M3UA Protocol Emulation for TTCN-3 Toolset with TITAN, Function Specification

Contents

[1 Introduction 2](#_Toc169674140)

[1.1 Revision History 2](#_Toc169674141)

[1.2 How to Read this Document 2](#_Toc169674142)

[1.3 Scope 2](#_Toc169674143)

[1.4 References 2](#_Toc169674144)

[1.5 Abbreviations 2](#_Toc169674145)

[1.6 Terminology 3](#_Toc169674146)

[2 Overview 3](#_Toc169674147)

[3 Functional Specification 3](#_Toc169674148)

[3.1 Implementation 3](#_Toc169674149)

[3.1.1 Starting the emulation 3](#_Toc169674150)

[3.1.2 Initialization 5](#_Toc169674151)

[3.1.3 The dynamic part 5](#_Toc169674152)

[3.2 Logging 7](#_Toc169674153)

[3.3 Limitations 7](#_Toc169674154)

# Introduction

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2007-06-20 | PA1 | First draft version | EFERKOV |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## How to Read this Document

This is the Function Specification for the M3UA Protocol Emulation (M3UA PE). M3UA PE is developed for the TTCN-3 Toolset with TITAN. This document should be read together with the Product Revision Information [3].

## Scope

The purpose of this document is to specify the functionality and configurability of the M3UA PE.

## References

1. 2/198 17-CRL 113 200 Uen  
   Programmer’s Technical Reference for the TITAN TTCN-3 Test Executor
2. ETSI ES 201 873-1 V3.2.1  
   The Testing and Test Control Notation version 3. Part 1: Core Language
3. 109 21-CNL 113 537-1 Uen  
   M3UA Protocol Emulation for TTCN-3 Toolset, Product Revision Information
4. 198 17-CNL 113 537 Uen  
   M3UA Protocol Emulation for TTCN-3 Toolset, User Guide
5. IETF RFC 3332  
   Signaling System 7 (SS7) Message Transfer Part 3 (MTP3) – User Adaptation Layer (M3UA)

## Abbreviations

ASP Abstract Service Primitive

M3UA MTP3 User Adaptation

M3UA PE M3UA Protocol Emulation

MTP3 Message Transfer Part Level 3

PE Protocol Emulation

SCTP Stream Control Transmission Protocol

SEP SS7 Signalling End Point

SGP Signalling Gateway Process

SS7 Signalling System 7

SUT System Under Test

TTCN-3 Testing and Test Control Notation Version 3

## Terminology

Protocol Emulation: An instance which implements messages and dynamic behaviour of a given protocol layer.

# Overview

It’s necessary to introduce the M3UA PE (working with an SCTP test port and SCTP implementation). The M3UA PE runs on the Application Server Process (see Figure 1). It uses a single SCTP test port (with a single SCTP association) to communicate with the SUT. On the other side, the MTP3 users (e.g. SCCP, ISUP entities) communicate with the M3UA PE using MTP3 ASPs.

Application Server Process

IP/SCTP

SS7

SGP

SEP

M3UA PE

# Functional Specification

Figure 1: The place of the M3UA PE

## Implementation

### Starting the emulation

The M3UA PE can be started with the f\_M3UA\_Emulation() function from a TTCN-3 test case. It requires a single parameter of type SCTP\_Association\_Address, which describes the single SCTP association between the SUT and the M3UA PE (on the Application Server Process). The structure looks like the following.

type record SCTP\_Association\_Address

{

integer local\_sctp\_port,

charstring local\_ip\_addr,

integer remote\_sctp\_port,

charstring remote\_ip\_addr

}

This information on the SCTP association is stored globally with the actual status of the M3UA PE (stored in the commStatus variable). The following scenario (Figure 2) shows a usual message flow between the SUT and the M3UA PE, which builds up the SCTP association for M3UA communication in client mode. Only client mode is explained, server mode is similar.

SGP/SUT

Application Server Process

SCTP Connect From

SCTP Result

SCTP Communication Up

M3UA ASPUP

M3UA ASPUP Ack

M3UA ASPAC

M3UA ASPAC Ack

M3UA PE

Figure 2: Building the SCTP association and the M3UA connection

The commStatus changes appropriately. At first the M3UA PE is in aSP\_Down\_initial\_State, but it changes to aSP\_Down\_sCTP\_Initialize\_Done immediately. After sending the SCTP Connect From message the state changes to aSP\_Down\_sCTP\_Associate\_done in case the SCTP Result was not an error message. After reception of the SCTP Communication Up message from the SUT (aSP\_Down\_commUP\_Received), the M3UA PE sends a M3UA ASPUP message, waits for an M3UA ASPUP Ack message, and goes to aSP\_Down\_ASPUP\_Sent state. The M3UA PE becomes aSP\_Inactive after receiving the M3UA ASPUP Ack message, and sends a M3UA ASPAC message (aSP\_Inact\_ASPAC\_Sent). Finally, after the reception of M3UA ASPAC Ack the state becomes aSP\_Active. At this point the SCTP association and the M3UA connection is up and running.

### Initialization

The SCTP part must be initialized with the f\_initialize\_SCTP() function. Setting the module parameters (described in [4]) is not part of this initialization process, they’re already set at this point via config file parameters. If the M3UA PE runs in server mode an ASP\_SCTP\_Listen message will be sent out to the SUT, otherwise the association will be built (an ASP\_SCTP\_ConnectFrom message will be sent) and a timer (T\_Assoc\_restart) will be started to detect, if the M3UA PE hasn’t received the appropriate ASP\_SCTP\_RESULT message in response to the ASP\_SCTP\_ConnectFrom message. If this timer expires the M3UA PE will try to reassociate.

For transferring M3UA messages as the payload of SCTP packets, the payload type must be adjusted to 3 (= M3UA). The tsp\_SCTP\_PayloadProtocolID module parameter represents this value. If this payload protocol identifier is set to 3 (= M3UA), some M3UA specific timers are started.

* T\_ASPUP\_resend is started in client mode only. If we don’t receive a PDU\_M3UA\_ASPUP\_Ack message in response to our PDU\_M3UA\_ASPUP message we need to resend it.
* T\_ASPAC\_resend is started in client mode only. If we don’t receive a PDU\_M3UA\_ASPAC\_Ack message in response to our PDU\_M3UA\_ASPAC message we need to resend it.
* T\_Heartbeat is started only if the tsp\_Enable\_M3UA\_Heartbeat module parameter is set. If this timer expires, a PDU\_M3UA\_Heartbeat message is sent through the SCTP association to the M3UA peer at the SGP. This message will inform the M3UA peer, that we’re still alive at the M3UA level (SCTP has its own heartbeat mechanism implemented in the kernel).

The M3UA PE transfers non-M3UA protocols (e.g. if the tsp\_SCTP\_PayloadProtocolID is 0 (= unspecified)) transparently via SCTP.

### The dynamic part

The dynamic part is implemented in the f\_M3UA\_ScanEvents() function. It’s a simple infinite loop (an alt statement with repeat statements for each alternative), which sends and receives messages.

|  |  |  |  |
| --- | --- | --- | --- |
| **Message type** | **Message name** | **Client mode** | **Server mode** |
| Transfer | M3UA\_DATA | Can be sent or received. | Can be sent or received. |
| SSNM | M3UA\_DUNA | Not implemented. | Not implemented. |
| M3UA\_DAVA | Sent in response of a M3UA\_DAUD message. | Sent in response of a M3UA\_DAUD message. |
| M3UA\_DAUD | Can be sent or received. A M3UA\_DAVA is sent back in response. | Can be sent or received. A M3UA\_DAVA is sent back in response. |
| M3UA\_SCON | Not implemented. | Not implemented. |
| M3UA\_DUPU | Not implemented. | Not implemented. |
| M3UA\_DRST | Not implemented. | Not implemented. |
| ASPSM | M3UA\_ASPUP | Sent by the client. | Received from the client. A M3UA\_ASPUP\_Ack is sent to the client. |
| M3UA\_ASPUP\_Ack | Received from the server in response of a M3UA\_ASPUP message. | Sent to the client in response of a M3UA\_ASPUP message. |
| M3UA\_ASPDN | Can be sent or received. A M3UA\_ASPDN\_Ack is sent back in response. | Can be sent or received. A M3UA\_ASPDN\_Ack is sent back in response. |
| M3UA\_ASPDN\_Ack | Not implemented. | Not implemented. |
| M3UA\_BEAT | Can be sent or received. An M3UA\_BEAT\_Ack is sent back in response. | Can be sent or received. An M3UA\_BEAT\_Ack is sent back in response. |
| M3UA\_BEAT\_Ack | Sent in response of a M3UA\_BEAT message. | Sent in response of a M3UA\_BEAT message. |
| RKM | M3UA\_REG\_REQ | Not implemented. | Not implemented. |
| M3UA\_REG\_RSP | Not implemented. | Not implemented. |
| M3UA\_DEREG\_REQ | Not implemented. | Not implemented. |
| M3UA\_DEREG\_RSP | Not implemented. | Not implemented. |
| ASPTM | M3UA\_ASPAC | Sent by the client. | Received from the client. A M3UA\_ASPAC\_Ack is sent to the client. |
| M3UA\_ASPAC\_Ack | Received from the server in response of a M3UA\_ASPAC message. | Sent to the client in response of a M3UA\_ASPAC message. |
| M3UA\_ASPIA | Can be sent or received. A M3UA\_ASPIA\_Ack is sent back in response. | Can be sent or received. A M3UA\_ASPIA\_Ack is sent back in response. |
| M3UA\_ASPIA\_Ack | Not implemented. | Not implemented. |
| MGMT | M3UA\_ERR | Not implemented. | Not implemented. |
| M3UA\_NOTIFY | Not implemented. | Not implemented. |

Table 1: M3UA message handling

The M3UA PE connected to the MTP3 user via the MTP3\_SP\_PORT port. From the MTP3 user only ASP\_MTP3\_TRANSFERreq messages can be received. If the SCTP association and the M3UA connection between the SUT and the M3UA PE is not already up (the state is not aSP\_Active), these messages will be buffered. After the association becomes ready (the state changes to aSP\_Active) these messages will be sent out. The MTP3\_SP\_PORT is used for receiving ASP\_MTP3\_TRANSFERreq messages from the MTP3 user, and for sending ASP\_MTP3\_TRANSFERind messages to the MTP3 user. Sending ASP\_MTP3\_PAUSE, ASP\_MTP3\_RESUME, ASP\_MTP3\_STATUS messages is not yet supported.

The communication with the SUT is done via the configured SCTP association. The M3UA peer from the SGP sends the M3UA PE M3UA messages embedded in SCTP packets. These packets are received on the SCTP\_PORT.

If the SCTP protocol identifier is set to 3 (= M3UA), M3UA PE emulates M3UA protocol and maintains static SCTP connectivity with the SUT. M3UA PE implements transfer messaging for MTP3 primitives as well as various management procedures (e.g. ASPSM, ASPTM, MGMT). Non-M3UA protocols are transferred transparently via SCTP. The SCTP protocol identifier is used to choose between M3UA and non-M3UA protocols.

Depending on the actual mode (client or server), different kinds of M3UA messages can be received and sent. E.g. the M3UA PE can’t receive M3UA\_ASPUP messages if it is in client mode, and the M3UA PE can’t receive M3UA\_ASPUP\_Ack messages if it is in server mode. Table 1 shows, how M3UA messages are handled in both modes. “Not implemented.” means, that the given message is just logged, M3UA PE doesn’t process it.

## Logging

The type of information will be logged can be categorized into the following groups. In most cases the tsp\_logVerbose module parameter must be set (see [4]) to make the log messages appear in the log files.

* Log messages from the SCTP test port.
* Changes in the state of the SCTP association.
* The type of the message received, and the type of the message sent back in response if any.
* If a message of a given type was received in an inappropriate state of the SCTP association, or in an inappropriate mode (client or server).

## Limitations

* Only a single SCTP association is supported between two endpoints. For example we can connect an IP1 and PORT1 pair to an IP2 and PORT2 pair.
* Sending ASP\_MTP3\_PAUSE, ASP\_MTP3\_RESUME, ASP\_MTP3\_STATUS messages is not supported.
* Multi-homing is not supported.