3GPP2 C.S0090-0

Version 1.0

Date: March 30 2009

3RD GENERATION PARTNERSHIP PROJECT 2 "3GPP2"

Test Application Specification for cdma2000 Ultra Mobile Broadband (UMB)

No Text.

CONTENTS

CONTENTS

FIGURES

1 No Text.

TABLES

No Text.

FOREWORD

(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 access terminal minimum performance tests in a factory/laboratory environment. It also allows measurements of certain forward and reverse link performances in a field environment.

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

No Text.

SCOPE

(This scope is not part of this standard)

The technical requirements specified in this document meet the following objectives. They provide a set of procedures that the access terminal and the access network can use

- To conduct the access terminal minimum performance and certain signaling conformance tests in a factory/laboratory environment.
- To conduct measurement of certain forward and reverse link performances in a field environment.

These technical requirements form a compatibility standard for test applications in Ultra Mobile Broadband™ (UMB™)[2] air interface standards. These requirements ensure that a compliant access terminal and a compliant access network can interoperate to execute tests in meeting the above objectives.

This specification is primarily oriented toward requirements necessary for the design and implementation of access terminals. As a result, detailed procedures are specified for access terminals to ensure a uniform response to all access networks. Access network procedures, however, are specified only to the extent necessary for compatibility with those specified for the access terminal.

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

1 No Text.

REFERENCES

NORMATIVE REFERENCES

The following standards and specifications contain provisions, which, through reference in this text, constitute provisions of this standard. 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.

[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.*

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

INFORMATIVE REFERENCES

[11] *C.R1001-F version 1.0, Administration of Parameter Value Assignments for cdma2000 Spread Spectrum Standards*, 2007.

No Text.

1 OVERVIEW

1.1 Introduction

These technical requirements form a compatibility standard for test applications in Ultra Mobile Broadband™ (UMB™)[3] air interface standards. These requirements ensure that a compliant access terminal and a compliant access network can interoperate to execute tests in meeting the objectives stated in 1.2. While the details of the tests are beyond the scope of this document, an informative section illustrating some examples is provided.

This specification is primarily oriented toward requirements necessary for the design and implementation of access terminals. As a result, detailed procedures are specified for access terminals to ensure a uniform response to all access networks. Access network procedures, however, are specified only to the extent necessary for compatibility with those specified for the access terminal.

1.2 Objectives

The Test Application Protocol specified in this document meet the following objectives. It provides a set of procedures that the access terminal and the access network can use,

- To conduct the access terminal minimum performance and certain signaling conformance tests in a factory/laboratory environment.
- To conduct measurement of certain forward and reverse link performances in a field environment.

1.3 Requirements Language

Compatibility, as used in connection with this standard, is understood to mean: Any access terminal can obtain service through any access network conforming to this standard. Conversely, all access networks conforming to this standard can service access terminals.

"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.

---

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

1.4 Protocol Overview

The following is a brief overview of the Test Application Protocol. A complete description of the components can be found in the following sections.

The Test Application Protocol specifies the procedures and messages to control the Forward and Reverse Traffic Channels and to configure reverse channels associated with the Forward Traffic Channel. It specifies generation and transmission of test packets sent on the Forward and Reverse Traffic Channels for the purpose of testing these channels. It also specifies statistics collection procedures for certain statistics as seen at the access terminal. This protocol operates with the Basic Physical Layer Protocol [2].

1.5 Basic Protocol Numbers

The Test Application Protocol shall use the Application Protocol ID subtype value specified in [1].

1.6 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 Test Application Protocol (TAP) Specifications: This section describes the procedures and messages of the Test Application Protocol.
- Section 3 Test Application Example Flow Diagrams: This section describes some examples using the procedures and messages specified in the document. This section is for informational purpose only.

1.7 Acronyms

Acronym	Expansion
AC	Access Channel
AN	Access Network
ASP	Active Set Pilot
AT	Access Terminal
CC	Control Channel
CDMA	Code Division Multiple Access
FCS	Frame Check Sequence
FER	Frame Erasure Rate
FL	Forward Link
FTC	Forward Traffic Channel
MAC	Media Access Control
NA	Not Applicable
PER	Packet Error Rate
RL	Reverse Link
RTC	Reverse Traffic Channel
SS	Serving Sector
TAP	Test Application Protocol

Table 1.7-1 List of Acronyms

1.8 Notation

A[i] The $(i+1)^{th}$ element of array A. The first element of the array is A[0].

A[i,j] The $(j+1)^{th}$ element of $(i+1)^{th}$ row of matrix A. The first element of the matrix is A[0,0].

x mod y Indicates the remainder after dividing x by y: $x \bmod y = x - (y \times \lfloor x/y \rfloor)$.

Min(a,b) The minimum of the two arguments a and b.

1 No Text.

2 TEST APPLICATION PROTOCOL SPECIFICATION

2.1 Overview

The Test Application Protocol (TAP) transmits and receives data using the air-interface protocols. Test Application Protocol can be used to test the access terminal and the access network.

TAP provides the procedures and messages used by the access terminal and the access network to:

- Control TAP test configurations at both the access terminal and the access network.
- Generate TAP test packets for transmission on the FTC or RTC and process the received packets.
- Generate and transmit information about the received TAP packets through TAP Loop Back packets.
- Transmit configured ACK Channel bits.
- Collect statistics on the changes in the serving sector as seen at the access terminal in the Idle State and the Connected State.
- Collect statistics on the number of successfully received Control Channel packets.

TAP is an application layer protocol. The Application Layer Protocol ID for TAP is specified in [1]. It uses Basic Signaling Protocol [5] for transmitting and receiving messages. It uses Radio Link Protocol [4] to transmit and receive data.

TAP provides multiple flows for transmitting and receiving data and Loop Back packets. A TAP flow is identified with a FlowID. Flows with the same FlowID may transmit and receive data from one another using the air-interface protocols. A TAP flow is mapped to a unique air-interface stream [4]. Multiple TAP flows shall not be mapped to the same stream [4][4]. TAP shall transmit packets belonging to a TAP flow using only a single stream. TAP flow shall not pass data to a stream for which the last 8 LSBs of the ProtocolTypeInUse field of the ProtocolParametersFwd attribute are not set to the application ProtocolID [1] assigned to TAP.

[4] There is only a single InUse instance of TAP. If n TAP flows are active, then the last 8 LSBs of the ProtocolTypeInUse field of the ProtocolParametersFwd and ProtocolParametersRev attribute for n streams [4] should set to the application ProtocolID assigned to TAP.

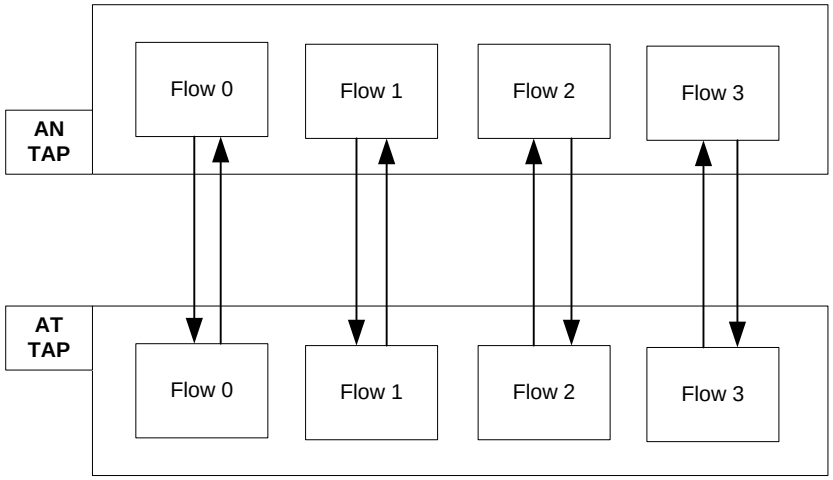

Figure 2.1-1 TAP Flows

The access terminal binds the packets from a TAP flow to an air-interface stream using the StreamIDRev parameter of the FlowN attribute.

2.2 Primitives, Local Common Data, and Public Data

2.2.1 Commands

This protocol does not define any commands.

2.2.2 Indications

This protocol does not return any indications.

2.2.3 Local Common Data

This protocol does not define any Local Common Data.

2.2.4 Public Data

This protocol shall make the following data public:

- All data defined as Static Attribute, Static Non-Attribute Data, and Local Common Data.

2.3 Protocol Data Unit

The protocol data unit for this protocol is a TAP Packet. A TAP Packet consists of a TAP data and a TAP Header.

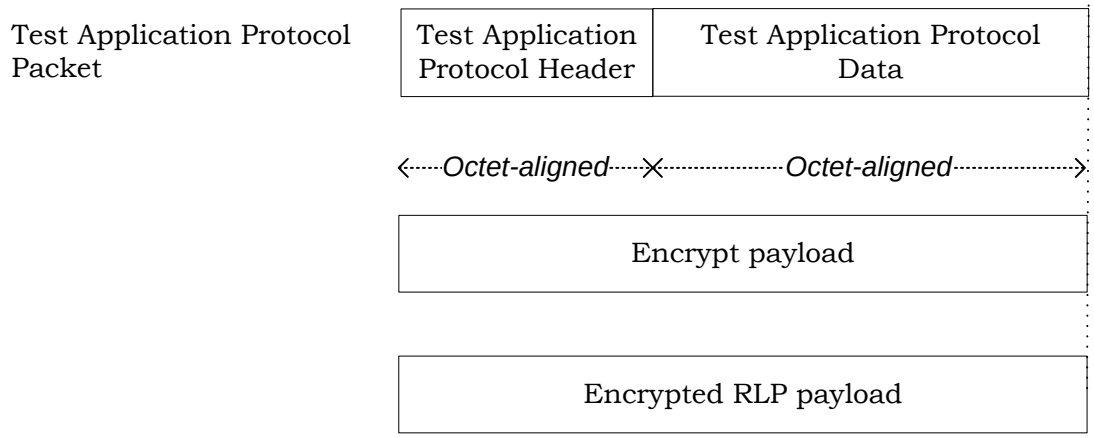

Figure 2.3-1 TAP Packet Encapsulation

In the transmit direction, the TAP generates test packets and forwards them to the Radio Link Layer.

In the receive direction the TAP receives test packets from the Radio Link Layer and processes them.

Figure 2.3-1 illustrates the relationship between the TAP packets and the Radio Link Layer payload. The size of the TAP data is variable and is determined by the PacketType feild in the header. The use of encryption is not mandatory for TAP packets.

TAP uses signaling messages for controlling and configuring the access terminal for conducting tests. When TAP sends these messages it shall use the Basic Signaling Protocol [5].

2.4 Procedures and Messages for the InConfiguration Instance of the Protocol

2.4.1 Procedures

This protocol uses the services of the Session Control Protocol [8] to perform negotiation of attribute values.

The access terminal shall not initiate the negotiation of any of the TAP attributes except MaxFlowsSupported and ATSimulator attributes.

The access network shall not initiate the negotiation of MaxFlowsSupported or ATSimulator attributes.

2.4.1.1 Protocol Initialization

Upon creation, the InConfiguration instance of this protocol in the access terminal and the access network shall perform the procedures specified in [1].

2.4.2 Message Formats

This protocol does not define any messages.

2.5 Test Statistics

2.5.1 Access Terminal Requirements

The access terminal may maintain the following statistics for each TAP flow for which Loop Back mode is enabled:

- TotalPktsTx: This counts the number of TAP Test Packets for this flow that were sent by the access terminal on the RTC.
- SuccessfulPacketTx: This counts the number of TAP Test Packets for this flow that were received by the access network on the RTC as indicated by the Loop Back packets.
- LBPktsLost: This counts the number of TAP Loop Back Packets for this flow that were lost during transmission on the RTC.

2.5.2 Access Network Requirements

The access network shall maintain the following statistics for each TAP flow for which Loop Back mode is enabled:

- TotalPktsTx: This counts the number of TAP Test Packets sent by the access network on the FTC.
- SuccessfulPacketTx: This counts the number of TAP Test Packets that were received by the access terminal on the FTC as indicated by the Loop Back packets.
- LBPktsLost: This counts the number of TAP Loop Back Packets for this flow that were lost during transmission on the RTC.

2.6 Procedures and Messages for the InUse Instance of the Protocol

2.6.1 Procedures

2.6.1.1 Protocol Initialization for the InUse Protocol Instance

Upon creation, the InUse instance of this protocol in the access terminal and access network shall perform the procedures specified in [1].

When the protocol is instantiated, the access terminal shall execute the procedures specified in 2.6.1.5.1.2.

When the protocol is instantiated, the access network shall execute the procedures specified in 2.6.1.5.1.3.

2.6.1.2 Hard Commit Procedures

The access terminal and the access network shall perform the procedures specified in [1] when directed by the InUse instance of the Session Control Protocol to execute the Hard Commit procedures.

2.6.1.3 Soft Commit Procedures

The access terminal and the access network shall perform the procedures specified in [1] when directed by the InUse instance of the Session Control Protocol to execute the Soft Commit procedures.

2.6.1.4 Command Processing

This protocol does not define any commands.

2.6.1.5 Common Procedures

TAP procedures

- Test Parameter Configuration: Procedures and messages for configuring parameters for different tests.
- TAP Packet Transmission and Reception: Procedures for sending and receiving Test data packets or Loop Back Packets on the Forward or Reverse Traffic Channel.
- CQI Channel Transmission: Procedures for transmitting fixed CQI values over the R-CDCCH [3].
- Total Transmit Power: Procedures for fixing the total transmit power of the access terminal.

2.6.1.5.1 Test Parameter Configuration

At the access terminal, parameters for each flow can be assigned a value through TAPParameterAssignment message. These parameters are initialized to various FlowN attributes values upon session configuration.

TestID is updated at the access terminal when a message or a packet with TestID value greater than the current value is received by the access terminal.

An active TAP flow generates TAP packets. A TAP flow at the access terminal is considered active if SourceActive is set to true. Othersiwe, the TAP flow is considered inactive and does not generate any data.

2.6.1.5.1.1 Variable and Statistics Initiatization

For all TAP flows perform the following:

- $V(S_{Test})$ shall be set to zero.
- $V(R_{Test})$ shall be set to zero.
- $V(R_{LB})$ shall be set to zero.
- LoopBackPeriod shall be set to zero.

The access terminal may initialize the following statics for all flows to zero

- TotalPktsTx
- SuccessfulPacketTx

- TAPLBPktsLost

2.6.1.5.1.2 Access Terminal Requirements

When TAP is instantiated access terminal shall

- Disable the Fixed CQI Mode.
- Disable the Fixed Tx Power mode.
- Perform initialization procedures specified in section 2.6.1.5.1.1 and 2.6.1.5.1.2.1.
- Initialize TestID to zero.

If the access terminal receives a TAPParameterAssignment message, it shall execute the following steps in sequence:

- If the TestID contained in the message is greater than the TestID of the TAP at the access terminal, then TAP shall update the TestID to the TestID carried in the message.
- Perform initialization procedures specified in section 2.6.1.5.1.1 and 2.6.1.5.1.2.1.
- If the message contains a TAPDataSoureConfiguration record for any flow, the access terminal shall process them as follows:
    - SourceActive for the flow shall be set to true if SourceActive parameter carried in the message is set to 0x01. Otherwise the SourceActive for the flow shall be set to false.
    - If the flow is active and the PacketType field of the FlowN attribute is set to 0x02, the access terminal shall process the remaining fields as following:
        - NumPackets for the flow shall be to NumPackets parameter carried in the message.
        - PacketSize for the flow shall be PacketSize parameter carried in the message.
        - Period for the flow shall be set to Period parameter carried in the message.
    - If the message contains a CQIValueFixedMode record and the ATSimulator attribute has been negotiated to a value of 0x01, then the access terminal shall enable the Fixed CQI mode and store the received CQIValue feild.
    - If the message contains a TxPowerValueFixedMode record and the ATSimulator attribute has been negotiated to a value of 0x01, then the access terminal shall enable the Fixed Tx Power mode and store the received TxPower feild.
- If all TAP flows are inactive, transmit a TAPParameterComplete message to the access network.

2.6.1.5.1.2.1 Access Terminal Configuration Initiatization

If the DataSourceActive field of the Flow*N* attribute set to 0x01, access terminal shall initialize the following parameter

- SourceActive shall be set to true.

Otherwise, the access terminal shall initialize the the following parameter

- SourceActive shall be set to false.

For all active flows, if the PacketType field of the FlowN attribute is set to 0x02, the access terminal shall initialize the following parameters:

- NumPackets shall be set to BurstNumberPacket.
- PacketSize shall be set to BurstPacketSize.
- Period shall be set to BurstPeriod.

2.6.1.5.1.3 Access Network Requirements

When TAP is instantiated access network shall

- Perform initialization procedures specified in section 2.6.1.5.1.1.
- Initialize TestID to zero.

To change the test configuration the access network shall perform the following steps:

- Increment the TestID.
- Send a TAPParameterAssignment message to the access terminal and wait for a TAPParameterComplete message containing the same TransactionID.
- When the expected TAPParameterComplete message or a packet with current TestID value is received from the access terminal, execute the initialization procedure specified in 2.6.1.5.1.1 and 2.6.1.5.1.3.1.

Access network shall not include CQIValueFixedMode parameter record unless access terminal has negotiated a value of 0x01 for the ATSimulator attribute.

2.6.1.5.1.3.1 Access Network Statistics Initiatization

The access network shall initialize the following statics to zero

- TotalPktsTx
- SuccessfulPacketTx
- LBPktsLost

2.6.1.5.2 TAP Packet Transmission and Reception

TAP flow can generate data and Loop Back packets. The generation of packet at the access terminal is controlled by the PacketType field of the FlowN attribute. Table 2.6-1 provides the encoding of the PacketType field for different packet types.

PacketType	Description
0x00	Loop Back Packets
0x01	Data Packet (Full Buffer)
0x02	Data Packet (Non-Full Buffer)

Table 2.6-1 PacketType Feild Encoding

2.6.1.5.2.1 Packet Transmission and Reception Procedures

Access terminal and access network shall maintain a 24-bit variable $V(S_{Test})$ for each TAP flow. $V(S_{Test})$ shall be included in every packet that it generated by the TAP flow. After generating the packet $V(S_{Test})$ shall be incremented by 1.

Access terminal and access network shall maintain two 24-bit variables $V(R_{Test})$ and $V(R_{LB})$ for each TAP flow.

Access terminal and access network shall maintain an 8-bit variable TestID. TAP shall include TestID in every packet that it transmits.

When the TAP at the access network increments the TestID, it shall perform the initialization procedures specified in section 2.6.1.5.1.1 and 2.6.1.5.1.3.1.

When the TAP at the access terminal receives a packet with TestID greater than the current TestID value at the access terminal, it shall update the TestID value to the value specified in the TAP packet and perform the initialization procedures specified in section 2.6.1.5.1.1 and 2.6.1.5.1.2.1.

If the access terminal or the access network receives a TAP packet with TestID value less than the current TestID value at the receiver, it shall ignore the packet.

All operations and comparisons performed on $V(S_{Test})$, $V(R_{Test})$, $V(R_{LB})$ and TestID shall be carried out in unsigned modulo 2^S arithmetic, where S is the number of bits used to represent the sequence number. For a packet sequence number x, the numbers in the range $[x+1, x + 2^{S-1} - 1]$ are considered greater than x and numbers in the range $[x - 2^{S-1}, x-1]$ are considered smaller than x.

2.6.1.5.2.1.1 Data Packet Transmission

A TAP flow carrying data packets can operate in two different modes: Full Buffer Mode and Non Full Buffer Mode. Full Buffer Mode is used when the test requires that the TAP flow ensure that the the air-interface has data to transmit for this flow at all times during the test. Full Buffer TAP data packets should not be dropped before transmission[5] as they may be used for calculating the FER of the physical layer. Non-Full Buffer Mode is used when data packets need to be produced periodically. Non-Full Buffer TAP Data packets may be dropped before transmission due to buffer limitations.

[5] Such a loss may occur due to buffer overflow in the air-interface

2.6.1.5.2.1.1.1 Common Procedure for Data Packet Transmission

TAP shall include a 32-bit long PseudoRandom PktValidationSeq sequence in every Full Buffer Data packet that it generates. TAP shall include variable length pseudorandom data for Non-Full Buffer Data packets that it generates. When generating PseudoRandom bits, TAP should include fill bits by extracting them from a circular buffer that stores the bits corresponding to one period of any Maximal Length (ML) Sequence of degree 15 or higher.

If the LBNumPackets field of the FlowN attribute is negotiated to a value other than 0x0000, TAP transmitter shall store the following fields of the full buffer data packet:

- $V(S_{Test})$
- PktValidationSeq

Before generating a Full Buffer data packet, TAP shall ensure that RLP has buffers available to receive the data packet. When generating Full Buffer data packet for a flow, TAP shall generate data at a rate such that lower layers have data available for transmission at all times.

When any of the TAP flows is generating Full Buffer data packets for transmission on the forward link, the access network shall transmit only one Route Protocol packet in every Consolidated Packet [3] and shall negotiate a value of 0x01 for PsuedoRandomPaddingEnabledFL attribute.

When the acess network negotiates a value of 0x01 for PacketType field of the FlowN attribute, it shall negotiate a value of 0x01 for SinglePacketEncapsulationEnabled and PsuedoRandomPaddingEnabledRL attribute.

If the transmitter is maintaining the TotalPktsTx for the flow, it shall increment the TotalPktsTx for the flow by 1 after generating a packet for the the flow.

2.6.1.5.2.1.1.2 Access Terminal Procedure for Data Packet Transmission

TAP flow at the access terminal shall not generate any packet if SourceActive for the flow is false.

If SourceActive for FlowN is true and PacketType field is set to 0x01, then the TAP flow N at the access terminal shall generate Full Buffer data packets.

If SourceActive of FlowN is true and PacketType field is set to 0x02, then the TAP flow shall generate Non-Full Buffer data packets. The TAP flow shall generate NumPackets number of PacketSize byte sized packets at the start of Period ms interval.

TAP flow shall pass the packet to the stream identified by the StreamIDRev field of FlowN attribute.

2.6.1.5.2.1.1.3 Access Network Procedure for Data Packet Transmission

If the DataSourceActive field of FlowN attribute is set to 0x01, and if the PacketType field of the FlowN attribute is not set to 0x00 (Loop Back), the TAP Flow N shall not generate packets with PacketType 0x01 or 0x02.

The access network may increment the TestID. The conditions under which the access network increments this variable are beyond the scope of this specification.

2.6.1.5.2.1.2 Data Packet Reception

If the TAP flow is not generating Loop Back packets in response to received TAP data packets then the receiver shall ignore the received packets. Otherwise, TAP will update $V(R_{Test})$ as follows. If the SequenceNumber of the TAP data packet is less than $V(R_{Test})$, the receiver shall ignore the TAP data packet. If the SequenceNumber of the TAP data packet is greater than $V(R_{Test})$, the receiver shall set $V(R_{Test})$ to the SequenceNumber contained in the TAP data packet and store the SequenceNumber and PsuedoRandomFill field. The receiver shall use these fields while transmitting Loop Back packets as specified in section 2.6.1.5.2.1.3.

2.6.1.5.2.1.3 Loop Back Packet Transmission

2.6.1.5.2.1.3.1.1 Common Procedure for Loop Back Transmission

If the LBNumPackets field of the FlowN attribute is set to 0x0000, the receiver of TAP data packets shall not transmit Loop Back packets. Otherwise, it shall transmit the Loop Back packets according to procedure specified below.

If TAP receives a data packet with SequenceNumber equal to SEQ:

- If LoopBackPeriodNum < $\lfloor SEQ / LBNumPackets \rfloor$
    - then the receiver shall generate a Loop Back packet and include the SequenceNumber of all successfully received data packets that satisfy the following equality
        - $\lfloor SequenceNumber / LBNumPackets \rfloor = LoopBackPeriodNum$
    - If no packets satisfy the above condition, then the receiver shall not generate Loop Back Packet
- Set LoopBackPeriodNum to $\lfloor SEQ / LBNumPackets \rfloor$

The receiver shall include the PktValidationSeq field of the data packet received along with the SequenceNumber. TAP flow shall include the SequenceNumber of the received data packets in ascending order.

When transmitting Loop Back TAP Packets, TAP should ensure that the packets are not dropped in lower layers due to buffer overflow.

2.6.1.5.2.1.3.1.2 Access Terminal Procedure for Loop Back Transmission

If SourceActive of FlowN is true and the PacketType field of the FlowN attribute is set to 0x00, the access terminal shall generate Loop Back packets. The access terminal shall follow the procedure specified in 2.6.1.5.2.1.3.1.1 when transmitting Loop Back packets.

2.6.1.5.2.1.3.1.3 Access Network Procedure for Loop Back Transmission

If the DataSourceActive field of FlowN attribute is set to 0x01 and the PacketType field of the FlowN attribute is set to 0x00, the access network shall not negotiate a value of 0x0000 LBNumPackets field of the FlowN attribute.

If the DataSourceActive field of FlowN attribute is set to 0x01, the PacketType field of the FlowN attribute is not set to 0x00 (Loop Back), and the LBNumPackets field of the FlowN attribute is set to a value greater than 0x0000, the access network shall generate Loop Back packets. The access network shall follow the procedure specified in 2.6.1.5.2.1.3.1.1 when transmitting Loop Back packets.

The access network shall configure the RLP attributes for the stream carrying the Loop Back packets to deliver the packets in-order to TAP.

2.6.1.5.2.1.4 Loop Back Packet Reception

The receiver of Loop Back packets shall update LBPktsLost and SuccessfulPacketTx statistics. It shall use the following information for updating these statistics:

- LBSeqNum, the LBSeqNum of the Loop Back packet.
- SEQ, the first SequenceNumber included in the Loop Back packet
- N = $\lfloor$SEQ/ LBNumPackets $\rfloor$

The receiver shall update LBPktsLost for the flow as follows:

- If LBSeq <= $V(R_{LB})$, the receiver shall consider the current test to have failed[6].
- Otherwise, receiver shall add (LBSeq - $V(R_{LB})$ – 1) to LBPktsLost
- Update $V(R_{LB})$ = LBSeq.

The receiver shall update SuccessfulPacketTx as follows:

- update $V(R_{Test})$ = N * LBNumPackets – 1
- for each SEQ and PktValidationSeq pair in the Loop Back Packet with SEQ > $V(R_{Test})$, compare the PktValidationSeq field with the stored PktValidationSeq field of the data packet transmitted with the corresponding SequenceNumber. If the PktValidationSeq field matches that of the transmitted packet, the receiver shall consider the data packet transmission with corresponding SequenceNumber to be successful and increment SuccessfulPacketTx for the flow by 1.
- Update $V(R_{Test})$ = ((N + 1) * LBNumPackets) – 1.

2.6.1.5.2.2 Packet Headers

2.6.1.5.2.2.1 Loop Back Packet Format

This packet is sent to convey information about the received TAP Test Packets.

---

[6] RLP for the stream transmitting Loop Backet packets should be configured for in-order delivery.

Field	Length (bits)
TestID	8
FlowID	3
PacketType	5
LBSeqNum	24
RecordCount	8

Up to RecordCount occurrences of the following record:

SEQ	24
PktValidationSeq	32

TestID — This sender shall set this field to the current TestID value.

FlowID — This sender shall set this field to the FlowID of the flow generating this packet.

PacketType — The sender shall set this field to '00000'.

LBSeqNum — The sender shall set this field to the $V(S_{Test})$ for the flow.

RecordCount — The Number of TAP Data packets acknowledged in this packet. The valid range for this field is 1 to LBNumPackets.

SEQ — The SequenceNumber of the TAP data packet being acknowledged.

PktValidationSeq — The PsuedorandomFill carried in the TAP data packet with SequenceNumber equal to SEQ field of the current record.

2.6.1.5.2.2.2 Data Packet Format

This packet is transmitted when the TAP flow is a full buffer source.

Field	Length (bits)
TestID	8
FlowID	3
PacketType	5
SequenceNumber	24
PktValidationSeq	32
PsuedorandomFill	Variable

TestID — This sender shall set this field to the current TestID value.

FlowID — This sender shall set this field to the FlowID of the flow generating this packet.

PacketType — The sender shall set this field to '00001' when transmitting a Full Buffer data packet. The sender shall set this field to '00002' when transmitting a Non Full Buffer data packet.

SequenceNumber — The sender shall set this field to the $V(S_{Test})$ for the flow.

PktValidationSeq — The sender shall include fill bits that are extracted from a circular buffer that stores bits corresponding to one period of any Maximal Length (ML) Sequence of degree 15 or higher. The buffer may be shared across TAP flows.

PsuedorandomFill — The sender shall not include this field if the PacketType is not set to '00002'. The sender shall include fill bits that are extracted from a circular buffer that stores bits corresponding to one period of any Maximal Length (ML) Sequence of degree 15 or higher. The buffer may be shared across TAP flows. The length for this field shall be 8k bits, where k is in the range 0 to 1500.

2.6.1.5.3 CQI Transmission

2.6.1.5.3.1 Access terminal requirements

If the Fixed CQI mode is enabled, then the access terminal shall set its transmitted CQI to the value specified in the CQIValueFixedMode parameter record.

2.6.1.5.4 Fixed TxPower Mode

2.6.1.5.4.1 Access terminal requirements

If the Fixed TxPower mode is enabled, then the access terminal shall set

P_{PICH} = -50 + 0.25 * TxPower – 10*log10(BW_{PICH}) dBm/Hz

where TxPower is specified in the TxPowerFixedMode parameter record and BW_{PICH} is the bandwidth of R-PICH (1.25MHz).

2.6.1.5.5 Lower Layer Protocol Configuration

If NumHARQFL is negotiated to a value in the range 0x01-0x06, then access network shall

- transmit exactly NumHARQFL number of HARQ transmission of the packet
- ignore the ack/nack for the forward link packet transmissions.
- For a persistent assignment, begin transmission of the next packet after NumHARQFL number of HARQ attempts of the previous packet.

If NumHARQFL is negotiated to a value in the range 0x01-0x06, then access terminal shall

- Expect to receive exactly NumHARQFL number of HARQ transmissions
- For a persistent assignment, begin decoding the next packet after NumHARQFL number of HARQ attempts of the previous packet, regardless of the previous packet being received in error.

If NumHARQRL is negotiated to a value in the range 0x01-0x06, then access terminal shall

- transmit exactly NumHARQRL number of HARQ transmission of the packet
- ignore the ack/nack for the reverse link packet transmissions.
- For a persistent assignment, begin transmission of the next packet after NumHARQRL number of HARQ attempts of the previous packet.

If NumHARQRL is negotiated to a value in the range 0x01-0x06, then access network shall

- Expect to receive exactly NumHARQRL number of HARQ transmissions
- For a persistent assignment, begin decoding the next packet after NumHARQRL number of HARQ attempts of the previous packet, regardless of the previous packet being received in error.

If SinglePacketEncapsulationEnabled is negotiated to a value of 0x01, then the access terminal shall only carry zero or one route packets in the consolidated packets [3].

If PsuedoRandomPaddingEnabledFL is negotiated to a value of 0x01, then the access network shall set last padding octet (the octet closest to the end of the MAC packet [3]) of the consolidated packet to 0x00, and set the remaining padding octets to randomly generated values.[7]

If PsuedoRandomPaddingEnabledRL is negotiated to a value of 0x01, then the access terminal shall set last padding octet (the octet closest to the end of the MAC packet [3]) of the consolidated packet to 0x00, and set the remaining padding octets to randomly generated values[7]

[7] The sender can extract the padding bits from a pre-generated pseudo-random circular buffer, with different offsets for different packets.

1 2.6.2 Type Definitions

2 Protocol Type values assigned to protocols are specified in [1] Protocol Type value for InUse
3 and InConfiguration instance of this protocol is negotiated by Test Application Protocol.

4 **2.7 Message Formats**

5 2.7.1 TAPParameterAssignment

6 The access network sends this message to configure the TAP parameters.

7

Field	Length (bits)
MessageID	8
TransactionID	8
TestID	8

Zero or more occurrences of the following record:

ParameterRecord	Parameter Record Dependent

8 MessageID The access network shall set this field to 0x00.

9 TransactionID The access network shall set this field to 1 higher than the
10 TransactionID field of the last TAPParameterAssignment message
11 (mod 256) sent to this access terminal.

12 TestID The access network shall set this to the current value of TestID.

13 ParameterRecord The permissible parameter records are
14 TAPDataSourceConfiguration, CQIValueFixedMode and
15 TxPowerFixedMode as specified in 2.7.1.1-2.7.1.3.

16

Channels	FTC	**RLP**	Reliable
Addressing	unicast	**AuthTag**	Required when key is available

17 2.7.1.1 TAPDataSourceConfiguration ParameterRecord

18 This parameter record is included if the access network requires the access terminal to
19 configure the TAP data sources.

20

Field	Length (bits)	Default
Length	8	N/A
ParameterRecordID	8	N/A
NumDataSource	8	N/A

Up to NumDataSource occurrences of the following record:

FlowID	8	N/A
SourceActive	8	N/A
Period	0 or 16	0x0000
NumPackets	0 or 16	0x0000
PacketSize	0 or 16	0x0000

Length — Length of the parameter record in octets. The access network shall set this field to length of the parameter record excluding the Length field.

ParameterRecordID — The access network shall set this field to 0x00.

NumDataSource — This field is contains the number of DataSourceRecords included in this message. The valid values of this field are in the range 0x00 to 0x07.

FlowID — FlowID of the TAP flow for which the configuration parameters are included in this record.

SourceActive — The sender may set this field to 0x00 or 0x01. If the value is set to 0x00, then the access terminal shall not generate any packets for this TAP flow.

Period — The sender shall not include this field if the SourceActive field is set to 0x00. The sender shall set this to the time period in units of ms at which the data source shall generate data.

NumPackets — The sender shall not include this field if the SourceActive field is set to 0x00. The Sender shall not include this field if the value of the Period field is set to 0x0000. The sender shall set this field to the number of packets to be generated by the data soure.

PacketSize — The sender shall not include this field if the SourceActive field is set to 0x00. The Sender shall not include this field if the value of the Period field is set to 0x0000. The sender shall

set this to the size of the packets to be generated by the data source excluding the TAP Packet header.

2.7.1.2 CQIValueFixedMode ParameterRecord

This parameter record is included if the access network requires the access terminal to fix the value of CQI being transmitted. The access network shall not include this record if ATSimulator attribute is set to 0x00.

Field	Length (bits)	Default
Length	8	N/A
ParameterRecordID	8	N/A
CQIValue	8	N/A

Length — Length of the parameter record in octets. The access network shall set this field to length of the parameter record excluding the Length field.

ParameterRecordID — The access network shall set this field to 0x01.

CQIValue — This field is coded as per CQI value specification in [3]. All values in the range '00000' to '11111' are permissible.

2.7.1.3 TxPowerFixedMode ParameterRecord

This parameter record is included if the access network requires the access terminal to fix transmit power. The access network shall not include this record if ATSimulator attribute is set to 0x00.

Field	Length (bits)	Default
Length	8	N/A
ParameterRecordID	8	N/A
TxPower	16	N/A

Length — Length of the parameter record in octets. The access network shall set this field to length of the parameter record excluding the Length field.

ParameterRecordID — The access network shall set this field to 0x02.

TxPower — This field specifies the P_{PICH} [3] to be used by the access terminal. The valid values for this field are in the range of 0-296.

2.7.2 TAPParameterComplete

The access terminal sends this message in response to TAPParameterAssignment message, if all TAP flows at the access terminal become inactive as a result of TAPParameterAssignment message.

Field	Length (bits)
MessageID	8
TransactionID	8

MessageID — The access network shall set this field to 0x01.

TransactionID — The access terminal shall set this field to to the TransactionID field value of the associated TAPParameterAssignment message.

2.8 Interface to other Protocols

2.8.1 Commands Sent

This protocol does not issue any commands.

2.8.2 Indications

This protocol does not register to receive any indications.

2.9 Configuration Attributes

2.9.1 Simple Attributes

The negotiable simple attributes for this protocol is listed in Table 2-2. The access terminal and the access network shall use as defaults the values in Table 2-2 typed in ***bold italics***.

Table 2-2. Simple Configurable Attributes

Attribute ID	Attribute	Commit / Scope	Values	Meaning
0x0000	MaxFlowsSupported	Hard / Static	**0x02**	Access terminal supports a maximum of 2 TAP flows.
			0x03-0x08	Maximum number of TAP flows supported by the access terminal.
			All other values	Reserved
0x0001	ATSimulator	Hard / Static	**0x00**	Access terminal does not support simulator capabilities
			0x01	Access terminal supports simulator capabilities.
			All other values	Reserved.
0x0002	NumHARQFL	Hard / Static	**0x00**	Access terminal and access network follow normal h-arq operation for the forward link transmissions.
			0x01-0x06	Access terminal expects exactly NumHARQFL h-arq transmissions for each forward link packet.
			All other values.	Reserved.

Attribute ID	Attribute	Commit / Scope	Values	Meaning
0x0003	NumHARQRL	Hard / Static	**0x00**	Access terminal follows normal h-arq operation for the reverse link transmissions.
			0x01-0x06	Access terminal uses exactly NumHARQRL h-arq transmissions for each packet transmitted on the reverse link.
			All other values.	Reserved.
0x0004	SinglePacketEncapsulationEnabled	Soft Allowed / Static	**0x00**	A consolidated packet transmitted by the access terminal may carry greater than one Route packets.
			0x01	A consolidated packet transmitted by the access terminal is not allowed to carry more than one Route packet.
			All other values.	Reserved.

Attribute ID	Attribute	Commit / Scope	Values	Meaning
0x0005	PsuedoRandomPaddingEnabledFL	Soft Allowed / Static	**0x00**	Padding bytes are set to zero for forward link transmissions.
			0x01	Pseudo random padding bytes are used for forward link transmissions.
			All other values.	Reserved.
0x0006	PsuedoRandomPaddingEnabledRL	Soft Allowed / Static	**0x00**	Padding bytes are set to zero for reverse link transmissions.
			0x01	Pseudo random padding bytes are used for reverse link transmissions.
			All other values.	Reserved.

2.9.2 Complex Attributes

The following complex attributes and default values are defined (see [1] for attribute record definition).

2.9.2.1 Flow*N* Attribute

N is a hexadecimal with the 3 LSB's same as FlowID, where hexadecimal digits A through F are specified in upper case letters.

The sender shall set the AttributeID field to 0x810*K*, where *K* is the hexadecimal identifier for FlowID with valid range of 0x00 to 0x07.

Field	Length (bits)	Default
DataSourceActive	8	***0x00***
StreamIDRev	8	***N/A***
PacketType	8	***0x00***
BurstPeriod	16	***0x0000***
BurstNumberPacket	16	***0x0000***
BurstPacketSize	16	***0x0000***
LBNumPackets	8	***0x01***

DataSourceActive — The sender shall set this field to 0x00 or 0x01. The receiver shall ignore the remaining fields if this field is set to 0x00.

StreamIDRev — The sender shall set this field to a value in the range 0x07-0x1f. The sender shall not repeat the value used in this field for any other FlowN attribute.

PacketType — The sender shall set this field to PacketType used for the flow as defined in Table 2.6-1.

BurstPeriod — The receiver shall ignore this field if the PacketType field is not set to 0x02. The sender shall set this to the time period in units of ms at which the data source shall generate data.

BurstPacketSize — The receiver shall ignore this field if the PacketType field is not set to 0x02. The receiver shall ignore this field if the value of the BurstPeriod field is set to 0x0000. The sender shall set this to the size of the packets to be generated by the data source excluding the TAP Packet header.

BurstNumberPacket — The receiver shall ignore this field if the PacketType field is not set to 0x02. The receiver shall ignore this field if the value of the BurstPeriod field is set to 0x0000. The sender shall set this field to the number of packets to be generated by the data soure.

LBNumPackets — If the PacketType field is set to 0x00, the sender shall include this field and assign a value greater than 0x0000. Otherwise, the sender may include this field[8]. For example, if the access terminal is transmitting data packets for this flow on the

[8] This condition will occur when TAP flow at the access network is transmitting the Loop Back packets.

reverse link and the access network is transmitting Loop Back packets for this flow on the forward link, then the sender will include this field for PacketType field is set to 0x01 or 0x02. The sender shall set the field to the number of data packets for which a Loop Back packet will be generated when no packets are lost.

2.10 Non-Attribute Data

2.11 Session State Information

The Session State Information record (see [1]) consists of parameter records. All configuration attributes and Non-attribute data are Session State Information records. This protocol does not define additional parameter records.

1 No Text.

3 TEST APPLICATION EXAMPLE FLOW DIAGRAMS AND STATISTICS COLLECTION

3.1 Overview

This section provides some Test Application Protocol flow diagrams to illustrate the use of TAP procedures for characterizing the Forward and Reverse Traffic Channels. It also provides some sample computations of performance measures. These are for informational purpose only. The tests that can be performed using these procedures and the manner, in which they can be conducted, are by no means limited to these few illustrations.

3.2 Configuration

This section illustrates the various configurations that are required for conducting a test. Session Configuration is used for negiotiating the following personalities

1. Reliable Communication Personality: This personality is used for reliable communication between the AT and the AN. The Reliable Configuration Personality should be used as the InUse instance while negotiating other personalities using SCP.
2. Initial Test Personality (ITP): The ITP serves as a starting point for session negotiation required for individual tests, and may be used to initialize the InConfig instance in order to configure the appropriate test run personality (TRP).
3. Test Run Personality (TRP): The TRP is used as the InUse personality for conducting the test.

TAP messaging is used to start and stop the collection of statistics and activate / de-activate TAP flows.

3.2.1 Reliable Communication Personality Configuration

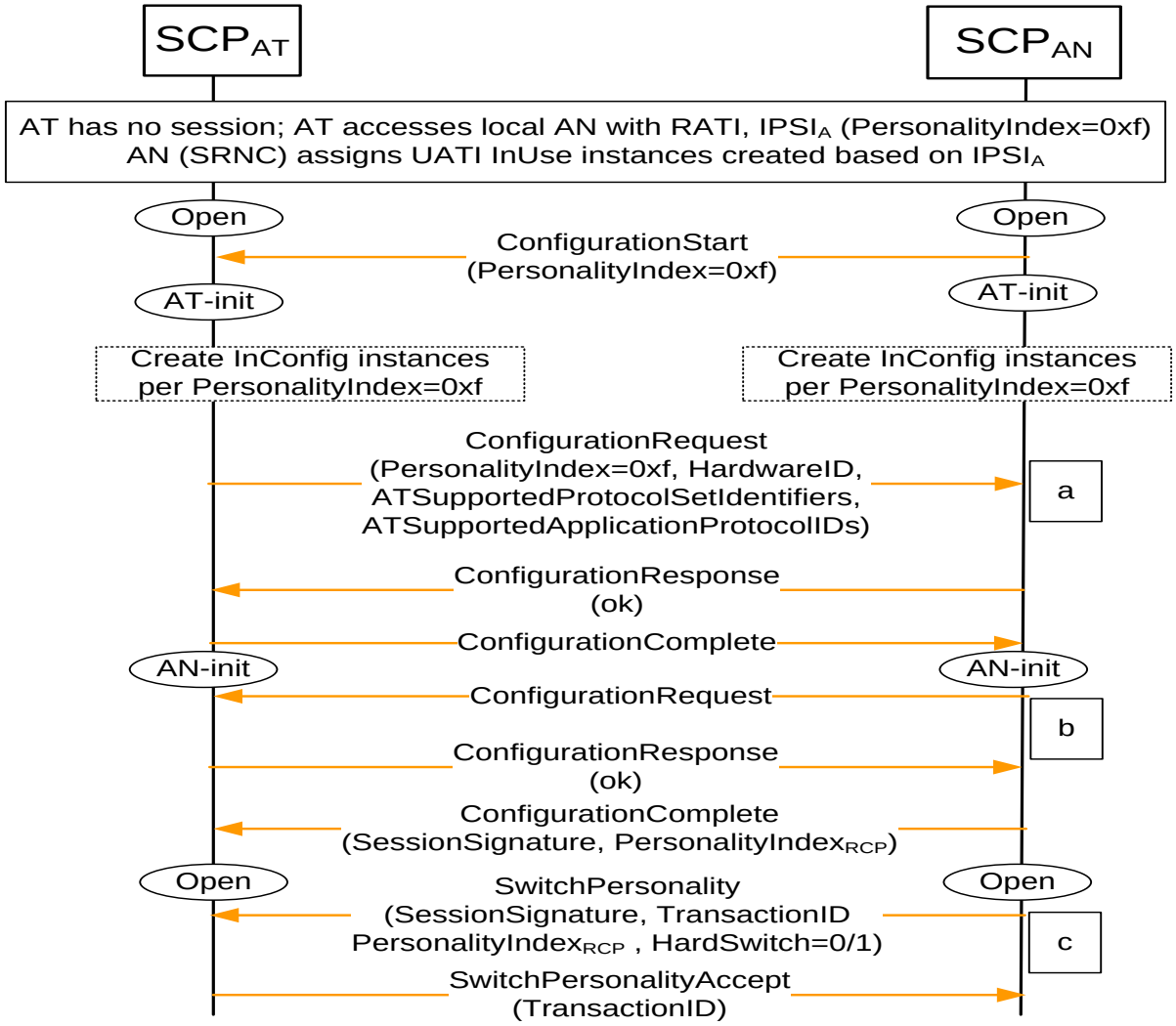

Figure 3.2-1. Flow Diagram for Initial Session Negotiation

Figure 3.2-1 shows the configuration of a test personality.

a. During AT-initiated session configuration, AT negotiates ATSupportedApplicationProtocolIDs. This attribute record application ProtocolID for TAP. AN accepts the values proposed by the AT.

b. During AN-initiated session configuration AN may negotiate attiributes of different air-interface protocols. The subtypes of the protocols are determined by $IPSI_A$.

c. Once the configuration is complete the AN stores the personality in bin $PersonalityIndex_{RCP}$ (Reliable Communication Personality). AN then switches the personality from InConfiguration instance to RCP.

Reliable Communication Personality provides reliable communication between the AT and the AN. Hence, it should not negotiate any attribute to a value that will result in a low

reliability for the signaling or data streams. AN will switch the AT to RCP when it is not conducting a test.

3.2.2 Initial Test Personality Configuration

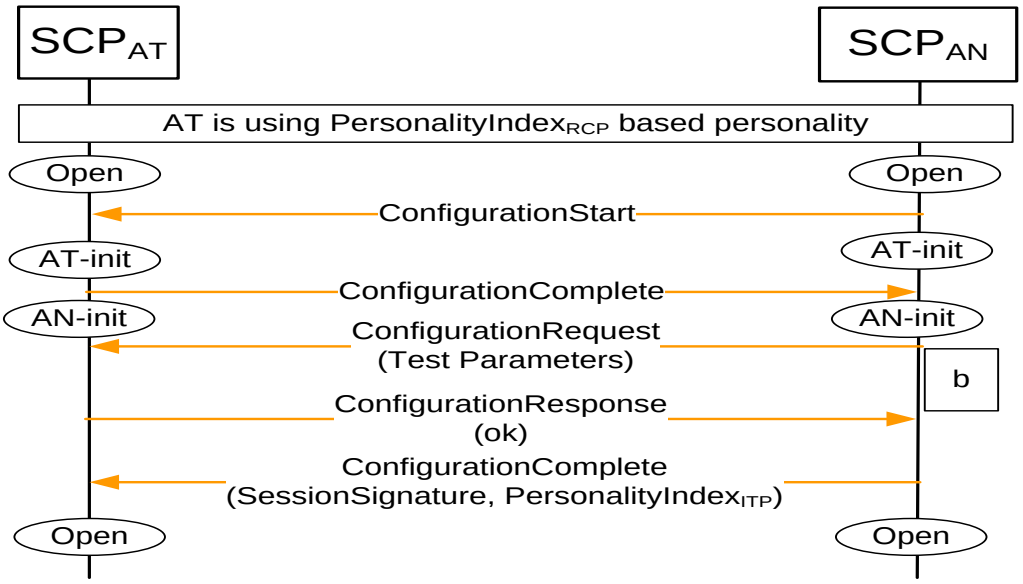

Figure 3.2-2. Flow Diagram for Initial Session Negotiation

Figure 3.2-1 shows the configuration of a test personality.

a. AN and AT are using $PersonalityIndex_{RCP}$ based personality.

b. AN transmits a ConfigurationStart message to start the negotiation of a new personality. AT transmits a ConfigurationComplete message as it does not need to negotiate any attribute.

c. During AN-initiated session configuration AN negotiates attiributes of different air-interface protocols. This includes ProtocolParametersFwd and ProtocolParametersRev for various streams that are needed by TAP. It may also negotiate PCP attributes that ensure appropriate priority between various streams used by TAP. AN also negotiates TAP attributes. This includes StreamIDRev that binds a TAP flow to air-interface stream.

The subtypes and attribute values form an Initial Test Personality. When the AN wishes to conduct a test, it shall include this personality in the ConfigurationStart message, i.e. this personality serves as a starting point for negotiation of attributes for any test.

3.2.3 TAP Messaging

AN may transmit ClearStats and /or ActivateFlow message to the AT. This may be needed prior to the session configuration for the test as the session configuration may result in a

physical layer channel with low reliablity. TAP messages may also be exchanged after the test configuration is complete.

3.2.4 Test Run Personality Configuration

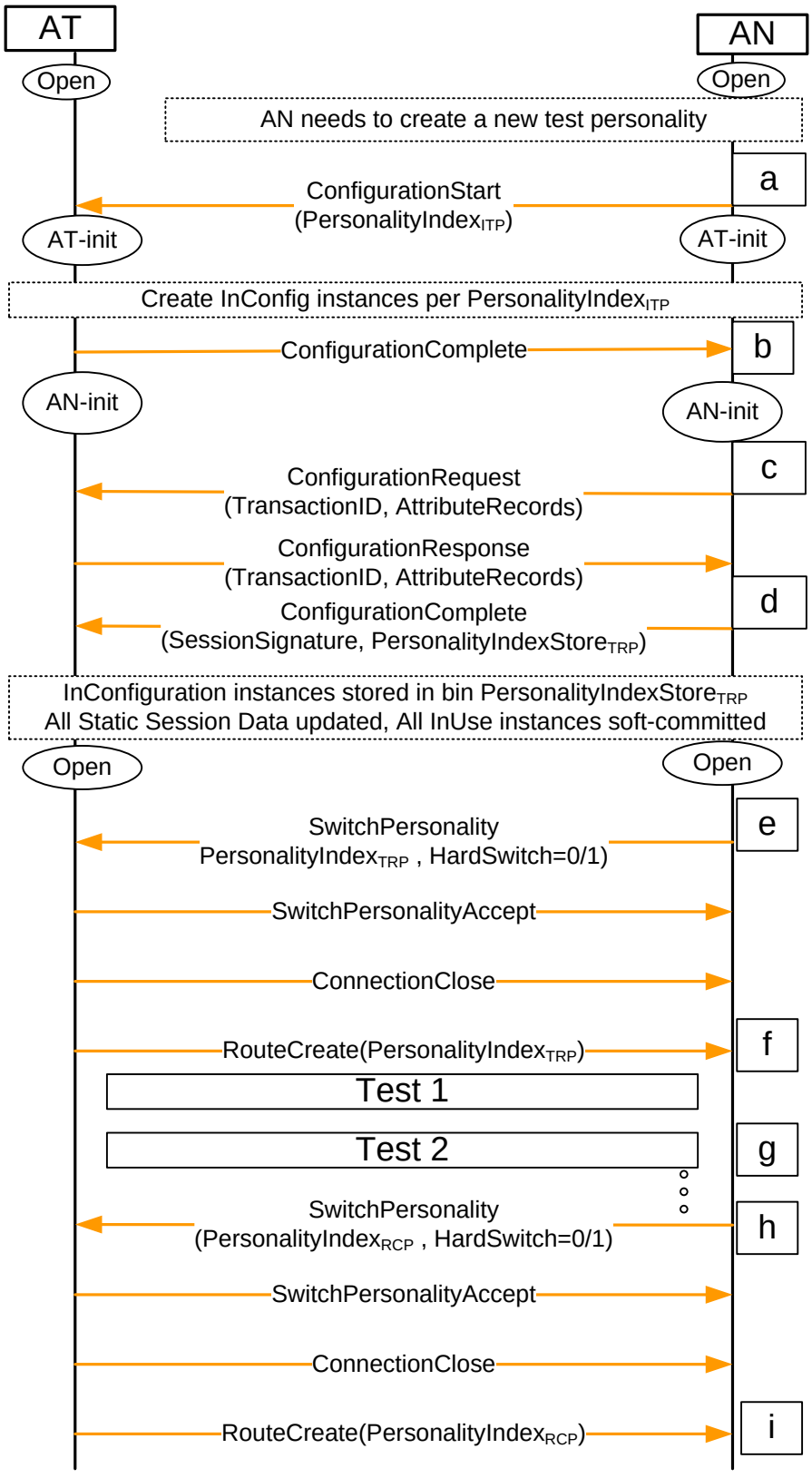

Figure 3.2-3. Session Negotiation for Test Configuration

Figure 3.2-3 shows the session configuration procedure needed for achieving test configuration.

a. AN transmits a ConfigurationStart message to start session configuration w.r.t. Initial Test Personality.

b. AT transmits a ConfigurationComplete message as it does not need to negotiate any attribute.

c. AN transmits a ConfigurationRequest message with the proposed values for various attributes as required for the test.

d. AN transmits a ConfigurationComplete message and stores the personality in $PersonalityIndex_{TRP}$ (Test-Run Personality).

e. AN switches the personality to Test-Run Personality. If any hard-commitable attribute has been negotiated w.r.t. Reliable Communication Personality, AT will close the connection.

f. If the AT needed to close the connection, AT will reopen the connection and include Test-Run Personality in the Route Create header.

g. AN runs test that using the Test-Run Personality.

h. Once the tests are completed, AN will switch the personality back to the Reliable Communication Personality.

i. If the AT needed to close the connection, AT will reopen the connection and include Reliable Test Personality in the Route Create header.

3.3 PER Measurement

When the Loop Back mode is enabled, the PER can be computed using the following statistics:

TotalPktsTx: This counts the number of TAP Test Packets for this flow that were transmitted.

SuccessfulPacketTx: This counts the number of TAP Test Packets for this flow that were received successfully as indicated by the Loop Back packets.

LBPktsLost: This counts the number of TAP Loop Back Packets for this flow that were lost during transmission.

LBNumPackets: The number of data packets for which a Loop Back packet is generated when there is no packet loss.

LBExpected: Total Number of Loop Back Packets that should be received under clean conditions = $\lfloor$ TotalPktsTx / LBNumPackets $\rfloor$

Then:

PER (%)= (1- SuccessfulPacketTx / ((LBExpected- LBPktsLost) * LBNumPackets)) * 100