT to negotiate ReservationKKPacketFiltersFwd for a reservation.

k. If the AT transmits a *ConfigurationRequest* message, verify that the sets the PersonalityIndex field in the message to the same value that it received in the *SwitchPersonality* message from the AN.

5.1.12.3 Minimum Standard

AT shall comply with steps i and k.

5.1.13 FastConfigurationRequest Processing

5.1.13.1 Definition

This test verifies that the processing of *FastConfigurationRequest* message at the AT. *FastConfigurationRequest* message in step f should update an attribute that can be verified in step i. For example TimerBasedRegistration could be negotiated to a small value and Registration message generation from the AT can be used for verification.

AN Control Requirements:

The AN needs to transmit *FastConfigurationRequest* message to the AT. It also needs to log messages/packets that can be used for verification of *FastConfigurationRequest* processing at the AT.

Traceability:

[8]

Section 2.5.1.14.1	Access Terminal Requirements (Open State)
Section 2.5.1.18.1	Access Terminal Requirements (AN Fast Configuration State)
Section 2.7.1	Simple Attributes

5.1.13.2 Test Procedure

a. If the AT does not have a session with the AN, cause the AT to negotiate a new session with the AN.

b. Ensure that the AN negotiates a value of 0x0001 for ATInitiatedConfigurationAllowed with the AT.

c. After the AT session configuration is complete, instruct the AN to transmit a *FastConfigurationRequest* message to the AT with invalid values for an attribute.

d. Verify that in response to the *FastConfigurationRequest* message the AT transmits a *FastConfigurationReject* message to the AN.

e. Instruct the AN to transmit a *FastConfigurationRequest* message to the AT with default value included in the proposed list of values for all attributes listed in the message.

f. Verify that in response to the *FastConfigurationRequest* message the AT transmits a *FastConfigurationAccept* message to the AN.

g. Ensure that the AN transmits a *ConfigurationComplete* message to the AT with the updated value of the SessionSignature.

h. Verify that the AT transmits a *SessionUpdated* message to all the AN in the Route Set.

i. Verify that after session configuration is over, the AT starts using the negotiated values of attributes contained in the *FastConfigurationRequest* message.

5.1.13.3 Minimum Standard

AT shall comply with step d, f, hand i.

5.2 Basic Session Control Protocol Access Network Tests

5.2.1 Initial Session Configuration

5.2.1.1 Definition

This test verifies the initial session configuration for the AN.

Part 1 verifies the transmission of *ConfigurationStart* message.

Part 2 verifies the response during AT initiated state.

Part 3 verifies the transmission of *ConfigurationComplete* at the end of AN initiated state.

Part 4 verifies the personality switch during open state.

Traceability:

[8]

Section 2.4.1.1	Procedures and Messages for the InConfiguration Instance of the Protocol, Other Procedures
Section 2.5.1.1	Generic Configuration Procedures
Section 2.5.2.13	ConfigurationStart Message
Section 2.5.2.2	ConfigurationRequest Message
Section 2.5.2.12	ConfigurationComplete Message
Section 2.5.1.15.1	Access Terminal Requirements (AT Initiated State)
Section 2.5.1.9	Processing the Reconfigured Indication
Section 2.5.1.16.1	Access Terminal Requirements (AN Initiated State)
Section 2.5.1.14.1	Access Terminal Requirements (Open State)

5.2.1.2 Test Procedure

a. If the AT has a session with the AN, instruct the AN to close the session.

b. Cause the AT to negotiate a new session with the AN. Ensure that the AT accesses the AN using an IPSI.

Part 1

c. Verify that the AN transmits a *ConfigurationStart* message with PersonalityIndex field set to 0xf and SessionSignature field set to 0x0000.

d. Ensure that the AT transmits a *ConfigurationRequest* message negotiating the HardwareID, ATSupportedProtocolSetIDs and ATSupportedApplicationProtocolIDs attributes with the AN.

e. Ensure that the AT uses a value of 0xf for the PersonalityIndex field of the *ConfigurationRequest* message.

Part 2

f. Verify that the AN transmits a *ConfigurationResponse* message accepting the proposed value for the values for HardwareID, ATSupportedProtocolSetIDs and ATSupportedApplicationProtocolIDs attributes.

g. Ensure that the AT transmits a *ConfigurationComplete* message after the AT initiated session negotiation is complete.

Part 3

h. Verify that the AN transmits a *ConfigurationComplete* message to the AT and sets the value of SessionSignature to 0x0001 and includes a PersonalityIndexStore value in the range 0 to PersonalityCount-1.

i. Ensure that the AT transmits a *SessionUpdated* message to the AN and includes the same SessionSignature that it received in the *ConfigurationComplete* message from the AN.

Part 4

j. Verify that the AN transmits a *SwitchPersonality* message to the AT with the same value for PersonalityIndex field as that of the PersonalityIndexStore field in the *ConfigurationComplete* message.

5.2.1.3 Minimum Standard

AN shall comply with steps c, f, h and j.

5.2.2 TransactionID Increment

5.2.2.1 Definition

This test verifies that the AN increments TransactionID for each new each new *ConfigurationRequest* or *FastConfigurationRequest* message that it transmits.

Traceability:

[8]

Section 2.5.1.1	Generic Configuration Procedures
Section 2.5.2.2	ConfigurationRequest message
Section 2.5.2.9	FastConfigurationRequest message
Section 2.5.2.4	ConfigurationCopyRequest message
Section 2.5.2.7	AttributeReset message

5.2.2.2 Test Procedure

a. If the AT has a session with the AN, instruct the AN to close the session.

b. Cause the AT to negotiate a new session with the AN.

c. After the session configuration has been completed, cause the AN to transmit a *FastConfigurationRequest* message.

d. Verify that the AN uses a value of 0 for the TransactionID field of the first *ConfigurationRequest* message, *FastConfigurationRequest* message, *ConfigurationCopyRequest* or *AttributeReset* message that it transmits.

e. Verify that the AN increments the value of TransactionID field by 1 for each *ConfigurationRequest* message, *FastConfigurationRequest* message, *ConfigurationCopyRequest* or *AttributeReset* message that it transmits.

5.2.2.3 Minimum Standard

AN shall comply with steps d and e.

No Text.

6 ROUTE CONTROL PROTOCOL TESTS

This section verifies the route establishment, route close, UATI Assigment, DAP move, and LinkID assignment functionalities of Route Control Protocol.

6.1 Route Control Protocol Access Terminal Tests

6.1.1 Initial Route Establishment

6.1.1.1 Definition

This test verifies the initial access and route establishment for the access terminal. Specifically, *RouteOpenRequest*, *UATIComplete* and *RouteMap* message transmission is verified.

AN Control Requirement:

The access network should be able to operate in centralized SRNC mode. Access network should be able to transmit *ConnectionClose* and *RouteMapRequest* messages.

Traceability:

[9]

Section 2.5.1.1	Protocol Initialization for the InUse Protocol Instance
Section 2.5.1.8.1	WaitingToOpen State requirements for Access Terminal
Section 2.5.2.1	RouteOpenRequest message
Section 2.5.2.11	RouteCreate message
Section 2.5.2.6	RouteMap message
Section 2.5.2.10	UATIComplete message

[4]

Section 5.5.1.4	Route Protocol Header

[7]

Section 5.5.1	ConnectionClose message

6.1.1.2 Test Procedure

a. If the access terminal has a session with the access network, instruct the access network to close the session.

b. Cause the access terminal to establish a session with the access network.

c. Verify that the access terminal transmits a *RouteOpenRequest* message to the access network with the RouteOpenRequestReason field set to 0x0.

d. Instruct the access network to transmit a *RouteCreate* message with the ANID of the SRNC and only 1 PSI in the *RouteCreate* message.

e. Ensure that the access network transmits a *RouteOpenAccept* message to the access terminal.

f. Verify that the access terminal transmits a *RouteOpenRequest* message to the SRNC with the RouteOpenRequestReason field set to 0x3 and includes the RouteCreate header with PSI set to the value provided in the *RouteCreate* message received from the local access network.

g. Ensure that the SRNC transmits a *RouteOpenAccept* message to the access terminal and assigns a UATI to the access terminal.

h. Verify that the access terminal transmits a *UATIComplete* message to the SRNC.

i. Verify that the access terminal transmits a *RouteMap* message to the access network and includes RouteID values of '0000000' and '0000001' for the local and SRNC access network respectively.

j. Instruct the access network to transmit a *ConnectionClose* message to the access terminal.

k. Ensure that the access terminal transmits a *ConnectionClose* message to the access network.

l. Cause the access terminal to establish a connection with the access network.

m. Verify that the access terminal transmits a *RouteOpenRequest* message to the access network with SessionAnchorRouteID field set to '0000001'.

n. Instruct the access network to transmit a *RouteMapRequest* message to the access terminal.

o. Verify that the access terminal transmits a *RouteMap* message to the access network and includes RouteID values of '0000002' and '0000001' for the local and SRNC access network respectively.

6.1.1.3 Minimum Standard

Access terminal shall comply with steps c, f, h, i, m, o.

6.1.2 *RouteClose* and Transmission and Response

6.1.2.1 Definition

This test verifies that the access terminal deletes a route when it receives a *RouteClose* message from the access network.

AN Control Requirements:

AN should be able to transmit a *RouteClose* message to the access terminal.

Traceability:

[9]

Section 2.5.2.3 RouteClose message

6.1.2.2 Test Procedure

a. If the access terminal has a session with the access network, instruct the access network to close the session.

b. Cause the access terminal to establish a connection with the access network.

c. Instruct the local access network to transmit a *RouteClose* message to the access terminal.

d. Verify that the access terminal transmits a *RouteClose* message to the access network.

6.1.2.3 Minimum Standard

Access terminal shall comply with step d.

6.1.3 DAPMoveRequest Transmission

6.1.3.1 Definition

This test verifies that the access terminal transmits a *DAPMoveRequest* message requesting DAP to be moved to the current FLSE after layer 2 handoff.

AN Control Requirements:

AN should be able to set the DAPRefreshTimer value in the *DAPAssignment* message.

Traceability:

[9]

Section 2.5.1.6.1	Access Terminal DataAtachmentPoint Procedures
Section 2.5.2.4	DAPMoveRequest message
Section 2.5.2.5	DAPAssignment message

[7]

Section 5.5.1	ConnectionClose message

6.1.3.2 Test Procedure

a. Connect the access terminal to the access network as shown in figure 1.

b. Ensure that the access terminal uses a value of ‘1’ for ATassistedDAPMove field of the ATassistedDAPMoveMode attribute. Note that the access terminal uses the default value of ‘1’ for ATassistedDAPMove field of the ATassistedDAPMoveMode attribute.

c. Cause the access terminal to establish a connection with the access network 1.

d. Cause the access terminal to open a route with the access network 2.

e. Ensure that the access network 2 assigns a LinkID to the access terminal.

f. Start data transfer on the forward link using access network 1.

g. Change the channel conditions such that the access terminal changes the FLSE from access network 1 to access network 2.

h. Ensure that the access network does not transmit a *DAPMoveRequestRequest* message or a *DAPAssignment* message to the access terminal.

i. Verify that the access terminal transmits *DAPMoveRequest* message to the access network 2 after a period of L2HOtoDAPMoveRequestTimer has elapsed from the time FLSE was changed to access network 2.

j. Instruct the access network to transmit *DAPAssignment* message to the access terminal with a DAPRefreshTimer value of 5 seconds.

k. Verify that the access terminal transmits a DAPMoveRequest to the access network 5 seconds after receiving the *DAPAssignment* message transmitted to the access terminal in step j.

6.1.3.3 Minimum Standard

Access terminal should comply with step i and k.

6.1.4 UATIAssignment Message Processing

6.1.4.1 Definition

This test verifies that the access terminal's processing of *UATIAssignment* message.

AN Control Requirements:

AN should control the assignment of the UATI to the access terminal and be able to assign it through *UATIAssignment* message. AN should be able to transmit a *ConnectionClose* message to the access terminal.

Traceability:

[9]

Section 2.5.1.4	UATI Assignment Processing
Section 2.5.2.9	UATIAssignment message
Section 2.5.2.10	UATIComplete message

[7]

Section 5.5.1	ConnectionClose message

6.1.4.2 Test Procedure

a. If the access terminal has a session with the access network, instruct the access network to close the session.

b. Cause the access terminal to establish a session with the access network.

c. Instruct the access network to assign a UATI to the access terminal through a *UATIAssignment* message. The access network should not include the UATI assignment in the *RouteOpenAccept* or *DAPAssignment* messages.

d. Verify that the access terminal transmits a *UATIComplete* message to the access network.

e. Instruct the access network to transmit a *ConnectionClose* message to the access terminal.

f. Ensure that the access terminal transmits a *ConnectionClose* message to the access network.

g. Cause the access terminal to establish a connection with the access network.

h. Verify that the access terminal access the access network with the UATI assigned in *UATIAssignment* message transmitted to the access terminal in step c.

6.1.4.3 Minimum Standard

Access terminal shall comply with steps d and h.

6.1.5 LinkIDAssignment Message Processing

6.1.5.1 Definition

This test verifies that the access terminal's processing of *LinkIDAssignment* message.

AN Control Requirements:

AN should be able to assign the LinkID to the AT through the *LinkIDAssignment* message.

Traceability:

[9]

Section 2.5.1.5	LinkID Procedures
Section 2.5.2.13	LinkIDAssignment message

6.1.5.2 Test Procedure

a. Ensure that the access terminal has an assigned IP address and no assigned LinkID on the anchor route.

b. If the access terminal has an open connection with the access network, instruct the access network to close the connection with the access network.

c. Cause the access terminal to open a connection with the access network by generating IP data for transmission.

d. Instruct the access network not to assign a LinkID to the access terminal in the *RouteOpenAccept* message.

e. Instruct the access network to transmit *LinkIDAssignment* message to the access terminal.

f. Verify that the access terminal does not start IP data transmission until it receives *LinkIDAssignment* message.

6.1.5.3 Minimum Standard

Access terminal shall comply with step f.

6.2 Basic Route Control Protocol Access Network Tests

6.2.1 AN Initial Route Establishment Test

6.2.1.1 Definition

This test verifies the initial access and route setup for the access network.

Traceability:

[9]

Section 2.5.1.1	Protocol Initialization for the InUse Protocol Instance
Section 2.5.1.8.1	WaitingToOpen State requirements for Access Terminal
Section 2.5.2.1	RouteOpenRequest message
Section 2.5.2.11	RouteCreate message
Section 2.5.2.6	RouteMap message

[4]

Section 5.5.1.4	Route Protocol Header

6.2.1.2 Test Procedure

a. If the access terminal has a session with the access network, instruct the access network to close the session.

b. Cause the access terminal to establish a connection with the access network.

c. Ensure that the access terminal transmits a *RouteOpenRequest* message to the access network and includes the RouteCreate and ATI headers in the Route Protocol packet.

d. Verify that the access network transmits a *RouteOpenAccept* message to the access terminal.

e. If the access network is configured with centralized SRNC, verify that the access network transmits a *RouteCreate* message to the access terminal.

f. If the access terminal receives *RouteCreate* message, ensure that the access terminal transmits a *RouteOpenRequest* message to the SRNC.

g. Verify that the access network assigns a UATI to the access terminal.

h. If the access network is configured with centralized SRNC, ensure that the access terminal transmits a *RouteMap* message to the access network.

i. If the access network receives a *RouteMap* message, verify that the access network transmits a *RouteMapAck* message to the access terminal.

6.2.1.3 Minimum Standard

Access network shall comply with steps d, e, g, and i.

No Text.

7 CONNECTION CONTROL PROTOCOL TESTS

This section provides signaling conformance tests for protocols belonging to the Connection Control Plane of UMB. Specifically, tests for Basic Air-Link Management Protocol, Basic Initialization State Protocol, Basic Idle State Protocol, Overhead Message Protocol, and Basic Active Set Management Protocol are provided.

7.1 Basic Air Link Management Protocol Access Terminal Tests

7.1.1 Redirect Message processing

7.1.1.1 Definition

This test verifies the initial access terminal processes the *Redirect* message transmitted by the access network.

Traceability:

[7]

Section 4.4.4.5	Procedure for ChannelBand Selection
Section 2.5.1	Redirect message

7.1.1.2 Test Procedure

a. Connect the access terminal to the access network as shown in figure 1.

b. AN 1 and AN 2 use different ChannelBand.

c. Ensure that the access terminal selects ChannelBand of access network 1 for access.

d. Cause the access terminal to access AN 1.

e. Instruct the AN 1 to transmit a *Redirect* message to the access terminal specifying the ChannelBand used by AN 2.

f. Verify that the access terminal accesses AN 2.

7.1.1.3 Minimum Standard

Access terminal should comply with step f.

7.1.2 Registration Message Transmission

7.1.2.1 Definition

This test verifies the transmission of *Registration* message under various conditions. Note, TimerBased registration is tested in Session Control Protocol tests.

Traceability:

[7]

Section 2.4.4.1.1.1	Zone Based Registration

Section 2.4.4.1.1.2	Distance Based Registration
Section 2.4.4.1.1.4	NetworkCode Based Registration
Section 2.4.4.1.1.1	Registration after loss of coverage
Section 2.4.7.2	Procedures for Maintaining Registration

7.1.2.2 Test Procedure

a. Connect the access terminal to the access network as shown in figure 1.

b. Configure the sectors in AN 1 and AN 2 with different PrimaryRegistrationZoneCode values. Ensure that the AN 1 and AN 2 are configured with same NetworkCode values and the Longitude and Latitude values such that the distance between AN 1 and AN 2 is smaller than RegistrationRadiusSmall.

c. Cause the access terminal to acquire the access network.

d. Vary the channel conditions such that the access terminal performs an idle handoff between the sectors of AN 1 and AN 2. Note, the handoff can be in either direction.

e. Verify that the access terminal transmits a *Registration* message.

f. Configure the Longitude and Latitude values of AN 1 and AN 2 such that the distance between the two networks is greater than RegistrationRadiusLarge. Ensure that the AN 1 and AN 2 are configured with same NetworkCode and PrimaryRegistrationZoneCode values.

g. Repeat steps c-d.

h. Verify that the access terminal transmits a *Registration* message.

i. Configure the sectors in AN 1 and AN 2 with different NetworkCode values. Ensure that the AN 1 and AN 2 are configured with same PrimaryRegistrationZoneCode values and the Longitude and Latitude values such that the distance between AN 1 and AN 2 is smaller than RegistrationRadiusSmall.

j. Repeat steps c-d.

k. Either disconnect the access terminal from AN 2, or configure the sectors in AN 1 and AN 2 with the same PrimaryRegistrationZoneCode and NetworkCode values, and Longitude and Latitude values such that the distance between AN 1 and AN 2 is smaller than RegistrationRadiusSmall.

l. Verify that the access terminal transmits a *Registration* message.

m. Cause the access terminal to access AN.

n. Vary the channel conditions such that the access terminal looses the access network and enters initialization state.

o. After InitializationStateRegistrationTimeout, vary the channel conditions such that the access terminal reacquires the AN.

p. Verify that the access terminal transmits a *Registration* message.

7.1.2.3 Minimum Standard

Access terminal shall comply with steps e, h, l and p.

7.1.3 ConnectionKeepAlive Message Transmission

7.1.3.1 Definition

This test verifies the transmission of *ConnectionKeepAlive* messages by the access terminal.

Traceability:

[7]

Section 2.5.1.1 Protocol Initialization for the InUse Protocol Instance

7.1.3.2 Test Procedure

a. Cause the access terminal to establish a connection with the access network.

b. Verify that the access terminal transmits a *ConnectionKeepAlive* message to the access network after every ConnectionKeepAlivePeriod interval.

7.1.3.3 Minimum Standard

Access terminal shall comply with step b.

7.2 Basic Initialization State Protocol Access Terminal Tests

7.2.1 UnSupported IPSI

7.2.1.1 Definition

This test verifies that the access terminal does not access an access network if it is unable to support any of the InitialProtocolSetIdentifier fields specified in the SystemInfo Block.

Traceability:

[7]

Section 3.4.6 Read SystemInfo State

Section 6.5.2 SystemInfo Block

7.2.1.2 Test Procedure

a. Connect the access terminal to the access network.

b. Configure the SystemInfoBlock of the access network to transmit IPSI values that are not supported by the access terminal.

c. Cause the access terminal to attempt accessing the access network.

d. Verify that the access terminal does not access the access network.

7.2.1.3 Minimum Standard

Access terminal should comply with step d.

7.3 Basic Idle State and Connected State Protocol Access Terminal Tests

7.3.1 ConnectionClose and Page Processing

7.3.1.1 Definition

This test verifies the initial access terminal transmits a *ConnectionClose* message in response to *ConnectionClose* message. It also verifies that the access terminal can receive Page and QuickPage from the access network

Traceability:

[7]

Section 2.1	Overview of Basic Air-Link Management Protocol
Section 5.5.1	ConnectionClose message
Section 4.4.7.1	Access Terminal Requirements in Monitor State

[9]

Section 2.5.2.1	RouteOpenRequest message

[3]

Section 3.5.8.1	Procedure for Reception of the QuickPageBlock
Section 6.5.5.1.3	Header Processing

7.3.1.2 Test Procedure

a. Cause the access terminal to establish a connection with the access network by sending a page to the access terminal.

b. Instruct the access network to transmit a *ConnectionClose* message to the access terminal.

c. Verify that the access terminal transmits a *ConnectionClose* message in response to the *ConnectionClose* message.

d. Cause the access network to transmit a QuickPage and Page to the access terminal.

e. Verify that the access terminal transmits a *RouteOpenRequest* message to the access network with the RouteOpenRequestReason field set to 0x1.

7.3.1.3 Minimum Standard

Access terminal shall comply with steps c and e.

7.3.2 AN Basic Connected State Protocol Tests

None.

7.4 Overhead Message Protocol Access Terminal Tests

7.4.1 ChannelInfoValidityTimers Functionality

7.4.1.1 Definition

This test verifies that the access terminal closes the connection if ChannelInfoValidityTimer

Traceability:

[7]

Section 6.4.5.1.6	Procedures for Validity Timers
Section 6.4.5.2.5	Maintaining ChannelInfoExpiryTime

7.4.1.2 Test Procedure

a. Configure the ChannelInfoValidityIndex to the smallest configurable value.

b. Cause the access terminal to establish a connection with the access network and start transmitting data.

c. Instruct the access network to stop transmitting all overhead parameters messages.

d. Wait for ChannelInfoExpiryTime to occur.

e. Instruct the access network to continue assignments on the reverse link (RLAB) to the access terminal.

f. Verify that the access terminal stops transmitting on the reverse link.

7.4.1.3 Minimum Standard

Access terminal should comply with step f.

7.5 Basic Active Set Management Protocol Access Terminal Tests

7.5.1 PilotReport Transmission Tests

7.5.1.1 Definition

This test verifies that the access terminal transmits the *PilotReport* message when pilot is added or dropped from the active set, or if the access terminal receives a *PilotReportRequest* message.

Traceability:

[7]

Section 7.4.3.3	Active Set Management
Section 7.4.3.2	Pilot Drop Timer Maintenance
Section 7.4.7.3	Pilot Strength Report Rules
Section 7.5.1	PilotReport message

Section 7.5.10	PilotReportRequest
Section 7.7.2.1	SetManagementParameters Attribute
Section 7.5.6	MACResourceStatus message

7.5.1.2 Test Procedure

a. Connect the access terminal to the access network as shown in figure 1.

b. Ensure that the pilot from the sector in AN 2 is received at a level lower than PilotAdd threshold.

c. Cause the access terminal to establish a connection with AN 1.

d. Vary the channel conditions such that the pilot from sector in AN 2 is received at a level higher than PilotAdd threshold.

e. Verify that the access terminal transmits a *PilotReport* message and *MACResourceStatus* message to the access network.

f. Ensure that the AN 2 assigns a MACID to the AT.

g. Instruct the AN 1 to transmit a *PilotReportRequest* message to the access terminal.

h. Verify that the access terminal transmits a *PilotReport* message to the AN 1.

i. Vary the channel conditions such that the pilot from the sector of AN 2 is received at a level lower than PilotDrop threshold.

j. Wait for PilotDropTimer duration.

k. Verify that the access terminal transmits a *PilotReport* message and drops the pilot from AN 2 from its active set.

7.5.1.3 Minimum Standard

Access terminal should comply with step e, h, and k.

7.5.2 VCQI Message Transmission Test

7.5.2.1 Definition

This test verifies that the access terminal periodically transmits the VCQIReport message.

Traceability:

[7]

Section 7.4.7.4	VCQI Report Rules
Section 7.4.3.2	VCQIReport message

7.5.2.2 Test Procedure

a. Cause the access terminal to establish a connection with AN 1.

b. Verify that the access terminal transmits VCQIReport message to the access network with non-zero NumPilots field at least once every MaxVCQIReportInterval and at most once every MinVCQIReportInterval.

7.5.2.3 Minimum Standard

Access terminal should comply with step b.

7.5.3 MACResourceAck Message Transmission Test

7.5.3.1 Definition

This test verifies that the access terminal transmits a *MACResourceAck* when it receives a *MACResourceAssignment* message from the access network.

Traceability:

[7]

Section 7 7.4.7.5	Processing the MACResourceAssignment Message
Section 7.5.5	MACResourceAck message

7.5.3.2 Test Procedure

a. Cause the access terminal to establish a connection with AN 1.

b. Verify that the access terminal transmits a *PilotReport* message to the access network.

c. Ensure that the access network transmits a *MACResourceAssignment* message to the access terminal.

d. Verify that the access terminal transmits a *MACResourceAck* message to the access network.

7.5.3.3 Minimum Standard

Access terminal should comply with steps b and d.

7.5.4 SectorIDResponse Transmission Tests

7.5.4.1 Definition

This test verifies that the access terminal transmits a *SectorIDResponse* message to the access network in response to a *SectorIDRequest* message.

Traceability:

[7]

Section 7.4.3.3	Active Set Management
Section 7.5.11	SectorIDRequest message
Section 7.5.12	SectorIDResponse message

7.5.4.2 Test Procedure

a. Connect the access terminal to the access network as shown in figure 1.

b. Ensure that the pilot from the sector in AN 2 is received at a level lower than PilotAdd threshold.

c. Cause the access terminal to establish a connection with AN 1.

d. Wait for access terminal to receive the overhead parameters message.

e. Vary the channel conditions such that the access terminal adds the pilot from the sector of AN 2 to its active set.

f. Instruct the AN 2 to transmit a *SectorIDRequest* message to the access terminal, requesting the SectorID of the pilot associated with AN 1.

g. Verify that the access terminal transmits a *SectorIDResponse* message to AN 2.

7.5.4.3 Minimum Standard

Access terminal should comply with step g.

8 SECURITY FUNCTIONS TESTS

8.1 Basic Key Exchange Protocol Access Terminal Tests

8.1.1 KeyRequest and KeyComplete message transmission

8.1.1.1 Definition

This test verifies the access terminal's transmission of *KeyRequest* and *KeyComplete* messages.

Traceability:

[6]

Section 4.5.2.1	KeyRequest message
Section 4.5.2.2	KeyResponse message
Section 4.5.2.3	KeyComplete message
Section 4.5.2.5	InitiateKeyRequest message
Section 4.5.1.4.2	Processing a KeyResponse message

8.1.1.2 Test Procedure

a. If the access terminal has a session with the access network, instruct the access network to close the session.

b. Cause the access terminal to establish a session with the access network.

c. Instruct the access network to transmit an *InitiateKeyRequest* message to the access terminal.

d. Verify that the access terminal transmits a *KeyRequest* message to the access network. Note the *KeyRequest* message may be transmitted before the *InitiateKeyRequest* message is received at the access terminal.

e. Ensure that the access network transmits a *KeyResponse* message to the access terminal.

f. Verify that the access terminal transmits a *KeyComplete* message to the access network with Result field set to 0x00 (Key exchange successful).

g. Repeat steps a-c.

h. Ensure that the access terminal transmits a *KeyRequest* message to the access network.

i. Instruct the access network to transmit a *KeyResponse* message to the access terminal with an incorrect value of the PairwiseMasterKeyID feild.

j. Verify that the access terminal transmits a *KeyComplete* message to the access network with Result field set to 0x03 (PairwiseMasterKey not found).

k. Repeat steps a-c.

l. Ensure that the access terminal transmits a *KeyRequest* message to the access network.

m. Instruct the access network to transmit a *KeyResponse* message to the access terminal with an incorrect value of the MessageIntegrityCode field.

n. Verify that the access terminal transmits a *KeyComplete* message to the access network with Result field set to 0x01 (MessageIntegrityCode failed).

o. Repeat steps a-c.

p. Ensure that the access terminal transmits a *KeyRequest* message to the access network.

q. Instruct the access network to transmit a *KeyResponse* message to the access terminal and include a PSI that is not supported by the access terminal. Note, access network should include the current in-use PSI in the *KeyResponse* message.

r. Verify that the access terminal transmits a *KeyComplete* message to the access network with Result field set to 0x02 (MessageIntegrityCode successful, but ProtocolSetIdentifiers verification failed).

s. Repeat steps a-c.

t. Ensure that the access terminal transmits a *KeyRequest* message to the access network.

u. Instruct the access network to transmit a *KeyResponse* message to the access terminal and include all the PSI correctly, with the first PSI being different than the current In-Use PSI.

v. Verify that the access terminal transmits a *KeyComplete* message to the access network with Result field set to 0x04 (MessageIntegrityCode successful, ProtocolSetIdentifiers verification successful, but verification of ProtocolSetIdentifier in use failed).

8.1.1.3 Minimum Standard

Access terminal shall comply with steps d, f, j, n, r, and v.

9 FIGURES

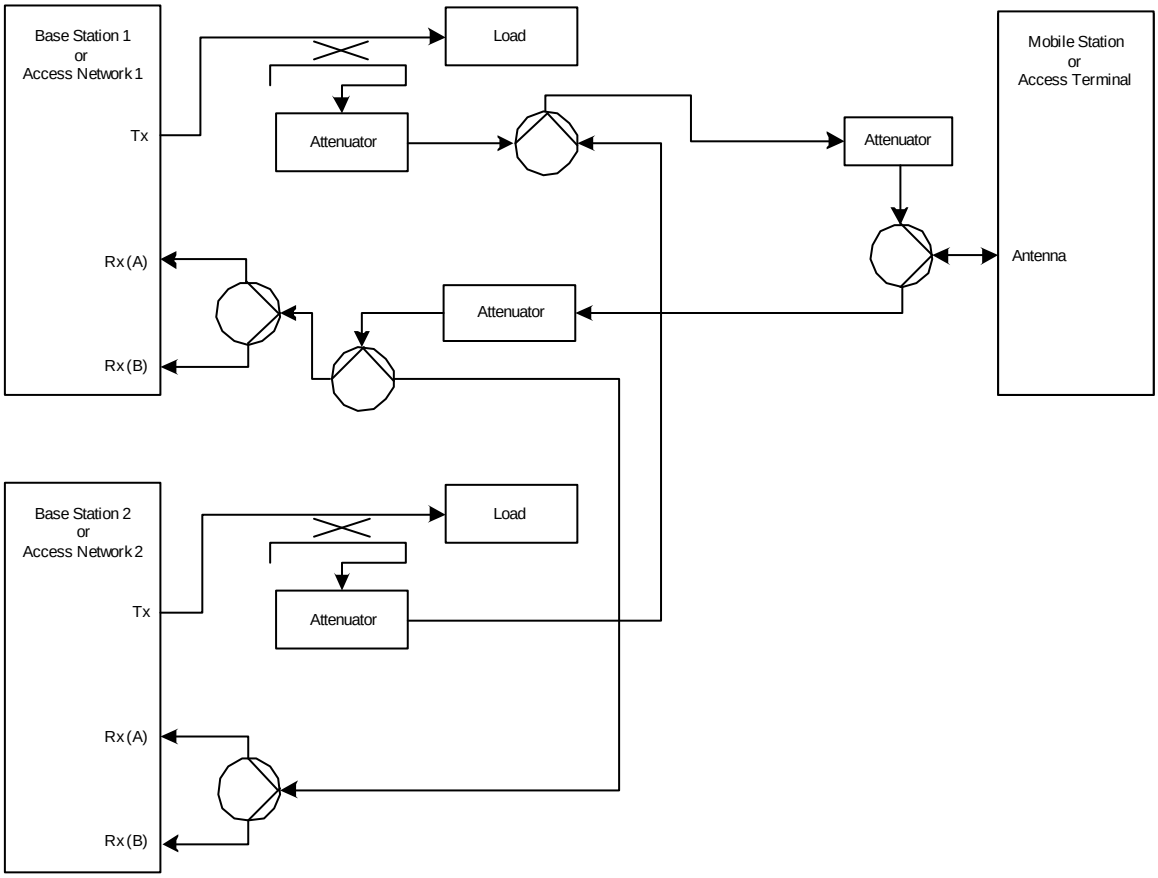

Figure 1. Set-Up for Route Contro Protocol handoff tests and Connection Control Protocol Tests that require two AN