|  |  |
| --- | --- |
| 3GPP TS 38.355 V1.2.0 (2023-11) | |
| Technical Specification | |
| 3rd Generation Partnership Project;  Technical Specification Group Radio Access Network;  NR;  Sidelink Positioning Protocol (SLPP);  Protocol specification  (Release 18) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| 3GPP  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| Copyright Notification  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2023, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 7

1 Scope 8

2 References 8

3 Definitions of terms, symbols and abbreviations 8

3.1 Terms 8

3.2 Abbreviations 9

4 Functionality of Protocol 9

4.1 General 9

4.1.1 SLPP Configuration 9

4.1.2 SLPP Sessions and Transactions 9

4.1.3 SLPP Position Methods 10

4.1.4 SLPP Messages 10

4.2 Common SLPP Session Procedure 10

4.3 SLPP Transport 10

4.3.1 Transport Layer Requirements 11

4.3.2 SLPP Duplicate Detection 11

4.3.3 SLPP Acknowledgement 11

4.3.3.1 General 11

4.3.3.2 Procedure related to Acknowledgement 11

4.3.4 SLPP Retransmission 12

4.3.4.1 General 12

4.3.4.2 Procedure related to Retransmission 12

5 SLPP Procedures 13

5.1 Procedures related to capability transfer 14

5.2 Procedures related to Assistance Data Transfer 14

5.3 Procedures related to Location Information Transfer 14

5.4 Error Handling Procedures 14

5.5 Abort Procedure 14

6 Protocol data units, formats and parameters (ASN.1) 16

6.1 General 16

6.2 SLPP messages 16

6.2.1 General message structure 16

*–* *SLPP-PDU-Definitions* 16

*–* *SLPP-Message* 18

*–* *SLPP-MessageBody* 18

*–* *SLPP-TransactionID* 19

6.2.2 Message definitions 20

– *RequestCapabilities* 20

– *ProvideCapabilities* 20

– *RequestAssistanceData* 21

– *ProvideAssistanceData* 21

– *RequestLocationInformation* 22

– *ProvideLocationInformation* 23

*–* *Abort* 23

*–* *Error* 24

6.3 SLPP information elements 24

6.3.1 Common information elements 24

6.3.2 UE capability information elements 25

6.3.3 Positioning Method information elements 25

6.4 Multiplicity and type constraint values 25

*–* *End of SLPP-PDU-Definitions* 25

6.5 SLPP PDU Common Contents 25

*–* *SLPP-PDU-Common-Contents* 25

*–* *CommonIEsRequestCapabilities* 25

*–* *CommonIEsProvideCapabilities* 26

*–* *CommonIEsRequestAssistanceData* 26

*–* *CommonIEsProvideAssistanceData* 26

*–* *CommonIEsRequestLocationInformation* 27

*–* *CommonIEsProvideLocationInformation* 27

*–* *End of SLPP-PDU-Common-Contents* 27

6.6 SLPP PDU Method-A Contents 28

*–* *SLPP-PDU-Method-A-Contents* 28

*–* *Method-A-RequestCapabilities* 28

*–* *Method-A-ProvideCapabilities* 28

*–* *Method-A-RequestAssistanceData* 29

*–* *Method-A-ProvideAssistanceData* 29

*–* *Method-A-RequestLocationInformation* 29

*–* *Method-A-ProvideLocationInformation* 29

*–* *End of SLPP-PDU-* *Method-A-Contents* 30

6.7 SLPP PDU Method-B Contents 30

*–* *SLPP-PDU-Method-B-Contents* 30

*–* *Method-B-RequestCapabilities* 30

*–* *Method-B-ProvideCapabilities* 31

*–* *Method-B-RequestAssistanceData* 31

*–* *Method-B-ProvideAssistanceData* 31

*–* *Method-B-RequestLocationInformation* 32

*–* *Method-B-ProvideLocationInformation* 32

*–* *End of SLPP-PDU-* *Method-B-Contents* 32

6.8 SLPP PDU Method-C Contents 32

*–* *SLPP-PDU-Method-C-Contents* 32

*–* *Method-C-RequestCapabilities* 33

*–* *Method-C-ProvideCapabilities* 33

*–* *Method-C-RequestAssistanceData* 33

*–* *Method-C-ProvideAssistanceData* 34

*–* *Method-C-RequestLocationInformation* 34

*–* *Method-C-ProvideLocationInformation* 34

*–* *End of SLPP-PDU-* *Method-C-Contents* 35

Annex <X> (informative): Change history 36

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

# 1 Scope

The present document specifies the Sidelink Positioning Protocol (SLPP) for the interface between UEs and between UE and LMF.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 38.331: "NR; Radio Resource Control (RRC); Protocol specification".

[3] 3GPP TS 38.305: "NG Radio Access Network (NG-RAN); Stage 2 functional specification of User Equipment (UE) positioning in NG-RAN".

[4] ITU-T Recommendation X.691 (07/2002) "Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)" (Same as the ISO/IEC International Standard 8825-2).

[5] 3GPP TS 23.273: "5G System (5GS) Location Services (LCS); Stage 2".

[6] 3GPP TS 23.271: "Functional stage 2 description of Location Services (LCS)".

[7] 3GPP TS 23.032: "Universal Geographical Area Description (GAD)".

[8] 3GPP TR 38.901: "Technical Specification Group Radio Access Network; Study on channel model for frequencies from 0.5 to 100 GHz".

…

[x] <doctype> <#>[ ([up to and including]{yyyy[-mm]|V<a[.b[.c]]>}[onwards])]: "<Title>".

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

**Field:** The individual contents of an information element are referred to as fields.

**Ranging**: Refers to the determination of the distance between two UEs or more UEs and/or the direction of one UE (i.e. Target UE) from another UE via PC5 interface.

**Anchor UE**: A UE, supporting positioning of target UE, e.g. by transmitting and/or receiving reference signals for positioning, providing positioning-related information, etc. over the Sidelink interface.

**Target UE**: A UE whose distance, direction and/or position is measured with the support from one or multiple Anchor UEs using Sidelink in the Ranging based service and Sidelink positioning.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

LMF Location Management Function

LOS Line-of-Sight

NLOS Non-Line-of-Sight

SL Sidelink

SL-AoA Sidelink Angle-of-Arrival

SLPP Sidelink Positioning Protocol

SL-PRS Sidelink Positioning Reference Signals

SL-PRS-RSRP Sidelink Positioning Reference Signals based Reference Signal Received Power

SL-PRS-RSRPP Sidelink Positioning Reference Signals based Reference Signal Received Path Power

SL-PRS-RSTD Sidelink Positioning Reference Signals based Reference Signal Time Difference

SL-PRS-RTOA Sidelink Positioning Reference Signals based Relative Time of Arrival

SL-RTT Sidelink Round Trip Time

SL-TDOA Sidelink Time Difference Of Arrival

SL-TOA Sidelink Time Of Arrival

UE User Equipment

# 4 Functionality of Protocol

## 4.1 General

### 4.1.1 SLPP Configuration

SLPP is used point-to-point between Endpoint A and Endpoint B in order to obtain absolute position, relative position, or ranging information of target UE using sidelink measurements obtained by one or more reference sources. Figure 4.1.1-1 shows the configuration as applied to the control-plane location solution for NG-RAN (as defined in TS 38.305 [3] and TS 23.273 [5]).



Figure 4.1.1-1: SLPP Configuration for Control-Plane Positioning in NG-RAN

### 4.1.2 SLPP Sessions and Transactions

An SLPP session is used between UEs or a Location Server and a UE in order to obtain location related measurements, a location estimate or to transfer assistance data. A single SLPP session is used to support a single location request (e.g., for a single SL-MT-LR, or SL-MO-LR). Multiple SLPP sessions can be used between the same endpoints to support multiple different location requests (as required by TS 23.271 [6]).

Each SLPP session comprises one or more SLPP transactions, with each SLPP transaction performing a single operation (capability exchange, assistance data transfer, or location information transfer). The SLPP transactions are realized as SLPP procedures. The instigator of an SLPP session will always instigate the first SLPP transaction, but subsequent transactions may be instigated by either end. SLPP transactions within a session may occur serially or in parallel. SLPP transactions are indicated at the SLPP protocol level with a transaction ID in order to associate messages with one another (e.g., request and response).

Messages within a transaction are linked by a common transaction identifier.

### 4.1.3 SLPP Position Methods

This version of the specification defines SL-TDOA, SL-TOA, SL-AoA and SL-RTT positioning methods.

### 4.1.4 SLPP Messages

Each SLPP transaction involves the exchange of one or more SLPP messages between Endpoint A and Endpoint B. The general format of an SLPP message consists of a set of common fields followed by a body. The body (which may be empty) contains information specific to a particular message type. Each message type contains information specific to one or more positioning methods and/or information common to all positioning methods.

The common fields are as follows:

|  |  |
| --- | --- |
| Field | **Role** |
| Session ID | Identify messages belonging to the same session |
| Transaction ID | Identify messages belonging to the same transaction |
| Transaction End Flag | Indicate when a transaction (e.g. one with periodic responses) has ended |
| Sequence Number | Enable detection of a duplicate LPP message at a receiver |
| Acknowledgement | Enable an acknowledgement to be requested and/or returned for any LPP message |

The following message types are defined:

- Request Capabilities;

- Provide Capabilities;

- Request Assistance Data;

- Provide Assistance Data;

- Request Location Information;

- Provide Location Information;

- Abort;

- Error.

## 4.2 Common SLPP Session Procedure

The purpose of this procedure is to support an SLPP session comprising a sequence of SLPP transactions. The procedure is described in Figure 4.2-1.



Figure 4.2-1 SLPP Session Procedure

1. Endpoint A, which is the Endpoint who receives the LCS request, initiates an SLPP session by sending an SLPP message containing an assigned session identifier for an initial SLPP transaction *j* to the other endpoint B.

2. Endpoints A and B may exchange further messages to continue the transaction started in step 1.

3. Either endpoint may instigate further transactions by sending additional SLPP messages.

4. A session is terminated by a final transaction *N* in which SLPP messages will be exchanged between the two endpoints.

Within the same session, all constituent messages shall contain the same session identifier and within each transaction, all constituent messages shall contain the same transaction identifier. The last message sent in each transaction shall have the IE *endTransaction* set to TRUE. Transactions that occur in parallel shall use different transaction IDs; transaction IDs for completed transactions may be reused at any time after the final message of the previous transaction with the same ID is known to have been received.

## 4.3 SLPP Transport

Editor's note FFS With regards to duplicate detection: the applicability of the 10min inactivity rule. With regards to retransmission: the applicability of the timeout period of 250ms.

### 4.3.1 Transport Layer Requirements

SLPP requires reliable, in-sequence delivery of SLPP messages from the underlying transport layers. This clause describes the transport capabilities that are available within SLPP. A UE implementing SLPP shall support SLPP reliable transport (including all three of duplicate detection, acknowledgement, and retransmission).

### 4.3.2 SLPP Duplicate Detection

A sender shall include a sequence number in all SLPP messages sent for a particular location session. The sequence number shall be distinct for different SLPP messages sent by the same endpoint for the same location session (e.g., may start at zero in the first SLPP message and increase monotonically in each succeeding SLPP message). Sequence numbers used in the messages transmitted from different endpoints are independent (e.g., can be the same).

A receiver shall record the most recent received sequence number for each location session. If a message is received carrying the same sequence number as that last received for the associated location session, it shall be discarded. Otherwise (i.e., if the sequence number is different), the message shall be processed.

Sending and receiving sequence numbers shall be deleted in a server when the associated location session is terminated and shall be deleted in a target device when there has been no activity for a particular location session for 10 minutes.

### 4.3.3 SLPP Acknowledgement

#### 4.3.3.1 General

Each SLPP message may carry an acknowledgement request and/or an acknowledgement indicator. A SLPP message including an acknowledgement request (i.e., that include the IE *ackRequested* set to TRUE) shall also include a sequence number. Upon reception of an SLPP message which includes the IE *ackRequested* set to TRUE, a receiver returns an SLPP message with an acknowledgement response (i.e., that includes the *ackIndicator* IE set to the same sequence number of the message being acknowledged). An acknowledgement response may contain no SLPP message body (in which case only the sequence number being acknowledged is significant); alternatively, the acknowledgement may be sent in an SLPP message along with an SLPP message body. An acknowledgement is returned for each received SLPP message that requested an acknowledgement including any duplicate(s). Once a sender receives an acknowledgement for an SLPP message, and provided any included sequence number is matching, it is permitted to send the next SLPP message. No message reordering is needed at the receiver since this stop-and-wait method of sending ensures that messages normally arrive in the correct order.

When an SLPP message is transported via a NAS SL-MO-LR request, the message does not request an acknowledgement.

#### 4.3.3.2 Procedure related to Acknowledgement

Figure 4.3.3.2-1 shows the procedure related to acknowledgement.



Figure 4.3.3.2-1: SLPP Acknowledgement procedure

1. Endpoint A sends an SLPP message *N* to Endpoint B which includes the IE *ackRequested* set to TRUE and a sequence number.

2. If SLPP message *N* is received and Endpoint B is able to decode the *ackRequested* value and sequence number, Endpoint B shall return an acknowledgement for message *N*. The acknowledgement shall contain the IE *ackIndicator* set to the same sequence number as that in message *N*.

3. When the acknowledgement for SLPP message *N* is received and provided the included *ackIndicator* IE matches the sequence number sent in message *N*, Endpoint A sends the next SLPP message *N+1* to Endpoint B when this message is available.

### 4.3.4 SLPP Retransmission

#### 4.3.4.1 General

This capability builds on the acknowledgement and duplicate detection capabilities. When an SLPP message which requires acknowledgement is sent and not acknowledged, it is resent by the sender following a timeout period up to three times. If still unacknowledged after that, the sender aborts all SLPP activity for this Endpoint. The timeout period is determined by the sender implementation but shall not be less than a minimum value of 250 ms.

#### 4.3.4.2 Procedure related to Retransmission

Figure 4.3.4.2-1 shows the procedure related to retransmission when combined with acknowledgement and duplicate detection.



Figure 4.3.4.2-1: SLPP Retransmission procedure

1. Endpoint A sends an SLPP message *N* to Endpoint B for a particular location session and includes a request for acknowledgement along with a sequence number.

2. If SLPP message *N* is received and Endpoint B is able to decode the *ackRequested* value and sequence number (regardless of whether the message body can be correctly decoded), Endpoint B shall return an acknowledgement for message *N*. If the acknowledgement is received by Endpoint A (such that the acknowledged message can be identified and sequence numbers are matching), Endpoint A skips steps 3 and 4.

3. If the acknowledgement in step 2 is not received after a timeout period, Endpoint A shall retransmit SLPP message *N* and shall include the same sequence number as in step 1.

4. If SLPP message *N* in step 3 is received and Endpoint B is able to decode the *ackRequested* value and sequence number (regardless of whether the message body can be correctly decoded and whether or not the message is considered a duplicate), Endpoint B shall return an acknowledgement. Steps 3 may be repeated one or more times if the acknowledgement in step 4 is not received after a timeout period by Endpoint A. If the acknowledgement in step 4 is still not received after sending three retransmissions, Endpoint A shall abort all procedures and activity associated with SLPP support for this Endpoint B.

5. Once an acknowledgement in step 2 or step 4 is received, Endpoint A sends the next SLPP message *N+1* for the location session to Endpoint B when this message is available.

# 5 SLPP Procedures

## 5.1 Procedures related to capability transfer

### 5.1.1 General

The purpose of the procedures that are grouped together in this clause is to enable the transfer of capabilities from Endpoint A to Endpoint B. Capabilities in this context refer to positioning and protocol capabilities related to SLPP and the positioning methods supported by SLPP.

Editor's note FFS if the server obtains the capabilities from corresponding UE directly or for some UEs based on forwarding.

Editor's note FFS if any UEs can request the capabilities from the peer UE.

### 5.1.2 Capability Transfer procedure

The Capability Transfer procedure is shown in Figure 5.1.2-1.



Figure 5.1.2-1: SLPP Capability Transfer procedure

1. Endpoint B sends a *RequestCapabilities* message to Endpoint A. Endpoint B may indicate the types of capability needed.

2. Endpoint A responds with a *ProvideCapabilities* message to Endpoint B. The capabilities shall correspond to any capability types specified in step 1. This message shall include the *endTransaction* IE set to TRUE.

### 5.1.3 Capability Indication procedure

The Capability Indication procedure allows the target to provide unsolicited capabilities to the Endpoint B and is shown in Figure 5.1.3-1.



Figure 5.1.3-1: SLPP Capability Indication procedure

1. Endpoint A sends a *ProvideCapabilities* message to Endpoint B. This message shall include the *endTransaction* IE set to TRUE.

### 5.1.4 Transmission of SLPP Request Capabilities

When triggered to transmit a *RequestCapabilities* message, Endpoint B shall:

1> set the method specific *RequestCapabilities* IEs in accordance with the information received from upper layers.

1> deliver the message to lower layers for transmission.

### 5.1.5 Reception of SLPP Request Capabilities

Upon receiving a *RequestCapabilities* message, Endpoint A shall generate a *ProvideCapabilities* message as a response.

Endpoint A shall:

1> for each positioning method for which a request for capabilities is included in the message:

2> if Endpoint A supports this positioning method:

3> include the capabilities of Endpoint A for that supported positioning method in the response message;

1> set the IE S*LPP-TransactionID* in the response message to the same value as the IE S*LPP-TransactionID* in the received message;

1> deliver the response message to lower layers for transmission.

### 5.1.6 Transmission of SLPP Provide Capabilities

When triggered to transmit a *ProvideCapabilities* message, Endpoint A shall:

1> for each positioning method whose capabilities are to be indicated:

2> set the corresponding IE to include Endpoint A's capabilities;

1> deliver the response to lower layers for transmission.

## 5.2 Procedures related to Assistance Data Transfer

### 5.2.1 General

The purpose of the procedures that are grouped together in this clause is to enable Endpoint A to request assistance data from Endpoint B to assist in positioning, and to enable Endpoint B to transfer assistance data to Endpoint A in the absence of a request.

Editor's note Reuse the Request/Provide Assistance Data messages for server to get the assistance data from Anchor UEs. FFS on how to capture.

Editor's note FFS on whether anchor UE location can be obtained via this procedure;

Editor's note FFS whether the server can communicate with corresponding UE directly or for some UEs based on forwarding.

Editor's note FFS if any UEs can trigger the assistance data transfer procedure.

### 5.2.2 Assistance Data Transfer procedure

The Assistance Data Transfer procedure is shown in Figure 5.2.2-1.



Figure 5.2.2-1: SLPP Assistance data transfer procedure

1. Endpoint A sends a *RequestAssistanceData* message to Endpoint B.

2. Endpoint B responds with a *ProvideAssistanceData* message to Endpoint A containing assistance data. The transferred assistance data should match or be a subset of the assistance data requested in step 1. Endpoint B may also provide any not requested information that it considers useful to Endpoint A. If step 3 does not occur, this message shall set the *endTransaction* IE to TRUE.

3. Endpoint B may transmit one or more additional *ProvideAssistanceData* messages to Endpoint A containing further assistance data. The transferred assistance data should match or be a subset of the assistance data requested in step 1. Endpoint B may also provide any not requested information that it considers useful to Endpoint A. The last message shall include the *endTransaction* IE set to TRUE.

### 5.2.3 Assistance Data Delivery procedure

The Assistance Data Delivery procedure allows Endpoint B to provide unsolicited assistance data to Endpoint A and is shown in Figure 5.2.3-1.



Figure 5.2.3-1: SLPP Assistance data transfer procedure

1. Endpoint B sends a *ProvideAssistanceData* message to Endpoint A containing assistance data. If step 2 does not occur, this message shall set the *endTransaction* IE to TRUE.

2. Endpoint B may transmit one or more additional *ProvideAssistanceData* messages to Endpoint A containing additional assistance data. The last message shall include the *endTransaction* IE set to TRUE.

### 5.2.4 Transmission of SLPP Request Assistance Data

When triggered to transmit a *RequestAssistanceData* message, Endpoint A shall:

1> set the method specific *RequestAssistanceData* IEs in accordance with the information received from upper layers.

1> deliver the message to lower layers for transmission.

### 5.2.5 Reception of SLPP Request Assistance Data

Upon receiving a *RequestAssistanceData* message, Endpoint B shall generate a *ProvideAssistanceData* message as a response.

Endpoint B shall:

1> for each positioning method for which a request for assistance data is included in the message:

2> if Endpoint B supports this positioning method:

3> include the assistance data for that supported positioning method in the response message;

1> set the IE S*LPP-TransactionID* in the response message to the same value as the IE S*LPP-TransactionID* in the received message;

1> deliver the response message to lower layers for transmission.

### 5.2.6 Reception of SLPP Provide Assistance Data

Upon receiving a *ProvideAssistanceData* message, Endpoint A shall:

1> for each positioning method contained in the message:

2> deliver the related assistance data to upper layers.

## 5.3 Procedures related to Location Information Transfer

### 5.3.1 General

The purpose of the procedures that are grouped together in this clause is to enable Endpoint B to request location measurement data and/or a location estimate from Endpoint A, and to enable Endpoint A to transfer location measurement data and/or a location estimate to Endpoint B in the absence of a request.

Editor's note FFS if the server obtains the location information from corresponding UE directly or for some UEs based on forwarding.

Editor's note FFS if the procedure is used by server to obtain anchor location from the anchor UE;

Editor's note FFS if any UEs can trigger the location information transfer procedure.

### 5.3.2 Location Information Transfer procedure

The Location Information Transfer procedure is shown in Figure 5.3.2-1.



Figure 5.3.2-1: SLPP Location Information transfer procedure

1. Endpoint B sends a *RequestLocationInformation* message to Endpoint A to request location information, indicating the type of location information needed and potentially the associated QoS.

2. Endpoint A sends a *ProvideLocationInformation* message to Endpoint B to transfer location information. The location information transferred should match or be a subset of the location information requested in step 1 unless Endpoint B explicitly allows additional location information. If step 3 does not occur, this message shall set the *endTransaction* IE to TRUE.

3. If requested in step 1, Endpoint A sends additional *ProvideLocationInformation* messages to Endpoint B to transfer location information. The location information transferred should match or be a subset of the location information requested in step 1 unless Endpoint B explicitly allows additional location information. The last message shall include the *endTransaction* IE set to TRUE.

### 5.3.3 Location Information Delivery procedure

The Location Information Delivery procedure allows Endpoint A to provide unsolicited location information to Endpoint B. The procedure is shown in Figure 5.3.3-1.



Figure 5.3.3-1: SLPP Location Information Delivery procedure

1. Endpoint A sends a *ProvideLocationInformation* message to Endpoint B to transfer location information. If step 2 does not occur, this message shall set the *endTransaction* IE to TRUE.

2. Endpoint A may send one or more additional *ProvideLocationInformation* messages to Endpoint B containing additional location information data. The last message shall include the *endTransaction* IE set to TRUE.

### 5.3.4 Transmission of Request Location Information

When triggered to transmit a *RequestLocationInformation* message, Endpoint B shall:

1> set the method specific *RequestLocationInformation* IEs in accordance with the information received from upper layers.

1> deliver the message to lower layers for transmission.

### 5.3.5 Reception of Request Location Information

Upon receiving a *RequestLocationInformation* message, Endpoint A shall:

1> if the requested information is compatible with Endpoint A’s capabilities and configuration:

2> include the requested information in a *ProvideLocationInformation* message;

2> set the IE *SLPP-TransactionID* in the response to the same value as the IE *SLPP-TransactionID* in the received message;

2> deliver the *ProvideLocationInformation* message to lower layers for transmission.

1> otherwise:

2> if one or more positioning methods are included that Endpoint A does not support:

3> continue to process the message as if it contained only information for the supported positioning methods;

3> handle the signaling content of the unsupported positioning methods by SLPP error detection as in 5.4.3.

### 5.3.6 Transmission of Provide Location Information

When triggered to transmit *ProvideLocationInformation* message, Endpoint A shall:

1> for each positioning method contained in the message:

2> set the corresponding IE to include the available location information;

1> deliver the response to lower layers for transmission.

## 5.4 Error Handling Procedures

### 5.4.1 General

This clause describes how a receiving entity behaves in cases when it receives erroneous or unexpected data or detects that certain data are missing.

### 5.4.2 Procedures related to Error Indication

Figure 5.4.2-1 shows the Error indication procedure.



Figure 5.4.2-1: SLPP Error Indication procedure

1. Endpoint A sends an SLPP message to Endpoint B.

2. Endpoint B determines that the SLPP message in step 1 contains an error. Endpoint B returns an *Error* message to Endpoint A indicating the error or errors and discards the message in step 1. If Endpoint B is able to determine that the erroneous SLPP message in step 1 is an SLPP Error or Abort Message, Endpoint B discards the message in step 1 without returning an *Error* message to Endpoint A.

### 5.4.3 SLPP Error Detection

Upon receiving any SLPP message, the receiving entity shall attempt to decode the message and verify the presence of any errors and:

1> if decoding errors are encountered:

2> if the receiver cannot determine that the received message is an SLPP *Error* or *Abort* message:

3> return an SLPP *Error* message to the sender and include the received *SLPP-TransactionID*, if this was decoded, and type of error;

3> discard the received message and stop the error detection procedure;

1> if the message is a duplicate of a previously received message:

2> discard the message and stop the error detection procedure;

1> if the *SLPP-TransactionID* matches the *SLPP-TransactionID* for a procedure that is still ongoing for the same session and the message type is invalid for the current state of the procedure:

2> abort the ongoing procedure;

2> return an SLPP *Error* message to the sender and include the received transaction ID and type of error;

2> discard the message and stop the error detection procedure;

1> if the message type is an SLPP *RequestCapabilities* and some of the requested information is not supported:

2> return any information that can be provided in a normal response.

1> if the message type is an SLPP *RequestAssistanceData* or *RequestLocationInformation* and some or all of the requested information is not supported:

2> return any information that can be provided in a normal response, which includes indications on other information that is not supported.

### 5.4.4 Reception of an SLPP Error Message

Upon receiving an *Error* message, Endpoint shall:

1> abort any ongoing procedure associated with the *SLPP-TransactionID* if included in the received message.

Endpoint may:

1> restart the aborted procedure taking into consideration the returned error information.

## 5.5 Abort Procedure

### 5.5.1 General

The purpose of the abort procedure is to allow Endpoints to abort an ongoing procedure due to some unexpected event (e.g., cancellation of a location request by an LCS client). It can also be used to stop an ongoing procedure (e.g., periodic location reporting from an Endpoint).

### 5.5.2 Procedures related to Abort

Figure 5.5.2-1 shows the Abort procedure.



Figure 5.5.2-1: SLPP Abort procedure

1. A procedure P is ongoing between endpoints A and B.

2. Endpoint A determines that the procedure must be aborted and sends an *Abort* message to Endpoint B carrying the *SLPP-TransactionID* for procedure P. Endpoint B aborts procedure P.

### 5.5.3 Reception of an SLPP Abort Message

Upon receiving an *Abort* message, Endpoint shall:

1> abort any ongoing procedure associated with the *SLPP-TransactionID* indicated in the message.

# 6 Protocol data units, formats and parameters (ASN.1)

## 6.1 General

The contents of each SLPP message is specified in clause 6.2 using ASN.1 to specify the message syntax and using tables when needed to provide further detailed information about the fields specified in the message syntax. The syntax of the information elements that are defined as stand-alone abstract types is further specified in a similar manner in clause 6.3.

Editor's note FFS on Need code (e.g. how to support no UL/DL), support of delta signalling, full configuration, import IE from LPP, setup/release.

The ASN.1 in this clause uses the same format and coding conventions as described in Annex A of TS 38.331 [2].

Transfer syntax for SLPP messages is derived from their ASN.1 definitions by use of Basic Packed Encoding Rules (BASIC-PER), Unaligned Variant, as specified in ITU-T Rec. X.691 [4]. The encoded SLPP message always contains a multiple of 8 bits.

Transfer syntax for SLPP IEs is derived from their ASN.1 definitions by use of Basic Packed Encoding Rules (BASIC-PER), Unaligned Variant, as specified in ITU-T Rec. X.691 [4]. The encoded SLPP IE always contains a multiple of 8 bits. This applies when a single SLPP IE is encoded as the basic production, i.e. for other purposes than encoding the SLPP IE within an SLPP message.

When specifying information elements which are to be represented by BIT STRINGs, if not otherwise specifically stated in the field description of the concerned IE or elsewhere, the following principle applies with regards to the ordering of bits:

- The first bit (leftmost bit) contains the most significant bit (MSB);

- the last bit (rightmost bit) contains the least significant bit (LSB).

## 6.2 SLPP messages

### 6.2.1 General message structure

#### *– SLPP-PDU-Definitions*

This ASN.1 segment is the start of the SLPP PDU definitions.

-- ASN1START

-- TAG-SLPP-PDU-DEFINITIONS-START

SLPP-PDU-Definitions DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

CommonIEsRequestCapabilities,

CommonIEsProvideCapabilities,

CommonIEsRequestAssistanceData,

CommonIEsProvideAssistanceData,

CommonIEsRequestLocationInformation,

CommonIEsProvideLocationInformation

FROM

SLPP-PDU-CommonContents

CommonSL-PRS-MethodsIEsRequestCapabilities,

CommonSL-PRS-MethodsIEsProvideCapabilities,

CommonSL-PRS-MethodsIEsRequestAssistanceData,

CommonSL-PRS-MethodsIEsProvideAssistanceData,

CommonSL-PRS-MethodsIEsRequestLocationInformation,

CommonSL-PRS-MethodsIEsProvideLocationInformation

FROM

SLPP-PDU-CommonSL-PRS-MethodsContents

SL-AOA-RequestCapabilities,

SL-AOA-ProvideCapabilities,

SL-AOA-RequestAssistanceData,

SL-AOA-ProvideAssistanceData,

SL-AOA-RequestLocationInformation,

SL-AOA-ProvideLocationInformation

FROM

SLPP-PDU-SL-AOA-Contents

SL-RTT-RequestCapabilities,

SL-RTT-ProvideCapabilities,

SL-RTT-RequestAssistanceData,

SL-RTT-ProvideAssistanceData,

SL-RTT-RequestLocationInformation,

SL-RTT-ProvideLocationInformation

FROM

SLPP-PDU-SL-RTT-Contents

SL-TDOA-RequestCapabilities,

SL-TDOA-ProvideCapabilities,

SL-TDOA-RequestAssistanceData,

SL-TDOA-ProvideAssistanceData,

SL-TDOA-RequestLocationInformation,

SL-TDOA-ProvideLocationInformation

FROM

SLPP-PDU-SL-TDOA-Contents

SL-TOA-RequestCapabilities,

SL-TOA-ProvideCapabilities,

SL-TOA-RequestAssistanceData,

SL-TOA-ProvideAssistanceData,

SL-TOA-RequestLocationInformation,

SL-TOA-ProvideLocationInformation

FROM

SLPP-PDU-SL-TOA-Contents;

-- TAG-SLPP-PDU-DEFINITIONS-STOP

-- ASN1STOP

NOTE 1: An implementation needs to include only the supported "Method" PDUs. Not supported methods do not need to be included, and therefore, do not contribute to the protocol size. For example, if SL-RTT is not supported by an implementation, the *SLPP-PDU-SL-RTT-Contents* PDU does not need to be included in the protocol.

NOTE 2: An implementation supporting SL-RTT, SL-AoA, SL-TDOA, or SL-TOA must also support the *SLPP-PDU-CommonSL-PRS-MethodsContents*.

#### *– SLPP-Message*

The *SLPP-Message* provides the complete set of information for an invocation or response pertaining to an SLPP transaction.

-- ASN1START

-- TAG-SLPP-MESSAGE-START

SLPP-Message ::= SEQUENCE {

transactionID SLPP-TransactionID OPTIONAL,

endTransaction BOOLEAN,

sequenceNumber SequenceNumber,

sessionID SessionID,

acknowledgement Acknowledgement OPTIONAL,

slpp-MessageBody SLPP-MessageBody OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

SequenceNumber ::= INTEGER (0..255)

SessionID ::= OCTET STRING (SIZE (6))

Acknowledgement ::= SEQUENCE {

ackRequested BOOLEAN,

ackIndicator SequenceNumber OPTIONAL

}

-- TAG-SLPP-MESSAGE-STOP

-- ASN1STOP

|  |
| --- |
| *SLPP-Message* field descriptions |
| ***acknowledgement***  This field is included in an SLPP acknowledgement and in any SLPP message requesting an acknowledgement when SLPP operates over the control plane and is omitted otherwise.  - ackRequested: This field indicates whether an SLPP acknowledgement is requested (TRUE) or not (FALSE). A value of TRUE may only be included when an *slpp-MessageBody* is included.  - ackIndicator: This field indicates the sequence number of the message being acknowledged. |
| ***endTransaction***  This field indicates whether an SLPP message is the last message carrying an *slpp-MessageBody* in a transaction (TRUE) or not last (FALSE). |
| ***sequenceNumber***  This field may be included when SLPP operates over the control plane and an s*lpp-MessageBody* is included but shall be omitted otherwise. |
| ***sessionID***  This field indicates the session ID which is used to identify messages belonging to the same session. |
| ***slpp-MessageBody***  This field may be omitted in the case the message is sent only to acknowledge a previously received message. |
| ***transactionID***  This field is omitted if an s*lpp-MessageBody* is not present (i.e. in an SLPP message sent only to acknowledge a previously received message) or if it is not available to the transmitting entity (e.g., in an S*LPP-Error* message triggered by a message that could not be parsed). If present, this field shall be ignored at a receiver in anSLPP message for which the s*lpp-MessageBody* is not present. |

#### *– SLPP-MessageBody*

The *SLPP-MessageBody* identifies the type of an SLPP message and contains all SLPP information specifically associated with that type.

-- ASN1START

-- TAG-SLPP-MESSAGEBODY-START

SLPP-MessageBody ::= CHOICE {

c1 CHOICE {

requestCapabilities RequestCapabilities,

provideCapabilities ProvideCapabilities,

requestAssistanceData RequestAssistanceData,

provideAssistanceData ProvideAssistanceData,

requestLocationInformation RequestLocationInformation,

provideLocationInformation ProvideLocationInformation,

abort Abort,

error Error,

spare8 NULL, spare7 NULL, spare6 NULL, spare5 NULL, spare4 NULL, spare3 NULL, spare2 NULL, spare1 NULL

},

messageClassExtension SEQUENCE {}

}

-- TAG-SLPP-MESSAGEBODY-STOP

-- ASN1STOP

#### *– SLPP-TransactionID*

The *SLPP-TransactionID* identifies a particular SLPP transaction.

-- ASN1START

-- TAG-SLPP-TRANSACTIONID-START

SLPP-TransactionID ::= SEQUENCE {

transactionNumber TransactionNumber

}

TransactionNumber ::= INTEGER (0..255)

-- TAG-SLPP-TRANSACTIONID-STOP

-- ASN1STOP

### 6.2.2 Message definitions

#### – *RequestCapabilities*

-- ASN1START

-- TAG-REQUESTCAPABILITIES-START

RequestCapabilities ::= SEQUENCE {

criticalExtensions CHOICE {

requestCapabilities RequestCapabilities-IEs,

criticalExtensionsFuture SEQUENCE {}

}

}

RequestCapabilities-IEs ::= SEQUENCE {

commonIEsRequestCapabilities OCTET STRING OPTIONAL, -- Containing CommonIEsRequestCapabilities

commonSL-PRS-MethodsIEsRequestCapabilities OCTET STRING OPTIONAL, -- Containing CommonSL-PRS-MethodsIEsRequestCapabilities

sl-AoA-RequestCapabilities OCTET STRING OPTIONAL, -- Containing SL-AoA-RequestCapabilities

sl-RTT-RequestCapabilities OCTET STRING OPTIONAL, -- Containing SL-RTT-RequestCapabilities

sl-TDOA-RequestCapabilities OCTET STRING OPTIONAL, -- Containing SL-TDOA-RequestCapabilities

sl-TOA-RequestCapabilities OCTET STRING OPTIONAL, -- Containing SL-TOA-RequestCapabilities

lateNonCriticalExtension OCTET STRING OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-REQUESTCAPABILITIES-STOP

-- ASN1STOP

#### – *ProvideCapabilities*

-- ASN1START

-- TAG-PROVIDECAPABILITIES-START

ProvideCapabilities ::= SEQUENCE {

criticalExtensions CHOICE {

provideCapabilities ProvideCapabilities-IEs,

criticalExtensionsFuture SEQUENCE {}

}

}

ProvideCapabilities-IEs ::= SEQUENCE {

commonIEsProvideCapabilities OCTET STRING OPTIONAL, -- Containing CommonIEsProvideCapabilities

commonSL-PRS-MethodsIEsProvideCapabilities OCTET STRING OPTIONAL, -- Containing CommonSL-PRS-MethodsIEsProvideCapabilities

sl-AOA-ProvideCapabilities OCTET STRING OPTIONAL, -- Containing SL-AOA-ProvideCapabilities

sl-RTT-ProvideCapabilities OCTET STRING OPTIONAL, -- Containing SL-RTT-ProvideCapabilities

sl-TDOA-ProvideCapabilities OCTET STRING OPTIONAL, -- Containing SL-TDOA-ProvideCapabilities

sl-TOA-ProvideCapabilities OCTET STRING OPTIONAL, -- Containing SL-TOA-ProvideCapabilities

lateNonCriticalExtension OCTET STRING OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-PROVIDECAPABILITIES-STOP

-- ASN1STOP

#### – *RequestAssistanceData*

-- ASN1START

-- TAG-REQUESTASSISTANCEDATA-START

RequestAssistanceData ::= SEQUENCE {

criticalExtensions CHOICE {

requestAssistanceData RequestAssistanceData-IEs,

criticalExtensionsFuture SEQUENCE {}

}

}

RequestAssistanceData-IEs ::= SEQUENCE {

commonIEsRequestAssistanceData OCTET STRING OPTIONAL, -- Containing CommonIEsRequestAssistanceData

commonSL-PRS-MethodsIEsRequestAssistanceData OCTET STRING OPTIONAL, -- Containing CommonSL-PRS-MethodsIEsRequestAssistanceData

sl-AOA-RequestAssistanceData OCTET STRING OPTIONAL, -- Containing SL-AOA-RequestAssistanceData

sl-RTT-RequestAssistanceData OCTET STRING OPTIONAL, -- Containing SL-RTT-RequestAssistanceData

slSL-TDOA-RequestAssistanceData OCTET STRING OPTIONAL, -- Containing SL-TDOA-RequestAssistanceData

sl-TOA-RequestAssistanceData OCTET STRING OPTIONAL, -- Containing SL-TOA-RequestAssistanceData

lateNonCriticalExtension OCTET STRING OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-REQUESTASSISTANCEDATA-STOP

-- ASN1STOP

#### – *ProvideAssistanceData*

-- ASN1START

-- TAG-PROVIDEASSISTANCEDATA-START

ProvideAssistanceData ::= SEQUENCE {

criticalExtensions CHOICE {

provideAssistanceData ProvideAssistanceData-IEs,

criticalExtensionsFuture SEQUENCE {}

}

}

ProvideAssistanceData-IEs ::= SEQUENCE {

commonIEsProvideAssistanceData OCTET STRING OPTIONAL, -- Containing CommonIEsProvideAssistanceData

commonSL-PRS-MethodsIEsProvideAssistanceData OCTET STRING OPTIONAL, -- Containing CommonSL-PRS-MethodsIEsProvideAssistanceData

sl-AOA-ProvideAssistanceData OCTET STRING OPTIONAL, -- Containing SL-AOA-ProvideAssistanceData

sl-RTT-ProvideAssistanceData OCTET STRING OPTIONAL, -- Containing SL-RTT-ProvideAssistanceData

sl-TDOA-ProvideAssistanceData OCTET STRING OPTIONAL, -- Containing SL-TDOA-ProvideAssistanceData

sl-TOA-ProvideAssistanceData OCTET STRING OPTIONAL, -- Containing SL-TOA-ProvideAssistanceData

lateNonCriticalExtension OCTET STRING OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-PROVIDEASSISTANCEDATA-STOP

-- ASN1STOP

#### – *RequestLocationInformation*

-- ASN1START

-- TAG-REQUESTLOCATIONINFORMATION-START

RequestLocationInformation ::= SEQUENCE {

criticalExtensions CHOICE {

requestLocationInformation RequestLocationInformation-IEs,

criticalExtensionsFuture SEQUENCE {}

}

}

RequestLocationInformation-IEs ::= SEQUENCE {

commonIEsRequestLocationInformation OCTET STRING OPTIONAL, -- Containing CommonIEsRequestLocationInformation

commonSL-PRS-MethodsIEsRequestLocationInformation OCTET STRING OPTIONAL, -- Containing CommonSL-PRS-MethodsIEsRequestLocationInformation

sl-AOA-RequestLocationInformation OCTET STRING OPTIONAL, -- Containing SL-AOA-RequestLocationInformation

sl-RTT-RequestLocationInformation OCTET STRING OPTIONAL, -- Containing SL-RTT-RequestLocationInformation

sl-TDOA-RequestLocationInformation OCTET STRING OPTIONAL, -- Containing SL-TDOA-RequestLocationInformation

sl-TOA-RequestLocationInformation OCTET STRING OPTIONAL, -- Containing SL-TOA-RequestLocationInformation

lateNonCriticalExtension OCTET STRING OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-REQUESTLOCATIONINFORMATION-STOP

-- ASN1STOP

#### – *ProvideLocationInformation*

-- ASN1START

-- TAG-PROVIDELOCATIONINFORMATION-START

ProvideLocationInformation ::= SEQUENCE {

criticalExtensions CHOICE {

provideLocationInformation ProvideLocationInformation-IEs,

criticalExtensionsFuture SEQUENCE {}

}

}

ProvideLocationInformation-IEs ::= SEQUENCE {

commonIEsProvideLocationInformation OCTET STRING OPTIONAL, -- Containing CommonIEsProvideLocationInformation

commonSL-PRS-MethodsIEsProvideLocationInformation OCTET STRING OPTIONAL, -- Containing CommonSL-PRS-MethodsIEsProvideLocationInformation

sl-AOA-ProvideLocationInformation OCTET STRING OPTIONAL, -- Containing SL-AOA-ProvideLocationInformation

sl-RTT-ProvideLocationInformation OCTET STRING OPTIONAL, -- Containing SL-RTT-ProvideLocationInformation

sl-TDOA-ProvideLocationInformation OCTET STRING OPTIONAL, -- Containing SL-TDOA-ProvideLocationInformation

sl-TOA-ProvideLocationInformation OCTET STRING OPTIONAL, -- Containing SL-TOA-ProvideLocationInformation

lateNonCriticalExtension OCTET STRING OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-PROVIDELOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– Abort*

-- ASN1START

-- TAG-ABORT-START

Abort ::= SEQUENCE {

criticalExtensions CHOICE {

abort Abort-IEs,

criticalExtensionsFuture SEQUENCE {}

}

}

Abort-IEs ::= SEQUENCE {

commonIEsAbort CommonIEsAbort OPTIONAL,

lateNonCriticalExtension OCTET STRING OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-ABORT-STOP

-- ASN1STOP

#### *– Error*

-- ASN1START

-- TAG-ERROR-START

Error ::= CHOICE {

criticalExtensions CHOICE {

error Error-IEs,

criticalExtensionsFuture SEQUENCE {}

}

}

Error-IEs ::= SEQUENCE {

commonIEsError CommonIEsError OPTIONAL,

lateNonCriticalExtension OCTET STRING OPTIONAL,

nonCriticalExtension SEQUENCE {} OPTIONAL

}

-- TAG-ERROR-STOP

-- ASN1STOP

## 6.3 SLPP information elements

### 6.3.1 Common information elements

#### *– CommonIEsAbort*

The *CommonIEsAbort* carries common IEs for an Abort SLPP message Type.

-- ASN1START

-- TAG-COMMONIESABORT-START

CommonIEsAbort ::= SEQUENCE {

abortCause ENUMERATED { undefined, stopPeriodicReporting }

}

-- TAG-COMMONIESABORT-STOP

-- ASN1STOP

|  |
| --- |
| *CommonIEsAbort* field descriptions |
| ***abortCause***  This IE defines the request to abort an ongoing procedure. The abort cause '*stopPeriodicReporting*' should be used by the location server to stop any ongoing location reporting configured as *periodicalReporting* or *triggeredReporting* in the *CommonIEsRequestLocationInformation*. |

#### – *CommonIEsError*

The *CommonIEsError* carries common IEs for an Error SLPP message Type.

-- ASN1START

-- TAG-COMMONIESERROR-START

CommonIEsError ::= SEQUENCE {

errorCause ENUMERATED { undefined, slppMessageHeaderError, slppMessageBodyError, incorrectDataValue }

}

-- TAG-COMMONIESERROR-STOP

-- ASN1STOP

|  |
| --- |
| *CommonIEsError* field descriptions |
| ***errorCause***  This IE defines the cause for an error. '*slppMessageHeaderError*' and '*slppMessageBodyError*' is used if a receiver is able to detect a coding error in the SLPP header (i.e., in the common fields) or SLPP message body respectively. '*incorrectDataValue*' is used if a receiver receives an incorrect data value. |

### 6.3.2 UE capability information elements

### 6.3.3 Positioning Method information elements

## 6.4 Multiplicity and type constraint values

#### *– Multiplicity and type constraint definitions*

-- ASN1START

-- TAG-MULTIPLICITY-AND-TYPE-CONSTRAINT-DEFINITIONS-START

slMaxTxUEs INTEGER ::= 256 -- Max Tx UEs per Rx UE, FFS on the value

-- TAG-MULTIPLICITY-AND-TYPE-CONSTRAINT-DEFINITIONS-STOP

-- ASN1STOP

#### *– End of SLPP-PDU-Definitions*

-- ASN1START

END

-- ASN1STOP

## 6.5 SLPP PDU Common Contents

#### *– SLPP-PDU-CommonContents*

This ASN.1 segment is the start of the SLPP PDU Common Contents definitions.

-- ASN1START

-- TAG-SLPP-PDU-COMMONCONTENTS-START

SLPP-PDU-Common-Contents DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- TAG-SLPP-PDU-COMMON-CONTENTS-STOP

-- ASN1STOP

#### *– CommonIEsRequestCapabilities*

-- ASN1START

-- TAG-COMMONIESREQUESTCAPABILITIES-START

CommonIEsRequestCapabilities ::= SEQUENCE {

}

-- TAG-COMMONIESREQUESTCAPABILITIES-STOP

-- ASN1STOP

#### *– CommonIEsProvideCapabilities*

-- ASN1START

-- TAG-COMMONIESPROVIDECAPABILITIES-START

CommonIEsProvideCapabilities ::= SEQUENCE {

}

-- TAG-COMMONIESPROVIDECAPABILITIES-STOP

-- ASN1STOP

#### *– CommonIEsRequestAssistanceData*

-- ASN1START

-- TAG-COMMONIESREQUESTASSISTANCEDATA-START

CommonIEsRequestAssistanceData ::= SEQUENCE {

}

-- TAG-COMMONIESREQUESTASSISTANCEDATA-STOP

-- ASN1STOP

#### *– CommonIEsProvideAssistanceData*

-- ASN1START

-- TAG-COMMONIESPROVIDEASSISTANCEDATA-START

CommonIEsProvideAssistanceData ::= SEQUENCE {

}

-- TAG-COMMONIESPROVIDEASSISTANCEDATA-STOP

-- ASN1STOP

#### *– CommonIEsRequestLocationInformation*

The *CommonIEsRequestLocationInformation* carries common IEs for a Request Location Information SLPP message Type.

-- ASN1START

-- TAG-COMMONIESREQUESTLOCATIONINFORMATION-START

CommonIEsRequestLocationInformation ::= SEQUENCE {

locationInformationType LocationInformationType,

periodicalReporting PeriodicalReportingCriteria OPTIONAL,

additionalInformation AdditionalInformation OPTIONAL,

qos QoS OPTIONAL,

environment Environment OPTIONAL,

...

}

LocationInformationType ::= ENUMERATED { locationEstimateRequired, locationMeasurementsRequired, locationEstimatePreferred,

locationMeasurementsPreferred, rangenEstimateRequired, rangeMeasurementsRequired, rangeEstimatePreferred,

rangeMeasurementsPreferred}

PeriodicalReportingCriteria ::= SEQUENCE {

reportingAmount ENUMERATED { ra1, ra2, ra4, ra8, ra16, ra32, ra64, ra-Infinity },

reportingInterval ENUMERATED { noPeriodicalReporting, ri0-25, ri0-5, ri1, ri2, ri4, ri8, ri16, ri32, ri64}

}

ReportingDuration ::= INTEGER (0..255)

AdditionalInformation ::= ENUMERATED { onlyReturnInformationRequested, mayReturnAdditionalInformation}

QoS ::= SEQUENCE {

horizontalAccuracy HorizontalAccuracy OPTIONAL,

verticalCoordinateRequest BOOLEAN,

verticalAccuracy VerticalAccuracy OPTIONAL,

responseTime ResponseTime OPTIONAL,

velocityRequest BOOLEAN,

...

}

HorizontalAccuracy ::= SEQUENCE {

accuracy INTEGER(0..255),

confidence INTEGER(0..100),

...

}

VerticalAccuracy ::= SEQUENCE {

accuracy INTEGER(0..255),

confidence INTEGER(0..100),

...

}

ResponseTime ::= SEQUENCE {

time INTEGER (1..128),

responseTimeEarlyFix INTEGER (1..128) OPTIONAL,

unit ENUMERATED { ten-seconds, ten-milli-seconds, ... } OPTIONAL,

...

}

Environment ::= ENUMERATED { badArea, notBadArea, mixedArea}

-- TAG-COMMONIESREQUESTLOCATIONINFORMATION-STOP

-- ASN1STOP

|  |
| --- |
| *CommonIEsRequestLocationInformation* field descriptions |
| ***additionalInformation***  This IE indicates whether a target device is allowed to return additional information to that requested. If this IE indicates '*onlyReturnInformationRequested'* then the target device shall not return any additional information to that requested by the server. If this IE indicates '*mayReturnAdditionalInformation'* then the target device may return additional information to that requested by the server. If a location estimate is returned, any additional information is restricted to that associated with a location estimate (e.g. might include velocity if velocity was not requested but cannot include measurements). If measurements are returned, any additional information is restricted to additional measurements (e.g. might include SL-AoA measurements if SL-TDOA measurements were requested but not SL-AoA measurements). |
| ***environment***  This field provides the target device with information about expected multipath and non line of sight (NLOS) in the current area. The following values are defined:  - badArea: possibly heavy multipath and NLOS conditions (e.g. bad urban or urban).  - notBadArea: no or light multipath and usually LOS conditions (e.g. suburban or rural).  - mixedArea: environment that is mixed or not defined.  If this field is absent, a default value of 'mixedArea' applies. |
| ***locationInformationType***  This IE indicates whether the server requires a location estimate or measurements. For '*locationEstimateRequired*' or '*rangeEstimateRequired*' , the target device shall return a location or range estimate if possible, or indicate a location error if not possible. For '*locationMeasurementsRequired*  '*rangeMeasurementsRequired*'', the target device shall return measurements if possible, or indicate a location error if not possible. For '*locationEstimatePreferred*' or '*rangeEstimatePreferred*', the target device shall return a location or range estimate if possible, but may also or instead return measurements for any requested position methods for which a location estimate is not possible. For '*locationMeasurementsPreferred or* '*rangeMeasurementsPreferred*'', the target device shall return location or range measurements if possible, but may also or instead return a location estimate for any requested position methods for which return of location measurements is not possible. |
| ***periodicalReporting***  This IE indicates that periodic reporting is requested and comprises the following subfields:  - ***reportingAmount*** indicates the number of periodic location information reports requested. Enumerated values correspond to 1, 2, 4, 8, 16, 32, 64, or infinite/indefinite number of reports. If the *reportingAmount* is '*infinite/indefinite'*, the target device should continue periodic reporting until an SLPP *Abort* message is received. The value '*ra1*' shall not be used by a sender.  - ***reportingInterval*** indicates the interval between location information reports and the response time requirement for the first location information report. Enumerated values ri0-25, ri0-5, ri1, ri2, ri4, ri8, ri16, ri32, ri64 correspond to reporting intervals of 1, 2, 4, 8, 10, 16, 20, 32, and 64 seconds, respectively. Measurement reports containing no measurements or no location estimate are required when a *reportingInterval* expires before a target device is able to obtain new measurements or obtain a new location estimate. The value '*noPeriodicalReporting*' shall not be used by a sender. |
| ***qos***  This IE indicates the quality of service and comprises a number of sub-fields. In the case of measurements, some of the sub-fields apply to the location estimate that could be obtained by the server from the measurements provided by the target device assuming that the measurements are the only sources of error. Fields are as follows:  - ***horizontalAccuracy*** indicates the maximum horizontal error in the location estimate at an indicated confidence level. The '*accuracy*' corresponds to the encoded uncertainty as defined in TS 23.032 [7] and '*confidence*' corresponds to confidence as defined in TS 23.032 [7].  - ***verticalCoordinateRequest*** indicates whether a vertical coordinate is required (TRUE) or not (FALSE)  - ***verticalAccuracy*** indicates the maximum vertical error in the location estimate at an indicated confidence level and is only applicable when a vertical coordinate is requested. The '*accuracy*' corresponds to the encoded uncertainty altitude as defined in TS 23.032 [7] and '*confidence*' corresponds to confidence as defined in TS 23.032 [7].  - ***responseTime***  - ***time*** indicates the maximum response time as measured between receipt of the *RequestLocationInformation* and transmission of a *ProvideLocationInformation*. If the *unit* field is absent, this is given as an integer number of seconds between 1 and 128. If the *unit* field is present with enumerated value '*ten-seconds*', the maximum response time is given in units of 10-seconds, between 10 and 1280 seconds. If the *unit* field is present with enumerated value '*ten-milli-seconds*', the maximum response time is given in units of 10-milli-seconds, between 0.01 and 1.28 seconds. If the *periodicalReporting* IE is included in *CommonIEsRequestLocationInformation*, this field should not be included by the location server and shall be ignored by the target device (if included).  - ***responseTimeEarlyFix*** indicates the maximum response time as measured between receipt of the *RequestLocationInformation* and transmission of a *ProvideLocationInformation* containing early location measurements or an early location estimate. If the *unit* field is absent, this is given as an integer number of seconds between 1 and 128. If the *unit* field is present with enumerated value '*ten-seconds*', the maximum response time is given in units of 10-seconds, between 10 and 1280 seconds. If the *unit* field is present with enumerated value '*ten-milli-seconds*', the maximum response time is given in units of 10-milli-seconds, between 0.01 and 1.28 seconds. When this IE is included, a target should send a *ProvideLocationInformation* (or more than one *ProvideLocationInformation* if location information will not fit into a single message) containing early location information according to the *responseTimeEarlyFix* IE and a subsequent *ProvideLocationInformation* (or more than one *ProvideLocationInformation* if location information will not fit into a single message) containing final location information according to the *time* IE. A target shallomit sending a *ProvideLocationInformation* if the early location information is not available at the expiration of the time value in the *responseTimeEarlyFix* IE. A server should set the *responseTimeEarlyFix* IE to a value less than that for the *time* IE. A target shall ignore the *responseTimeEarlyFix* IE if its value is not less than that for the *time* IE.  - ***unit*** indicates the unit of the *time* and *responseTimeEarlyFix* fields. Enumerated value '*ten-seconds*' corresponds to a resolution of 10 seconds. Enumerated value '*ten-milli-seconds*' corresponds to a resolution of 0.01 seconds. If this field is absent, the unit/resolution is 1 second.  - ***velocityRequest*** indicates whether velocity (or measurements related to velocity) is requested (TRUE) or not (FALSE).  All QoS requirements shall be obtained by the target device to the degree possible but it is permitted to return a response that does not fulfill all QoS requirements if some were not attainable. |

#### *– CommonIEsProvideLocationInformation*

The *CommonIEsProvideLocationInformation* carries common IEs for a Provide Location Information SLPP message Type.

-- ASN1START

-- TAG-COMMONIESPROVIDELOCATIONINFORMATION-START

CommonIEsProvideLocationInformation ::= SEQUENCE {

locationEstimate LocationCoordinates OPTIONAL, -- [locationTargetUe-sl-pos](Up to RAN2)

velocityEstimate Velocity OPTIONAL,

locationError LocationError OPTIONAL,

earlyFixReport EarlyFixReport OPTIONAL,

...

}

LocationCoordinates ::= CHOICE {

ellipsoidPoint Ellipsoid-Point,

ellipsoidPointWithUncertaintyCircle Ellipsoid-PointWithUncertaintyCircle,

ellipsoidPointWithUncertaintyEllipse EllipsoidPointWithUncertaintyEllipse,

polygon Polygon,

ellipsoidPointWithAltitude EllipsoidPointWithAltitude,

ellipsoidPointWithAltitudeAndUncertaintyEllipsoid EllipsoidPointWithAltitudeAndUncertaintyEllipsoid,

ellipsoidArc EllipsoidArc,

rangeAndDirection RangeAndDirection,

...

}

Velocity ::= CHOICE {

horizontalVelocity HorizontalVelocity,

horizontalWithVerticalVelocity HorizontalWithVerticalVelocity,

horizontalVelocityWithUncertainty HorizontalVelocityWithUncertainty,

horizontalWithVerticalVelocityAndUncertainty HorizontalWithVerticalVelocityAndUncertainty,

...

}

LocationError ::= SEQUENCE {

locationfailurecause LocationFailureCause,

...

}

LocationFailureCause ::= ENUMERATED { undefined, requestedMethodNotSupported, positionMethodFailure, periodicLocationMeasurementsNotAvailable}

EarlyFixReport ::= ENUMERATED { noMoreMessages, moreMessagesOnTheWay}

Ellipsoid-Point ::= SEQUENCE {

latitudeSign ENUMERATED {north, south},

degreesLatitude INTEGER (0..8388607), -- 23 bit field

degreesLongitude INTEGER (-8388608..8388607) -- 24 bit field

}

Ellipsoid-PointWithUncertaintyCircle ::= SEQUENCE {

latitudeSign ENUMERATED {north, south},

degreesLatitude INTEGER (0..8388607), -- 23 bit field

degreesLongitude INTEGER (-8388608..8388607), -- 24 bit field

uncertainty INTEGER (0..127)

}

EllipsoidPointWithUncertaintyEllipse ::= SEQUENCE {

latitudeSign ENUMERATED {north, south},

degreesLatitude INTEGER (0..8388607), -- 23 bit field

degreesLongitude INTEGER (-8388608..8388607), -- 24 bit field

uncertaintySemiMajor INTEGER (0..127),

uncertaintySemiMinor INTEGER (0..127),

orientationMajorAxis INTEGER (0..179),

confidence INTEGER (0..100)

}

EllipsoidPointWithAltitude ::= SEQUENCE {

latitudeSign ENUMERATED {north, south},

degreesLatitude INTEGER (0..8388607), -- 23 bit field

degreesLongitude INTEGER (-8388608..8388607), -- 24 bit field

altitudeDirection ENUMERATED {height, depth},

altitude INTEGER (0..32767) -- 15 bit field

}

EllipsoidPointWithAltitudeAndUncertaintyEllipsoid ::= SEQUENCE {

latitudeSign ENUMERATED {north, south},

degreesLatitude INTEGER (0..8388607), -- 23 bit field

degreesLongitude INTEGER (-8388608..8388607), -- 24 bit field

altitudeDirection ENUMERATED {height, depth},

altitude INTEGER (0..32767), -- 15 bit field

uncertaintySemiMajor INTEGER (0..127),

uncertaintySemiMinor INTEGER (0..127),

orientationMajorAxis INTEGER (0..179),

uncertaintyAltitude INTEGER (0..127),

confidence INTEGER (0..100)

}

EllipsoidArc ::= SEQUENCE {

latitudeSign ENUMERATED {north, south},

degreesLatitude INTEGER (0..8388607), -- 23 bit field

degreesLongitude INTEGER (-8388608..8388607), -- 24 bit field

innerRadius INTEGER (0..65535), -- 16 bit field,

uncertaintyRadius INTEGER (0..127),

offsetAngle INTEGER (0..179),

includedAngle INTEGER (0..179),

confidence INTEGER (0..100)

}

RangeAndDirection ::= SEQUENCE {

range Range OPTIONAL,

azimuth Azimuth OPTIONAL,

elevation Elevation OPTIONAL

}

Range ::= SEQUENCE {

rangeResult INTEGER (0..50000),

uncertainty INTEGER (0..127),

confidence INTEGER (0..100) OPTIONAL

}

Azimuth ::= SEQUENCE {

azimuthResult INTEGER (0..359),

uncertainty INTEGER (0..127),

confidence INTEGER (0..100) OPTIONAL

}

Elevation ::= SEQUENCE {

elevationResult INTEGER (0..179),

uncertainty INTEGER (0..63),

confidence INTEGER (0..100) OPTIONAL

}

HorizontalVelocity ::= SEQUENCE {

bearing INTEGER(0..359),

horizontalSpeed INTEGER(0..2047)

}

HorizontalWithVerticalVelocity ::= SEQUENCE {

bearing INTEGER(0..359),

horizontalSpeed INTEGER(0..2047),

verticalDirection ENUMERATED{upward, downward},

verticalSpeed INTEGER(0..255)

}

HorizontalVelocityWithUncertainty ::= SEQUENCE {

bearing INTEGER(0..359),

horizontalSpeed INTEGER(0..2047),

uncertaintySpeed INTEGER(0..255)

}

HorizontalWithVerticalVelocityAndUncertainty ::= SEQUENCE {

bearing INTEGER(0..359),

horizontalSpeed INTEGER(0..2047),

verticalDirection ENUMERATED{upward, downward},

verticalSpeed INTEGER(0..255),

horizontalUncertaintySpeed INTEGER(0..255),

verticalUncertaintySpeed INTEGER(0..255)

}

Polygon ::= SEQUENCE (SIZE (3..15)) OF PolygonPoints

PolygonPoints ::= SEQUENCE {

latitudeSign ENUMERATED {north, south},

degreesLatitude INTEGER (0..8388607), -- 23 bit field

degreesLongitude INTEGER (-8388608..8388607) -- 24 bit field

}

-- TAG-COMMONIESPROVIDELOCATIONINFORMATION-STOP

-- ASN1STOP

|  |
| --- |
| *CommonIEsProvideLocationInformation* field descriptions |
| ***earlyFixReport***  This field shall be included if and only if the *ProvideLocationInformation* message contains early location measurements or an early location estimate. The target device shall set the values of this field as follows:  - noMoreMessages: This is the only or last *ProvideLocationInformation* message used to deliver the entire set of early location information.  - moreMessagesOnTheWay: This is one of multiple *ProvideLocationInformation* messages used to deliver the entire set of early location information (if early location information will not fit into a single message). |
| ***locationError***  This field shall be included if and only if a location estimate and measurements are not included in the SLPP PDU. The field includes information concerning the reason for the lack of location information. The *LocationFailureCause* '*periodicLocationMeasurementsNotAvailable*' shall be used by the target device if periodic location reporting was requested, but no measurements or location estimate are available when *the reportingInterval* expired. |
| ***locationEstimate***  This field provides a location estimate using one of the geographic shapes defined in TS 23.032 [7]. Coding of the values of the various fields internal to each geographic shape follow the rules in TS 23.032 [7]. The conditions for including this field are defined for the *locationInformationType* field in a Request Location Information message. |
| ***velocityEstimate***  This field provides a velocity estimate using one of the velocity shapes defined in TS 23.032 [7]. Coding of the values of the various fields internal to each velocity shape follow the rules in TS 23.032 [7]. |

#### *– End of SLPP-PDU-CommonContents*

-- ASN1START

END

-- ASN1STOP

## 6.6 SLPP PDU Common SL-PRS Methods Contents

#### *– SLPP-PDU-CommonSL-PRS-MethodsContents*

This ASN.1 segment is the start of the SLPP PDU Common SL-PRS Methods Contents definitions.

-- ASN1START

-- TAG-SLPP-PDU-COMMONSL-PRS-METHODSCONTENTS-START

SLPP-PDU-CommonSL-PRS-MethodsContents DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

LocationCoordinates

FROM

SLPP-PDU-CommonContents

slMaxTxUEs

FROM

SLPP-PDU-Definitions;

-- TAG-SLPP-PDU-COMMONSL-PRS-METHODSCONTENTS-STOP

-- ASN1STOP

#### *– CommonSL-PRS-MethodsIEsRequestCapabilities*

-- ASN1START

-- TAG-COMMONSL-PRS-METHODSIESREQUESTCAPABILITIES-START

CommonSL-PRS-MethodsIEsRequestCapabilities ::= SEQUENCE {

}

-- TAG-COMMONSL-PRS-METHODSIESREQUESTCAPABILITIES-STOP

-- ASN1STOP

#### *– CommonSL-PRS-MethodsIEsProvideCapabilities*

-- ASN1START

-- TAG-COMMONSL-PRS-METHODSIESPROVIDECAPABILITIES-START

CommonSL-PRS-MethodsIEsProvideCapabilities ::= SEQUENCE {

}

-- TAG-COMMONSL-PRS-METHODSIESPROVIDECAPABILITIES-STOP

-- ASN1STOP

#### *– CommonSL-PRS-MethodsIEsRequestAssistanceData*

-- ASN1START

-- TAG-COMMONSL-PRS-METHODSIESREQUESTASSISTANCEDATA-START

CommonSL-PRS-MethodsIEsRequestAssistanceData ::= SEQUENCE {

sl-PRS-SequenceID INTEGER(0..4095) OPTIONAL, -- SL PRS sequence generation, from server to Tx UE, FFS on signalling

...

}

-- TAG-COMMONSL-PRS-METHODSIESREQUESTASSISTANCEDATA-STOP

-- ASN1STOP

|  |
| --- |
| *CommonSL-PRS-MethodsIEsRequestAssistanceData* field descriptions |
| ***sl-PRS-SequenceID***  This field specifies the sequence Id used to initialize cinit value used in pseudo random generator for generation of SL PRS sequence for transmission on a given SL PRS Resource. ). If the Tx UE does not receive a sequence ID via SLPP message from the server, the Tx UE is expected to select one by itself. |

#### *– CommonSL-PRS-MethodsIEsProvideAssistanceData*

-- ASN1START

-- TAG-COMMONSL-PRS-METHODSIESPROVIDEASSISTANCEDDATA-START

CommonSL-PRS-MethodsIEsProvideAssistanceData ::= SEQUENCE {

sl-PRS-AssistanceData SEQUENCE (SIZE (1..slMaxTxUEs)) OF SL-PRS-Config OPTIONAL,

...

}

SL-PRS-Config ::= SEQUENCE {

-- For absolute sidelink positioning, the locations of the anchor UEs are provided to the entity that does the location calculation.

anchorUE-LocationInformation LocationCoordinates OPTIONAL,

expectedSL-AzimuthAoA-AndUncertainty INTEGER(0..3599), -- expected-SL-AoA-and-Uncertainty

expectedSL-ZenithAoA-AndUncertainty INTEGER(0..1799), -- expected-SL-AoA-and-Uncertainty

...

}

-- TAG-COMMONSL-PRS-METHODSIESPROVIDEASSISTANCEDDATA-STOP

-- ASN1STOP

|  |
| --- |
| *CommonSL-PRS-MethodsIEsProvideAssistanceData* field descriptions |
| ***anchorUE-LocationInformation***  This field provides anchor UE location information to the entity that does the location calculation for absolute SL positioning. |
| ***expectedSL-AzimuthAoA-AndUncertainty***  This field provides expected SL-AzimuthAoA and uncertainty range to a measuring UE. |
| ***expectedSL-ZenithAoA-AndUncertainty***  This field provides expected SL-ZenithAoA and uncertainty range to a measuring UE. |
| ***sl-PRS-AssistanceData***  This field specifies the sidelink PRS assistance data of Tx UEs. |

#### *– CommonSL-PRS-MethodsIEsRequestLocationInformation*

-- ASN1START

-- TAG-COMMONSL-PRS-METHODSIESREQUESTLOCATIONINFORMATION-START

CommonSL-PRS-MethodsIEsRequestLocationInformation ::= SEQUENCE {

}

-- TAG-COMMONSL-PRS-METHODSIESREQUESTLOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– Common-SL-PRS-MethodsIEsProvideLocationInformation*

-- ASN1START

-- TAG-COMMONSL-PRS-METHODSIESPROVIDELOCATIONINFORMATION-START

CommonSL-PRS-MethodsIEsProvideLocationInformation ::= SEQUENCE {

sl-PRS-SignalMeasurementInformation SL-PRS-SignalMeasurementInformation OPTIONAL,

...

}

SL-PRS-SignalMeasurementInformation ::= SEQUENCE {

sl-PRS-MeasList SL-PRS-MeasList,

...

}

SL-PRS-MeasList::= SEQUENCE (SIZE(1..slMaxTxUEs)) OF SL-PRS-MeasElement

SL-PRS-MeasElement ::= SEQUENCE {

los-NLOS-Indicator LOS-NLOS-Indicator OPTIONAL, -- sl-losNlosIndicator

sl-AzimuthAoA-FirstPathResult INTEGER (TBD) OPTIONAL, -- sl-PRS-AoA

sl-AzimuthAoA-LCS-GCS-Translation LCS-GCS-Translation OPTIONAL, -- sl-LCS-to-GCS-translation

sl-POS-ARP-ID-Rx INTEGER (1..4) OPTIONAL, -- sl-pos-arpID-Rx

sl-PRS-RxTxTimeDiffFirstPathResult INTEGER (TBD) OPTIONAL, -- sl-PRS-RxTxTimeDiff

sl-PRS-RSRP-Result INTEGER (TBD) OPTIONAL, -- sl-PRS-RSRP

sl-PRS-FirstPathRSRPP-Result INTEGER (TBD) OPTIONAL, -- sl-PRS-RSRPP

sl-PRS-AdditionalPathList SL-PRS-AdditionalPathList OPTIONAL,

sl-RSTD-FirstPathResult INTEGER (TBD) OPTIONAL, -- sl-PRS-RSTD

sl-RTOA-FirstPathResult INTEGER (TBD) OPTIONAL, -- sl-PRS-RTOA

sl-ZenithAoA-FirstPathResult INTEGER (TBD) OPTIONAL, -- sl-PRS-AoA

sl-ZenithAoA-LCS-GCS-Translation LCS-GCS-Translation OPTIONAL, -- sl-LCS-to-GCS-translation

...

}

LOS-NLOS-Indicator ::= SEQUENCE {

indicator CHOICE {

soft INTEGER (0..10),

hard BOOLEAN

},

...

}

SL-PRS-AdditionalPathList ::= SEQUENCE (SIZE(1..2)) OF SL-PRS-AdditionalPath

SL-PRS-AdditionalPath ::= SEQUENCE {

sl-AzimuthAoA-AdditionalPathResult INTEGER (TBD) OPTIONAL, -- additionalPath-SL-PRS-AoA

sl-AzimuthAoA-LCS-GCS-Translation LCS-GCS-Translation OPTIONAL, -- sl-LCS-to-GCS-translation

sl-ZenithAoA-AdditionalPathResult INTEGER (TBD) OPTIONAL, -- additionalPath-SL-PRS-AoA

sl-ZenithAoA-LCS-GCS-Translation LCS-GCS-Translation OPTIONAL, -- sl-LCS-to-GCS-translation

sl-RSTD-AdditionalPathResult INTEGER (TBD) OPTIONAL, -- additionalPath-SL-PRS-RSTD

sl-RTOA-AdditionalPathResult INTEGER (TBD) OPTIONAL, -- additionalPath-SL-PRS-RTOA

sl-PRS-RxTxTimeDiffAdditionalPathResult INTEGER (TBD) OPTIONAL, -- additionalPath-SL-PRS-Rx-Tx-TimeDiff

sl-PRS-AdditionalPathRSRPP-Result INTEGER (TBD) OPTIONAL, -- additionalPath-SL-PRS-RSRPP

...

}

LCS-GCS-Translation ::= SEQUENCE {

alpha INTEGER (0..3599),

beta INTEGER (0..3599),

gamma INTEGER (0..3599),

...

}

-- TAG-COMMONSL-PRS-METHODSIESPROVIDELOCATIONINFORMATION-STOP

-- ASN1STOP

|  |
| --- |
| *CommonSL-PRS-MethodsIEsProvideLocationInformation* field descriptions |
| ***los-NLOS-Indicator***  This field specifies the target device's best estimate of the LOS or NLOS of the UE measurements (including RSTD, RTOA, RSRP, RSRPP, AoA and UE Rx-Tx time difference). |
| ***sl-AzimuthAoA-FirstPathResult***  This field specifies the first path result of SL-AzimuthAoA. |
| ***sl-AzimuthAoA-LCS-GCS-Translation***  This field provides the angles α (bearing angle), β (downtilt angle) and γ (slant angle) for the translation of a Local Coordinate System (LCS) to a Global Coordinate System (GCS) as defined in TR 38.901 [8]. |
| ***sl-POS-ARP-ID-Rx***  This field indicates ARP ID of an ARP used for reception for per-ARP measurement reporting. The ARP ID is used to uniquely identify an ARP associated with a UE. |
| ***sl-PRS-RxTxTimeDiffFirstPathResult***  This field specifies SL Rx-Tx time difference measurement based on first path of arrival. |
| ***sl-PRS-RSRP-Result***  This field specifies the sidelink PRS reference signal received power (RSRP) measurement. |
| ***sl-PRS-FirstPathRSRPP-Result***  This field specifies the SL-RSRPP measurement based on first path of arrival. |
| ***sl-PRS-AdditionalPathList***  This field specifies the sidelink PRS measurements based on additional path of arrival. |
| ***sl-RSTD-FirstPathResult***  This field specifies the SL-RSTD measurement based on first path of arrival. |
| ***sl-RTOA-FirstPathResult***  This field specifies the SL-RTOA measurement based on first path of arrival. |
| ***sl-ZenithAoA-FirstPathResult***  This field specifies the first path result of SL-ZenithAoA. |
| ***sl-AzimuthAoA-LCS-GCS-Translation***  This field provides the angles α (bearing angle), β (downtilt angle) and γ (slant angle) for the translation of a Local Coordinate System (LCS) to a Global Coordinate System (GCS) as defined in TR 38.901 [8]. |

#### *– End of SLPP-PDU-CommonSL-PRS-MethodsContents*

-- ASN1START

END

-- ASN1STOP

## 6.7 SLPP PDU SL-AoA Contents

#### *– SLPP-PDU-SL-AoA-Contents*

This ASN.1 segment is the start of the SLPP PDU SL-AoA Contents definitions.

-- ASN1START

-- TAG-SLPP-PDU-SL-AOA-CONTENTS-START

SLPP-PDU-SL-AoA-Contents DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- TAG-SLPP-PDU-SL-A0A-CONTENTS-STOP

-- ASN1STOP

#### *– SL-AoA-RequestCapabilities*

-- ASN1START

-- TAG-SL-AOA-REQUESTCAPABILITIES-START

SL-AoA-RequestCapabilities ::= SEQUENCE {

}

-- TAG-SL-AOA-REQUESTCAPABILITIES-STOP

-- ASN1STOP

#### *– SL-AoA-ProvideCapabilities*

-- ASN1START

-- TAG-SL-AOA-PROVIDECAPABILITIES-START

SL-AoA-ProvideCapabilities ::= SEQUENCE {

}

-- TAG-SL-AOA-PROVIDECAPABILITIES-STOP

-- ASN1STOP

#### *– SL-AoA-RequestAssistanceData*

-- ASN1START

-- TAG-SL-AOA-REQUESTASSISTANCEDATA-START

SL-AoA-RequestAssistanceData ::= SEQUENCE {

}

-- TAG-SL-AOA-REQUESTASSISTANCEDATA-STOP

-- ASN1STOP

#### *– SL-AoA-ProvideAssistanceData*

-- ASN1START

-- TAG-SL-AOA-PROVIDEASSISTANCEDATA-START

SL-AoA-ProvideAssistanceData ::= SEQUENCE {

}

-- TAG-SL-AoA-PROVIDEASSISTANCEDATA-STOP

-- ASN1STOP

#### *– SL-AoA-RequestLocationInformation*

-- ASN1START

-- TAG-SL-AOA-REQUESTLOCATIONINFORMATION-START

SL-AoA-RequestLocationInformation ::= SEQUENCE {

}

-- TAG-SL-AOA-REQUESTLOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– SL-AoA-ProvideLocationInformation*

-- ASN1START

-- TAG-SL-AOA-PROVIDELOCATIONINFORMATION-START

SL-AoA-ProvideLocationInformation ::= SEQUENCE {

}

-- TAG-SL-AOA-PROVIDELOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– End of SLPP-PDU-SL-AoA-Contents*

-- ASN1START

END

-- ASN1STOP

## 6.8 SLPP PDU SL-RTT Contents

#### *– SLPP-PDU-SL-RTT-Contents*

This ASN.1 segment is the start of the SLPP PDU SL-RTT Contents definitions.

-- ASN1START

-- TAG-SLPP-PDU-SL-RTT-CONTENTS-START

SLPP-PDU-SL-RTT-CONTENTS DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- TAG-SLPP-PDU-SL-RTT-CONTENTS-STOP

-- ASN1STOP

#### *– SL-RTT-RequestCapabilities*

-- ASN1START

-- TAG-SL-RTT-REQUESTCAPABILITIES-START

SL-RTT-RequestCapabilities ::= SEQUENCE {

}

-- TAG-SL-RTT-REQUESTCAPABILITIES-STOP

-- ASN1STOP

#### *– SL-RTT-ProvideCapabilities*

-- ASN1START

-- TAG-SL-RTT-PROVIDECAPABILITIES-START

SL-RTT-ProvideCapabilities ::= SEQUENCE {

}

-- TAG-SL-RTT-PROVIDECAPABILITIES-STOP

-- ASN1STOP

#### *– SL-RTT-RequestAssistanceData*

-- ASN1START

-- TAG-SL-RTT-REQUESTASSISTANCEDATA-START

SL-RTT-RequestAssistanceData ::= SEQUENCE {

}

-- TAG-SL-RTT-REQUESTASSISTANCEDATA-STOP

-- ASN1STOP

#### *– SL-RTT-ProvideAssistanceData*

-- ASN1START

-- TAG-SL-RTT-PROVIDEASSISTANCEDATA-START

SL-RTT-ProvideAssistanceData ::= SEQUENCE {

}

-- TAG-SL-RTT-PROVIDEASSISTANCEDATA-STOP

-- ASN1STOP

#### *– SL-RTT-RequestLocationInformation*

-- ASN1START

-- TAG-SL-RTT-REQUESTLOCATIONINFORMATION-START

SL-RTT-RequestLocationInformation ::= SEQUENCE {

}

-- TAG-SL-RTT-REQUESTLOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– SL-RTT-ProvideLocationInformation*

-- ASN1START

-- TAG-SL-RTT-PROVIDELOCATIONINFORMATION-START

SL-RTT-ProvideLocationInformation ::= SEQUENCE {

}

-- TAG-SL-RTT-PROVIDELOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– End of SLPP-PDU-SL-RTT-Contents*

-- ASN1START

END

-- ASN1STOP

## 6.9 SLPP PDU SL-TDOA Contents

#### *– SLPP-PDU-**SL-TDOA-Contents*

This ASN.1 segment is the start of the SLPP PDU SL-TDOA Contents definitions.

-- ASN1START

-- TAG-SLPP-PDU-SL-TDOA-CONTENTS-START

SLPP-PDU-SL-TDOA-CONTENTS DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- TAG-SLPP-PDU-SL-TDOA-CONTENTS-STOP

-- ASN1STOP

#### *– SL-TDOA-RequestCapabilities*

-- ASN1START

-- TAG-SL-TDOA-REQUESTCAPABILITIES-START

SL-TDOA-RequestCapabilities ::= SEQUENCE {

}

-- TAG-SL-TDOA-REQUESTCAPABILITIES-STOP

-- ASN1STOP

#### *– SL-TDOA-ProvideCapabilities*

-- ASN1START

-- TAG-SL-TDOA-PROVIDECAPABILITIES-START

SL-TDOA-ProvideCapabilities ::= SEQUENCE {

}

-- TAG-SL-TDOA-PROVIDECAPABILITIES-STOP

-- ASN1STOP

#### *– SL-TDOA-RequestAssistanceData*

-- ASN1START

-- TAG-SL-TDOA-REQUESTASSISTANCEDATA-START

SL-TDOA-RequestAssistanceData ::= SEQUENCE {

}

-- TAG-SL-TDOA-REQUESTASSISTANCEDATA-STOP

-- ASN1STOP

#### *– SL-TDOA-ProvideAssistanceData*

-- ASN1START

-- TAG-SL-TDOA-PROVIDEASSISTANCEDATA-START

SL-TDOA-ProvideAssistanceData ::= SEQUENCE {

}

-- TAG-SL-TDOA-PROVIDEASSISTANCEDATA-STOP

-- ASN1STOP

#### *– SL-TDOA-RequestLocationInformation*

-- ASN1START

-- TAG-SL-TDOA-REQUESTLOCATIONINFORMATION-START

SL-TDOA-RequestLocationInformation ::= SEQUENCE {

}

-- TAG-SL-TDOA-REQUESTLOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– SL-TDOA-ProvideLocationInformation*

-- ASN1START

-- TAG-SL-TDOA-PROVIDELOCATIONINFORMATION-START

SL-TDOA-ProvideLocationInformation ::= SEQUENCE {

}

-- TAG-SL-TDOA-PROVIDELOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– End of SLPP-PDU-SL-TDOA-Contents*

-- ASN1START

END

-- ASN1STOP

## 6.10 SLPP PDU SL-TOA Contents

#### *– SLPP-PDU-SL-TOA-Contents*

This ASN.1 segment is the start of the SLPP PDU SL-TOA Contents definitions.

-- ASN1START

-- TAG-SLPP-PDU-SL-TOA-CONTENTS-START

SLPP-PDU-SL-TOA-CONTENTS DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- TAG-SLPP-PDU-SL-TOA-CONTENTS-STOP

-- ASN1STOP

#### *– SL-TOA-RequestCapabilities*

-- ASN1START

-- TAG-SL-TOA-REQUESTCAPABILITIES-START

SL-TOA-RequestCapabilities ::= SEQUENCE {

}

-- TAG-SL-TOA-REQUESTCAPABILITIES-STOP

-- ASN1STOP

#### *– SL-TOA-ProvideCapabilities*

-- ASN1START

-- TAG-SL-TOA-PROVIDECAPABILITIES-START

SL-TOA-ProvideCapabilities ::= SEQUENCE {

}

-- TAG-SL-TOA-PROVIDECAPABILITIES-STOP

-- ASN1STOP

#### *– SL-TOA-RequestAssistanceData*

-- ASN1START

-- TAG-SL-TOA-REQUESTASSISTANCEDATA-START

SL-TOA-RequestAssistanceData ::= SEQUENCE {

}

-- TAG-SL-TOA-REQUESTASSISTANCEDATA-STOP

-- ASN1STOP

#### *– SL-TOA-ProvideAssistanceData*

-- ASN1START

-- TAG-SL-TOA-PROVIDEASSISTANCEDATA-START

SL-TOA-ProvideAssistanceData ::= SEQUENCE {

}

-- TAG-SL-TOA-PROVIDEASSISTANCEDATA-STOP

-- ASN1STOP

#### *– SL-TOA-RequestLocationInformation*

-- ASN1START

-- TAG-SL-TOA-REQUESTLOCATIONINFORMATION-START

SL-TOA-RequestLocationInformation ::= SEQUENCE {

}

-- TAG-SL-TOA-REQUESTLOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– SL-TOA-ProvideLocationInformation*

-- ASN1START

-- TAG-SL-TOA-PROVIDELOCATIONINFORMATION-START

SL-TOA-ProvideLocationInformation ::= SEQUENCE {

}

-- TAG-SL-TOA-PROVIDELOCATIONINFORMATION-STOP

-- ASN1STOP

#### *– End of SLPP-PDU-SL-TOA-Contents*

-- ASN1START

END

-- ASN1STOP

Annex <X> (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Change history | | | | | | | |
| Date | Meeting | TDoc | CR | Rev | Cat | Subject/Comment | New version |
| 04/2023 | RAN2#121bis-e | R2-2302739 |  |  |  |  | 0.0.1 |
| 04/2023 | RAN2#121bis-e | R2-2304306 |  |  |  |  | 0.0.2 |
| 05/2023 | RAN2#122 | R2-2305439 |  |  |  |  | 0.0.3 |
| 08/2023 | RAN2#123 | R2-2307663 |  |  |  |  | 0.0.4 |
| 09/2023 | RAN2#123 | R2-2309183 |  |  |  | Endorsed by RAN2 in email discussion [Post123][415] | 0.0.5 |
| 09/2023 | RAN#101 | RP-232009 |  |  |  | To be presented to RAN for information | 1.0.0 |
| 10/2023 | RAN2#123bis | R2-2310222 |  |  |  |  | 1.1.0 |
| 11/2023 | RAN2#124 | R2-231xxxx |  |  |  |  | 1.2.0 |