ETSI TS 124 334 V12.0.0 (2014-10)

TECHNICAL SPECIFICATION

Universal Mobile Telecommunications System (UMTS);
LTE;
Proximity-services (ProSe) User Equipment (UE) to ProSe function protocol aspects;
Stage 3
(3GPP TS 24.334 version 12.0.0 Release 12)

Reference

DTS/TSGC-0124334vc00

Keywords

LTE,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

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

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

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 protocols for Proximity-based Services (ProSe) between:

- the ProSe-enabled UE and the ProSe Function (over the PC3 interface); and
- two ProSe-enabled UEs (over the PC5 interface).

The present document defines the associated procedures for ProSe service authorisation, ProSe direct discovery, EPC-level ProSe discovery, and EPC support for WLAN direct discovery and communication.

The present document also defines the message format, message contents, error handling and system parameters applied by the protocols for ProSe.

The present document is applicable to:

- the ProSe-enabled UE; and
- the ProSe Function.

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 23.303: "Proximity-based services (ProSe); Stage 2".

[3] 3GPP TS 29.344: "Proximity-services (Prose) Function to Home Subscriber Server (HSS) aspects; Stage 3".

[4] 3GPP TS 23.003: "Numbering, addressing and identification".

[5] 3GPP TS 29.345: "Inter-Proximity-services (Prose) Function signalling aspects; Stage 3".

[6] 3GPP TS 33.303: "TS on ProSe Security".

[7] W3C REC-xmlschema-2-20041028: "XML Schema Part 2: Datatypes".

[8] IETF RFC 4122: "A Universally Unique IDentifier (UUID) URN Namespace".

[9] 3GPP TS 24.333: "Proximity-services (ProSe) Management Objects (MO)".

[10] IETF RFC 1035: "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION".

[11] 3GPP TS 24.301: "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3".

[12] 3GPP TS 36.331: "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification".

[13] Wi-Fi Alliance Technical Committee P2P Task Group, "Wi-Fi Peer-to-Peer (P2P) Technical Specification", Version 1.1.

[14] IEEE Std 802.11-2012: "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[15] IETF RFC 4862: "IPv6 Stateless Address Autoconfiguration".

[16] IETF RFC 3927: "Dynamic Configuration of IPv4 Link-Local Addresses".

[17] 3GPP TS 31.102: "Characteristics of the Universal Subscriber Identity Module (USIM) application".

[18] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

Editor's note: Reference to be updated if RFC 2616 is obsoleted by a new IETF RFC and new changes do not affect the HTTP functions described in 3GPP TS 24.334.

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP 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 3GPP TR 21.905 [1].

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.303 [2] apply:

Local PLMN
ProSe-enabled UE

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP 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 3GPP TR 21.905 [1].

FQDN	Fully Qualified Domain Name
MIC	Message Integrity Check
MIME	Multi-Purpose Internet Mail Extensions
ProSe	Proximity-based Services
TTL	Time To Live
UUID	Universally Unique IDentifier

4 General

4.1 Overview

Editor's note: This clause will describe the overall functionalities of the UE to ProSe Function interface based on 3GPP TS 23.303 [2].

5 ProSe service authorisation and authorisation revocation procedure

5.1 Service authorisation and authorisation revocation for ProSe direct discovery and ProSe direct communication

5.1.1 General

The ProSe direct discovery service authorisation determines whether the UE is authorised to use ProSe direct discovery announcing or ProSe direct discovery monitoring or both in a particular PLMN. The service authorisation is transferred between the UE and the ProSe Function over the PC3 interface with the ProSe Direct Services Provisioning Management Object as specified in 3GPP TS 24.333 [9]. The communication security over the PC3 interface is specified in 3GPP TS 33.303 [6].

The UE first requests service authorisation to use ProSe direct discovery from the ProSe Function of the HPLMN. The service authorisation provided by the ProSe Function of the HPLMN contains a list of PLMNs and indicates for each PLMN whether:

- the UE is pre-authorised to use ProSe direct discovery without contacting the ProSe Function of that PLMN; or
- the UE is conditionally authorised to use ProSe direct discovery in that PLMN, in which case it is required to also request service authorisation from the ProSe Function of that PLMN.

The UE then requests service authorisation from the ProSe Function of the local PLMNs or VPLMN in which it is conditionally authorised to use ProSe direct discovery announcing or monitoring. The ProSe Function of each local PLMN or VPLMN contacts the ProSe Function of the UE's HPLMN to obtain the service authorisation from the HPLMN as specified in 3GPP TS 29.345 [5], merges it with its own service authorisation and sends the merged service authorisation to the UE.

The ProSe direct communication service authorisation determines whether the UE is authorised to use ProSe direct communication in a particular PLMN or when not served by E-UTRAN. In this release of the specification, ProSe direct communication is supported only for Public Safety ProSe-enabled UE.

The service authorisation for ProSe direct communication is either:

1) pre-configured in the UE. The pre-configured service authorisation for ProSe direct communication may be stored in the ME, or in the USIM as specified in 3GPP TS 31.102 [17], or in both the ME and the USIM. If both the ME and the USIM contain the same parameters, the values stored in the USIM shall take precedence. The UE shall not use the pre-configured service authorisation if the contents of the USIM indicate that the UE is not authorised to use them (see 3GPP TS 31.102 [17]); or
2) transferred between the UE and the ProSe Function over the PC3 interface with the ProSe Public Safety Direct Services Provisioning Management Object as specified in 3GPP TS 24.333 [9]. The communication security over the PC3 interface is specified in 3GPP TS 33.303 [6].

When using option 2) above, the UE shall:

a) first request authorisation to use ProSe direct communication from the ProSe Function of the HPLMN. The service authorisation provided by the ProSe Function of the HPLMN indicates:
   - whether the UE is is authorised to perform ProSe direct communication when not served by E-UTRAN, and if so, the required radio parameters to be used for ProSe direct communication when not served by E-UTRAN;
   - the ProSe direct communication policy parameters; and
   - the list of PLMNs in which the UE is authorised to use direct communication when served by E-UTRAN, and for each PLMN, it indicates whether:

- the UE is pre-authorised to use ProSe direct communication without contacting the ProSe Function of that PLMN; or
- the UE is conditionally authorised to use ProSe direct communication in that PLMN, in which case it is required to also request service authorisation from the ProSe Function of that PLMN; and

b) then request service authorisation from the ProSe Function of the PLMNs in which it is conditionally authorised to use ProSe direct communication. The ProSe Function of each of these PLMNs contacts the ProSe Function of the UE's HPLMN to obtain the service authorisation from the HPLMN as specified in 3GPP TS 29.345 [5], merges" it with its own service authorisation and sends the merged service authorisation to the UE.

Alternatively, the ProSe direct communication policy parameters mentioned above can be provided by the third party public safety provider application server, using mechanisms that are out of scope of the present specification. If the UE receives the same parameters from the third party public safety provider application server as those which had been previously transferred between the UE and the ProSe Function over the PC3 interface with the ProSe Public Safety Direct Services Provisioning Management Object, the UE shall use the parameters provided by the third party public safety provider application server for ProSe direct communication.

The UE discovers the IP address of the ProSe Functions of the HPLMN, VPLMN and local PLMN as specified in subclause 5.1.2.

After the UE is authorised to use ProSe direct discovery or ProSe direct communication or both, the ProSe Function of the HPLMN shall revoke the service authorisation:

a) when the ProSe Function of the HPLMN is informed the ProSe related subscription data is revoked at the HSS;

b) when the ProSe Function of the HPLMN decides to revoke the authorization for ProSe direct service; or

c) when the ProSe Function of the HPLMN is informed the ProSe Function of the VPLMN or local PLMN decides to revoke the authorization for ProSe direct service.

The ProSe Function of the HPLMN sends the updated authorization for ProSe direct service to the UE, e.g. by sending an OMA push message. The updated authorization for ProSe direct service does not include:

a) the authorization for ProSe direct service (discovery or communication or both) which is to be revoked; and

b) the PLMN ID of the PLMN in which the service authorisation is to be revoked.

NOTE: The ProSe Function of the HPLMN can send the updated authorization for ProSe direct service to the UE immediately or wait for the next time when the UE communicates with the ProSe Function of the HPLMN based on operator's policy; in the latter case, the UE is allowed to use ProSe direct services until the next time that it will communicate with the ProSe Function of the HPLMN.

5.1.2 ProSe Function discovery

The IP address of the ProSe function in the HPLMN may be pre-configured in the UE and in this case, the UE may use the pre-configured IP address. Alternatively, the FQDN of the ProSe Function in the HPLMN may be self-constructed by the UE, i.e. derived from the PLMN ID of the HPLMN. The UE may perform DNS lookup as specified in IETF RFC 1035 [10].

The FQDN of a ProSe Function in a VPLMN or local PLMN may be provided to the UE by the HPLMN during the service authorisation procedure as specified in subclause 5.1.3 and in this case, the UE may use the provided FQDN. Alternatively, the FQDN of the ProSe Function in a VPLMN or local PLMN may be self-constructed by the UE, i.e. derived from the PLMN ID of the VPLMN or local PLMN. The UE may use DNS lookup as specified in IETF RFC 1035 [10].

5.1.3 Service authorisation from ProSe Function of HPLMN

The UE shall initiate the service authorisation procedure to the ProSe Function of the HPLMN:

a) when the UE receives a request from upper layer to perform ProSe direct discovery announcing or monitoring or direct communication and has no valid service authorisation;

b) when the UE is performing ProSe direct discovery announcing or monitoring or direct communication and changes its registered PLMN; or

c) when timer T4005 associated with a valid service authorisation policy expires and the request from upper layer to perform ProSe direct discovery announcing or monitoring or direct communication in the corresponding PLMN is still in place.

NOTE: In order to ensure continuity of ProSe direct discovery service, the UE can request service authorisation from the ProSe Function of a PLMN before the timer T4005 associated with a service authorisation policy in that PLMN expires.

The UE shall obtain the service authorisation from the ProSe Function of the HPLMN over the PC3 interface by requesting the ProSe Direct Services Provisioning Management Object or the ProSe Public Safety Direct Services Provisioning MO as specified in 3GPP TS 24.333 [9]. The UE waits for an implementation dependent time for an answer from the ProSe Function. If the ProSe Function does not respond within that time, the UE may retry the service authorisation procedure. The number of retries performed by the UE is implementation dependent. Unless the UE receives a response from the ProSe function for service authorisation, the UE shall not consider that the request has been authorised.

The ProSe direct discovery service authorisation from the ProSe Function of the HPLMN may include:

a) the PLMNs in which the UE is authorised to perform ProSe direct discovery monitoring, and for each PLMN, it indicates:

    1) a validity timer T4005 indicating for how long the monitoring authorisation policy in that PLMN is valid; and

    2) whether the UE is:

        - pre-authorised to use ProSe direct discovery monitoring in that PLMN without contacting the ProSe Function of that PLMN; or

        - conditionally authorised to use ProSe direct discovery monitoring in that PLMN, in which case it is required to also request service authorisation from the ProSe Function of that PLMN; and

b) the PLMNs in which the UE is authorised to perform ProSe direct discovery announcing , and for each PLMN, it indicates:

    1) a validity timer T4005 indicating for how long the announcing authorisation policy in that PLMN is valid.

    2) the authorised announcing range; and

Editor's note: The description of the authorised announcing range will need to be updated once 3GPP TSG RAN2 TS is available and contains the relevant parameter description.

    3) whether the UE is:

        - pre-authorised to use ProSe direct discovery announcing in that PLMN without contacting the ProSe Function of that PLMN; or

        - conditionally authorised to use ProSe direct discovery announcing in that PLMN, in which case it is required to also request service authorisation from the ProSe Function of that PLMN.

The ProSe direct communication service authorisation from the ProSe Function of the HPLMN may include:

a) whether the UE is is authorised to perform ProSe direct communication when not served by E-UTRAN;

b) the radio parameters to be used for ProSe direct communication when not served by E-UTRAN;

Editor's note: The radio parameters that need to be configured on the UE to perform ProSe direct communication when the UE is not served by E-UTRAN are to be defined by 3GPP TSG RAN WG2.

c) the PLMNs in which the UE is authorised to perform ProSe direct communication when served by E-UTRAN, and for each PLMN, it indicates:

    1) whether the UE is:

- pre-authorised to use ProSe direct direct communication in that PLMN without contacting the ProSe Function of that PLMN; or

- conditionally authorised to use ProSe direct communication in that PLMN, in which case it is required to also request service authorisation from the ProSe Function of that PLMN; and

2) a validity timer T4005 indicating for how long the direct communication authorisation policy in that PLMN is valid; and

d) the ProSe Direct communication policy parameters, consisting of:

1) the ProSe Layer-2 Group ID;

2) the ProSe Group IP multicast address;

3) whether the UE should use IPv4 or IPv6 for that group;

4) an IPv4 address to be used by the UE as a source address in case IPv4 is used; and

5) group-related security contents.

The UE shall start validity timer(s) T4005 with the values included in this service authorisation. The UE shall consider that an authorisation policy is valid in the associated PLMN until the corresponding validity timer T4005 expires.

5.1.4 Service authorisation from ProSe Function of VPLMN or local PLMN

The UE may perform ProSe direct discovery monitoring in a local PLMN or VPLMN without requesting authorisation from the ProSe Function of that PLMN if the UE has been pre-authorised for ProSe direct discovery monitoring in that PLMN by the ProSe Function of the HPLMN.

The UE may perform ProSe direct discovery announcing in a VPLMN without requesting authorisation from the ProSe Function of that PLMN if the UE has been pre-authorised for ProSe direct discovery announcing in that PLMN by the ProSe Function of the HPLMN.

NOTE 1: ProSe direct discovery announcing can only be performed in the bands indicated by the registered PLMN so it is not applicable to local PLMNs.

The UE may perform ProSe direct communication in a VPLMN without requesting authorisation from the ProSe Function of that PLMN if the UE has been pre-authorised for ProSe direct communication in that PLMN by the ProSe Function of the HPLMN.

NOTE 2: ProSe direct communication can only be performed in the bands indicated by the registered PLMN so it is not applicable to local PLMNs.

The UE shall request service authorisation for ProSe direct discovery or ProSe direct communication from the ProSe Function of the PLMNs in which the UE has been conditionally authorised for ProSe direct discovery announcing or monitoring or ProSe direct communication by the ProSe Function of the HPLMN. The UE shall obtain service authorisation over the PC3 interface by requesting the ProSe Direct Services Provisioning Management Object or the ProSe Public Safety Direct Services Provisioning MO as specified in 3GPP TS 24.333 [9]. The UE waits for an implementation dependent time for an answer from the ProSe Function. If the ProSe Function does not respond within that time, the UE may retry the service authorisation procedure. The number of retries performed by the UE is implementation dependent. Unless the UE receives a response from the ProSe function for service authorisation, the UE shall not consider that the request has been authorised.

NOTE 3: The ProSe Function of the VPLMN or local PLMN requests service authorisation information from the ProSe Function in the UE's HPLMN and merges it with its own service authorisation information, then provides the merged service authorisation to the UE.

The UE shall start validity timer(s) T4005 with the values included in this service authorisation. The UE shall consider that an authorisation policy is valid in the associated PLMN until the corresponding validity timer T4005 expires. The service authorisation is applicable only in the corresponding PLMN.

6 ProSe direct discovery

6.1 Overview

This clause describes the PC3 Control Protocol procedures between the UE and the ProSe Function for ProSe direct discovery announcing and monitoring. It also describes the ProSe Protocol procedures at the UE for ProSe direct discovery of other ProSe-enabled UEs over the PC5 interface.

6.1.1 Transport protocol for PC3 Control Protocol messages for ProSe direct discovery

The UE and ProSe Function shall use HTTP 1.1 as specified in IETF RFC 2616 [18] as the transport protocol for ProSe messages over the PC3 interface. The ProSe messages described here shall be included in the body of either an HTTP request message or an HTTP response message. The following rules apply:

- The UE initiates ProSe transactions with an HTTP request message containing the PC3 request(s);
- The ProSe Function responds to the requests with an HTTP response message containing the PC3 response(s) for the PC3 request(s); and
- HTTP POST methods are used for PC3 direct discovery procedures.

6.2 Procedures

6.2.1 Types of ProSe direct discovery procedures

The following PC3 Control Protocol procedures are defined:

- announce request;
- monitor request; and
- match report.

All procedures are initiated from the UE to the ProSe Function in the UE's HPLMN.

In the following descriptions of PC3 Control Protocol procedures, the terms "request" and "response" refer to the corresponding PC3 Control Protocol messages, not to the HTTP request or response. The following procedure descriptions use a single PC3 Control Protocol message for illustration purposes.

NOTE: A single HTTP request message can contain multiple PC3 Control Protocol requests and a single HTTP response message can contain multiple PC3 Control Protocol responses.

6.2.2 Announce request procedure

6.2.2.1 General

The purpose of the announce request procedure is for the UE to obtain a ProSe Application Code to be announced over the PC5 interface, upon a request for announcing from upper layers as defined in 3GPP TS 23.303 [2].

The UE shall be authorised for ProSe direct discovery announcing in the registered PLMN based on the service authorisation procedure as specified in clause 5, before initiating the announce request procedure.

The UE includes the ProSe Application Code obtained as a result of a successful announce request procedure in a PC5_DISCOVERY message and passes the PC5_DISCOVERY message to the lower layers for transmission over the PC5 interface.

6.2.2.2 Announce request procedure initiation

Before initiating the announce request procedure, the UE is configured with the data structure of the ProSe Application IDs appropriate for its HPLMN. This step is performed using mechanisms out of scope of 3GPP.

If the UE is authorised to perform ProSe direct discovery announcing in the registered PLMN, it shall initiate an announce request procedure:

a) when the UE is triggered by an upper layer application to announce a ProSe Application ID and the UE has no valid corresponding ProSe Application Code for that upper layer application;

b) when the validity timer T4000 assigned by the ProSe Function to a ProSe Application Code has expired and the request from upper layers to announce the ProSe Application ID corresponding to that ProSe Application Code is still in place; or

c) when the UE selects a new PLMN while announcing a ProSe Application Code.

NOTE 1: To ensure service continuity if the UE needs to keep announcing a ProSe Application Code corresponding to the same ProSe Application ID, the UE can initiate the announce request procedure before the TTL timer T4000 assigned by the ProSe Function for a Prose Application Code expires.

The UE initiates the announce request procedure by sending a DISCOVERY_REQUEST message with a new transaction ID, the ProSe Application ID set to the ProSe Application ID received from upper layers, the command set to "announce", the UE identity set to the UE's IMSI, and the Application Identity set to the Application Identity of the upper layer application that requested the announcing.

NOTE 2: A UE can include one or multiple transactions in one DISCOVERY_REQUEST message for different ProSe Application IDs, and receive corresponding <response-announce> element or <response-reject> element in a DISCOVERY_RESPONSE message for each respective transaction. In the following description of the announce request procedure, only one transaction is included.

Figure 6.2.2.2.1 illustrates the interaction of the UE and the ProSe Function in the announce request procedure.

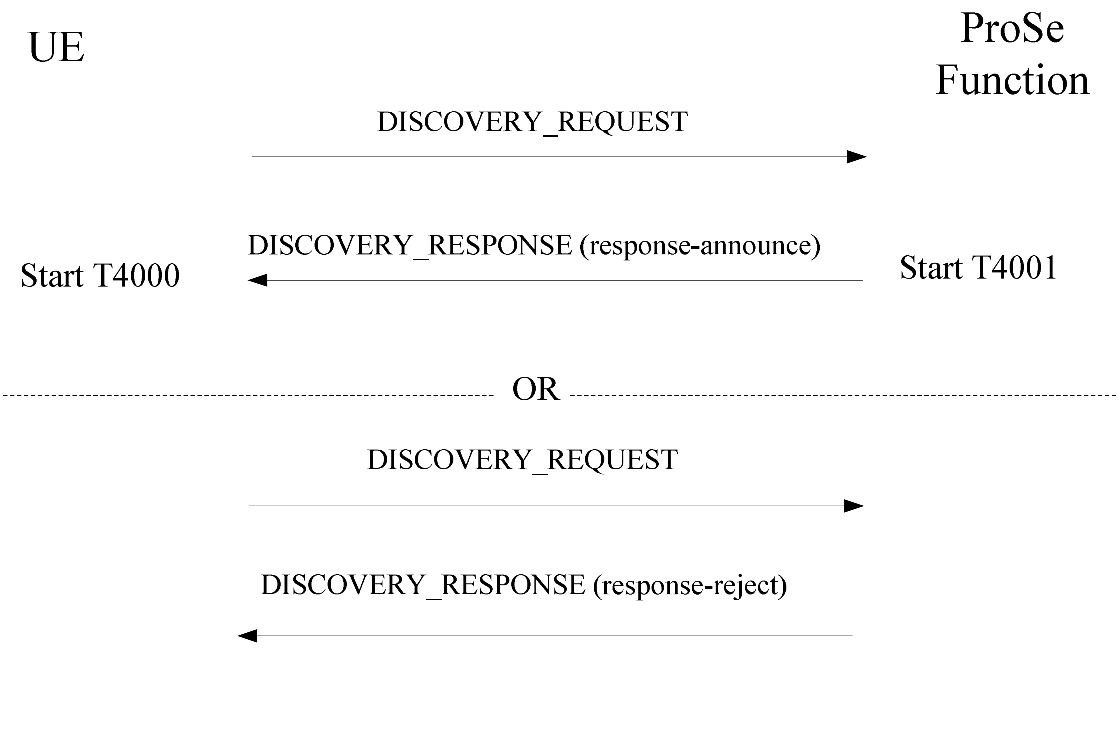
UE
ProSe Function
DISCOVERY_REQUEST
DISCOVERY_RESPONSE (response-announce)
Start T4000
Start T4001
OR
DISCOVERY_REQUEST
DISCOVERY_RESPONSE (response-reject)

Figure 6.2.2.2.1: Announce request procedure

6.2.2.3 Announce request procedure accepted by the ProSe Function

Upon receiving a DISCOVERY_REQUEST message, the ProSe Function shall check that the application corresponding to the Application Identity contained in the DISCOVERY_REQUEST message is authorised for ProSe direct discovery announcing in the registered PLMN. If the application is authorised for ProSe direct discovery announcing in the registered PLMN, the ProSe Function shall check whether there is an existing context for the UE associated with the requested ProSe Application ID.

If there is no associated UE context, the ProSe Function checks with the HSS whether the UE is authorised for ProSe direct discovery announcing as described in 3GPP TS 29.344 [3]. If the check indicates that the UE is authorised, the ProSe Function shall allocate the corresponding ProSe Application Code and a value for validity timer T4000.

If the requested ProSe Application ID is country-specific or global as described in subclause 24.2 of 3GPP TS 23.003 [4], the ProSe Function shall allocate a corresponding ProSe Application Code according to subclause 24.3 of 3GPP TS 23.003 [4]. The temporary identity part of the ProSe Application Code is taken from the data structure corresponding to the country-specific or global ProSe Application ID namespace according to subclause 24.3 of 3GPP TS 23.003 [4]. The ProSe Function shall use the MCC and MNC of the PLMN ID of this ProSe Function for the PLMN ID part of the ProSe Application Code.

After the ProSe Application Code is allocated, the ProSe Function then associates this ProSe Application Code and its validity timer T4000 with a new context for the UE that contains the UE's subscription parameters obtained from the HSS, and starts timer T4001. The HSS also provides to the ProSe Function the PLMN ID of the PLMN in which the UE is currently registered.

NOTE: For a given ProSe Application Code, T4001 is longer than T4000 to ensure service continuity.

Editor's note: The gap value between T4001 and T4000 is FFS.

If there is an existing context for the UE associated with the requested ProSe Application ID, the ProSe function shall either update the context with a new validity timer T4000 and restart timer T4001, or allocate a new ProSe Application Code for the requested ProSe Application ID with a new validity timer T4000.

If a new context was created or an existing context was updated and the UE is currently roaming, the ProSe Function checks with the VPLMN's ProSe function whether the UE is authorised for ProSe direct discovery announcing as described in 3GPP TS 29.345 [5].

If the check indicates that the UE is authorised then the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-announce> element with the transaction ID set to the value of the transaction ID received in the DISCOVERY_REQUEST message from the UE, the ProSe Application Code set to the ProSe Application Code allocated by the ProSe Function for the ProSe Application ID received in the DISCOVERY_REQUEST message from the UE, Validity Timer T4000 set to the T4000 timer value assigned by the ProSe Function to the ProSe Application Code, the Discovery Type set to "Open Discovery" and the Discovery Key set to a value provided by the ProSe Function.

If the UE does not initiate a new announce request procedure before timer T4001 expires, the ProSe Function shall remove the UE's context which associates the UE with the corresponding ProSe Application ID.

6.2.2.4 Announce request procedure completion by the UE

Upon receipt of the DISCOVERY_RESPONSE message, if the transaction ID contained in the <response-announce> element matches the value sent by the UE in a DISCOVERY_REQUEST message with the command set to "announce", the UE shall start performing direct discovery announcing as described below. Otherwise the UE shall discard the DISCOVERY_RESPONSE message.

The UE requests the ProSe direct discovery parameters from the lower layers (see 3GPP TS 36.331 [12]). If the lower layers indicate that ProSe direct discovery is not supported by the network, the UE shall not perform direct discovery announcing. Else if the lower layers indicate that ProSe direct discovery is supported by the network:

- if the lower layers indicate that the UE is not required to send a discovery indication to the eNodeB, the UE shall request the time parameter for the next discovery transmission opportunity from lower layers;
- if the lower layers indicate that the UE is required to send a discovery indication to the eNodeB and the UE is in EMM-IDLE mode, the UE shall perform a Service Request procedure to send a discovery indication for ProSe direct discovery announcing as specified in 3GPP TS 24.301 [11] to transfer to EMM-CONNECTED mode.

Upon successful completion of the Service Request procedure, the UE shall trigger the lower layers to send the discovery indication to the eNodeB as specified in 3GPP TS 36.331 [12]. Upon indication from the lower layers that the discovery indication has been sent successfully to the eNodeB, the UE shall request the time parameter for the next discovery transmission opportunity from lower layers; or

- if the lower layers indicate that the UE is required to send a discovery indication to the eNodeB and the UE is in EMM-CONNECTED mode, the UE shall trigger the lower layers to send the discovery indication to the eNodeB as specified in 3GPP TS 36.331 [12]. Upon indication from lower layers that the discovery indication has been sent successfully to the eNodeB, the UE shall request the time parameter for the next discovery transmission opportunity from lower layers;

If the time parameter obtained from lower layers is valid, the UE uses this parameter to compute the MIC field for the PC5_DISCOVERY message as described in 3GPP TS 33.303 [6].

The UE shall use the ProSe Application Code received in the DISCOVERY_RESPONSE message, along with the Discovery Key, in order to construct a PC5_DISCOVERY message, according to the format defined in subclause 11.2.5.

The UE then passes the PC5_DISCOVERY message to the lower layers for transmission if:

- the UE is currently authorised to perform direct discovery announcing in the registered PLMN;
- the validity timer T4000 for the allocated ProSe Application Code has not expired; and
- a request from upper layers to announce the ProSe Application ID associated with both the ProSe Application Code and the authorised Application Identity is still in place.

6.2.2.5 Announce request procedure not accepted by the ProSe Function

If the DISCOVERY_REQUEST message cannot be accepted by the ProSe Function, the ProSe Function sends a DISCOVERY_RESPONSE message containing a <response-reject> element to the UE including an appropriate PC3 Control Protocol cause value.

If the application corresponding to the Application Identity contained in the DISCOVERY_REQUEST message is not authorised for ProSe direct discovery announcing in the registered PLMN, the ProSe Function shall send the DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #1 "Invalid application".

If the UE is not authorised for ProSe direct discovery announcing, the ProSe Function shall send the DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #3 "UE authorisation failure".

If the UE requests a country-specific ProSe Application ID for a country that does not correspond to the country of its HPLMN, and the ProSe Function has not authorized the UE to announce in that country, the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #8 "Scope Violation in Prose Application ID".

If the UE requests a country-specific ProSe Application ID for a country that does not correspond to the country of its HPLMN, and the ProSe Function has no agreement to access the country-wide ProSe Application ID database of that country, the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #8 "Scope Violation in Prose Application ID".

6.2.2.6 Abnormal cases

6.2.2.6.1 Abnormal cases in the UE

The following abnormal cases can be identified:

a) Indication from the transport layer of transmission failure of DISCOVERY_REQUEST message (e.g. after TCP retransmission timeout)

   The UE shall close the existing secure connection to the ProSe Function, establish a new secure connection and then restart the announce request procedure.

b) No response from the ProSe Function after the DISCOVERY_REQUEST message has been successfully delivered (e.g. TCP ACK has been received for the DISCOVERY_REQUEST message)

The UE shall retransmit the DISCOVERY_REQUEST message.

Editor's note: the timer to trigger retransmission and the maximum number of allowed retransmission are FFS.

c) Indication from upper layers that the request to announce the ProSe Application ID is no longer in place after sending the DISCOVERY_REQUEST message, but before the announce request procedure is completed

The UE shall acknowledge the DISCOVERY_RESPONSE message received from the ProSe Function but discard its contents and then abort the procedure.

d) Change of PLMN

If a PLMN change occurs before the announce request procedure is completed, the procedure shall be aborted. If the UE is authorized to announce in the new PLMN, the procedure shall be restarted once the UE is registered on the new PLMN.

6.2.2.6.2 Abnormal cases in the ProSe Function

The following abnormal cases can be identified:

a) Indication from the lower layer of transmission failure of DISCOVERY_RESPONSE message

After receiving an indication from lower layer that the DISCOVERY_RESPONSE message has not been successfully acknowledged (e.g. TCP ACK is not received), the ProSe Function shall abort the procedure, and stop any associated timer(s) T4001, if running.

6.2.3 Monitor request procedure

6.2.3.1 General

The purpose of the monitor request procedure is to allow a UE to receive and process PC5_DISCOVERY messages upon a request for monitoring from upper layers as defined in 3GPP TS 23.303 [2].

The UE shall only initiate the monitor request procedure if it has been authorised for ProSe direct discovery monitoring at least in one PLMN based on the service authorisation procedure.

As a result of the monitor request procedure completing successfully, the UE obtains one or more Discovery Filters, along with a TTL (Time-To-Live) timer T4002 for each Discovery Filter indicating the time during which the filter is valid.

6.2.3.2 Monitor request procedure Initiation

Before initiating the monitor request procedure, the UE is configured with the data structure of the ProSe Application IDs it wants to monitor. This step is performed using mechanisms that are out of scope of 3GPP.

If the UE is authorised to perform ProSe direct discovery monitoring in at least one PLMN, it shall initiate a monitor request procedure:

a) when the UE is triggered by an upper layer application to perform ProSe direct discovery monitoring corresponding to a ProSe Application ID and the UE has no valid Discovery Filters corresponding to the requested ProSe Application ID for that upper layer application; or

b) when the TTL timer T4002 assigned by the ProSe Function to a Discovery Filter has expired and the request from upper layers to monitor that ProSe Application ID is still in place.

NOTE 1: To ensure service continuity if the UE needs to keep monitoring the same Discovery Filter, the UE can initiate the monitor request procedure before the TTL timer T4002 assigned by the ProSe Function for a Discovery Filter expires.

The UE initiates the monitor request procedure by sending a DISCOVERY_REQUEST message with a new transaction ID, the ProSe Application ID set to the ProSe Application ID received from upper layers, the command set to "monitor", the UE identity set to the UE's IMSI, and the Application Identity set to the Application Identity of the upper layer application that requested the monitoring.

NOTE 2: A UE can include one or multiple transactions in one DISCOVERY_REQUEST message for one or more ProSe Application IDs, and receive corresponding <response-monitor> element or <response-reject> element in the DISCOVERY_RESPONSE message for each respective transaction. In the following description of the monitor request procedure, only one transaction is included.

Figure 6.2.3.2.1 illustrates the interaction between the UE and the ProSe Function in the monitor request procedure.

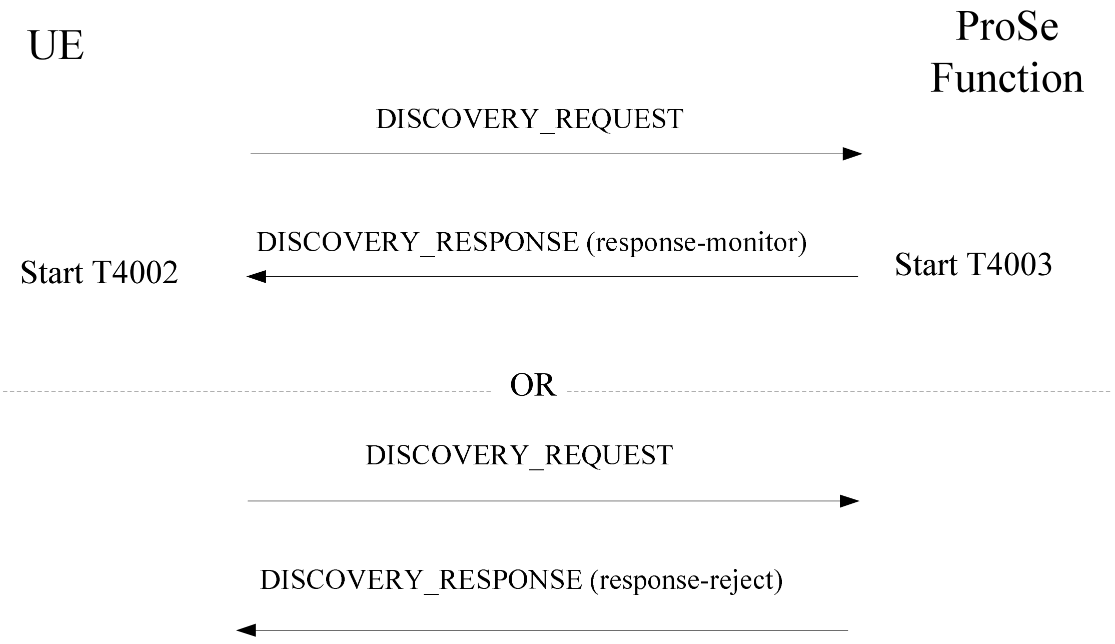

Figure 6.2.3.2.1: Monitor request procedure

6.2.3.3 Monitor request procedure accepted by the ProSe Function

Upon receiving a DISCOVERY_REQUEST message, the ProSe Function shall check that the application corresponding to the Application Identity contained in the DISCOVERY_REQUEST message is authorised for ProSe direct discovery monitoring in the registered PLMN. If the application is authorised for ProSe direct discovery monitoring in the registered PLMN, the ProSe Function shall check whether there is an existing context for the UE associated with the requested ProSe Application ID.

If there is no associated UE context, the ProSe Function checks with the HSS whether the UE is authorised for ProSe direct discovery monitoring as described in 3GPP TS 29.344 [3]. If the check indicates that the UE is authorised, the ProSe Function shall allocate one or more Discovery Filters to the requested ProSe Application ID. Each Discovery Filter consists of a ProSe Application Code, one or more ProSe Application Masks, and a TTL timer T4002.

If the requested ProSe Application ID is country-specific or global or PLMN-specific as defined respectively in subclause 24.2 of 3GPP TS 23.003 [4], the ProSe Function shall allocate the Discovery Filter which contains ProSe Application Code and ProSe Application Mask(s) in the corresponding scope. If the ProSe Application ID is country-specific or global, the ProSe Application Mask(s) enclosed in the Discovery Filter hides the PLMN ID part correspondingly and the temporary identity part is taken from the data structure corresponding to the global or country-wide ProSe Application ID namespace, as specified in subclause 24.3 of 3GPP TS 23.003 [4]. If the requested ProSe Application ID is PLMN-specific, the ProSe Function shall allocate one or more PLMN-specific Discovery Filters in the format defined in subclause 24.3 of 3GPP TS 23.003 [4], in which the PLMN ID portion of the mask shall be used to match the PLMN ID of that specific PLMN.

After the Discovery Filter(s) are allocated, the ProSe Function then associates the Discovery Filters with a new context for this UE that contains the subscription parameters for this UE obtained from the HSS, and starts timer T4003 assigned for each Discovery Filter. The HSS also provides to the ProSe Function the PLMN ID of the PLMN in which the UE is currently registered.

NOTE: For a given Discovery Filter, T4003 is longer than T4002 to ensure service continuity.

Editor's note: The gap value between T4003 and T4002 is FFS.

If there is an existing context for the UE associated with the requested ProSe Application ID, the ProSe function shall update the context for the same Discovery Filter(s) with new TTL timer(s) T4002 and restart timer T4003 for each filter.

The ProSe Function shall also check the PLMN ID in the ProSe Application ID. If the PLMN ID in the ProSe Application ID is not the same as that of the PLMN to which the ProSe Function belongs, then the ProSe Function executes the procedures defined in 3GPP TS 29.345 [5].

Then the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-monitor> element with the transaction ID set to the value of the transaction ID received in the DISCOVERY_REQUEST message from the UE, and one or more Discover Filters allocated by the ProSe Function(s) for the ProSe Application ID received in the DISCOVERY_REQUEST message from the UE.

If the UE does not initiate a new monitor request procedure before timer T4003 expires, the ProSe Function shall remove the UE's association with the corresponding Discovery Filter.

6.2.3.4 Monitor request procedure completion by the UE

Upon receipt of the DISCOVERY_RESPONSE message, if the transaction ID contained in the <response-monitor> element matches the value sent by the UE in a DISCOVERY_REQUEST message with the command set to "monitor", the UE may start monitoring for discovery messages received over the PC5 interface as described below. Otherwise the UE shall discard the DISCOVERY_RESPONSE message.

The UE shall start TTL timer T4002 for each Discovery Filter assigned by the ProSe Function.

For a ProSe Application ID requested by the monitoring UE, the ProSe Function may have assigned one or more Discovery Filters. The UE should apply all assigned Discovery Filters to its monitoring operation. Using these filters may result in a match event. In case of such an event, the UE shall consider that the ProSe Application ID it seeks to monitor has been discovered. A match event is defined as follows:

There is a match event when, for any of the masks in a Discovery Filter, the output of a bitwise AND operation between the ProSe Application Code contained in the PC5_DISCOVERY message and this mask matches the output of a bitwise AND operation between the mask and the ProSe Application Code in the same filter.

The UE may instruct the lower layers to start monitoring if all of the following conditions are met:

- the UE is currently authorized to perform monitoring in at least one PLMN;
- the UE has obtained at least one Discovery Filter and their respective TTL timer T4002(s) have not expired; and
- a request from upper layers to monitor for the ProSe Application ID associated with an authorised Application Identity is still in place.

During the monitoring operation, the UE receives all PC5_DISCOVERY messages and associated time parameters from the lower layers.

During the monitoring operation, if one of above condition is not met, the UE may instruct the lower layers to stop monitoring.

6.2.3.5 Monitor request procedure not accepted by the ProSe Function

If the DISCOVERY_REQUEST message is not accepted by the ProSe Function, the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-reject> element to the UE including an appropriate PC3 Control Protocol cause value.

If the application corresponding to the Application Identity contained in the DISCOVERY_REQUEST message is not authorised for ProSe direct discovery monitoring in the registered PLMN, the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #1 "Invalid application".

If the ProSe Application ID contained in the DISCOVERY_REQUEST message is unknown to the ProSe Function, the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #2 "Unknown ProSe Application ID".

If the UE is not authorised for ProSe direct discovery announcing, the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #3 "UE authorisation failure".

If the UE requests a country-specific ProSe Application ID for a country that does not correspond to the country of its HPLMN, and the ProSe Function has not authorized the UE to monitor in that country, it shall send a DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #8 "Scope Violation in Prose Application ID".

If the UE requests a country-specific ProSe Application ID for a country that does not correspond to the country of its HPLMN, and the ProSe Function has no agreement to access the country-specific ProSe Application ID database of that country, the ProSe Function shall send a DISCOVERY_RESPONSE message containing a <response-reject> element with PC3 Control Protocol cause value #8 "Scope Violation in Prose Application ID".

6.2.3.6 Abnormal cases

6.2.3.6.1 Abnormal cases in the UE

The following abnormal cases can be identified:

a) Indication from the transport layer of transmission failure of DISCOVERY_REQUEST message (e.g. after TCP retransmission timeout)

   The UE shall close the existing secure connection to the ProSe Function, establish a new secure connection and then restart the monitor request procedure.

b) No response from the ProSe Function after the DISCOVERY_REQUEST message has been successfully delivered (e.g. TCP ACK has been received for the DISCOVERY_REQUEST message)

   The UE shall retransmit the DISCOVERY_REQUEST message.

Editor's note: the timer to trigger retransmission and the maximum number of allowed retransmission are FFS.

c) Indication from upper layers that the request to monitor the ProSe Application ID is no longer in place after sending the DISCOVERY_REQUEST message, but before the monitor request procedure is completed

   The UE shall acknowledge the DISCOVERY_RESPONSE message received from the ProSe Function but discard its contents and then abort the procedure.

d) Change of PLMN

   If a PLMN change occurs before the monitor request procedure is completed, the procedure shall be aborted. If the UE is authorized to monitor in the new PLMN, the procedures shall be restarted once the UE is registered on the new PLMN.

6.2.3.6.2 Abnormal cases in the ProSe Function

The following abnormal cases can be identified:

a) Indication from the lower layer of transmission failure of DISCOVERY_RESPONSE message

After receiving an indication from lower layer that the DISCOVEY_RESPONSE message has not been successfully acknowledged, the ProSe Function shall abort the procedure, and stop any associated timer(s) T4003, if running.

6.2.4 Match report procedure

6.2.4.1 General

The purpose of the Match report procedure is to allow a UE to send a ProSe Application Code that was matched during the monitoring operation and receive the corresponding ProSe Application ID, if there is no such a mapping stored locally.

The UE shall only initiate the match report procedure if it has been authorised for ProSe direct discovery monitoring at least in one PLMN based on the service authorisation procedure.

As a result of the match report procedure completing successfully, the UE obtains a ProSe Application ID and potentially other information, which the UE may store locally and pass to the upper layers.

6.2.4.2 Match report procedure initiation

The UE shall meet the following pre-conditions before initiating this procedure:

- a request from upper layers to monitor for the ProSe Application ID, which resulted in the matched ProSe Application Code, is still in place;
- the lower layers have provided a "Monitored PLMN ID" value, and time parameter information, along with the discovery message containing a ProSe Application Code; and
- the TTL timer T4002 associated with the Discovery Filter, which resulted in a match event of the ProSe Application Code, has not expired.

If the UE is authorised to perform ProSe direct discovery monitoring in the registered PLMN, it should initiate a match report procedure:

a) when there is a match event of one of the ProSe Application Codes received from the lower layers, and the UE does not have a corresponding ProSe Application ID already locally stored; or

b) when the UE has a locally stored mapping for the ProSe Application Code that resulted in a match event, but the validity timer T4004 of the ProSe Application Code has expired.

The UE initiates the match report procedure by sending a MATCH_REPORT message with a new transaction ID and shall set the message contents as follows:

- the UE shall set the ProSe Application Code to the ProSe Application Code for which there was a match event;
- the UE shall set the UE identity to the UE's IMSI;
- the UE shall set the Time Parameter to the time parameter provided by the lower layers for the PC5_DISCOVERY message that contained the ProSe Application Code for which there was a match event;
- the UE shall set the MIC to the MIC of the PC5_DISCOVERY message that contained the ProSe Application Code for which there was a match event;
- the UE shall set the Monitored PLMN ID to the PLMN ID of the PLMN where the PC5_DISCOVERY message was received, as provided by the lower layers;
- if the UE was roaming when the match event occurred, the UE shall set the VPLMN ID to the PLMN ID of the PLMN in which the UE was registered when the match event occurred; and
- the UE shall set the Metadata Flag to indicate whether or not it wishes to receive metadata information associated with the ProSe Application ID in the MATCH_REPORT_ACK message from the ProSe Function.

NOTE 1: A UE can include one or multiple transactions in one MATCH_REPORT message for different ProSe Application Codes, and receive corresponding <match-ack> element or <match-reject> element in the MATCH_REPORT_ACK message for each respective transaction. In the following description of match report procedure, only one transaction is included.

NOTE 2: The value of the Metadata Flag is determined through an indication from upper layers in the original request to monitor for a ProSe Application ID.

Figure 6.2.4.2.1 illustrates the interaction between the UE and the ProSe Function in the match report procedure.

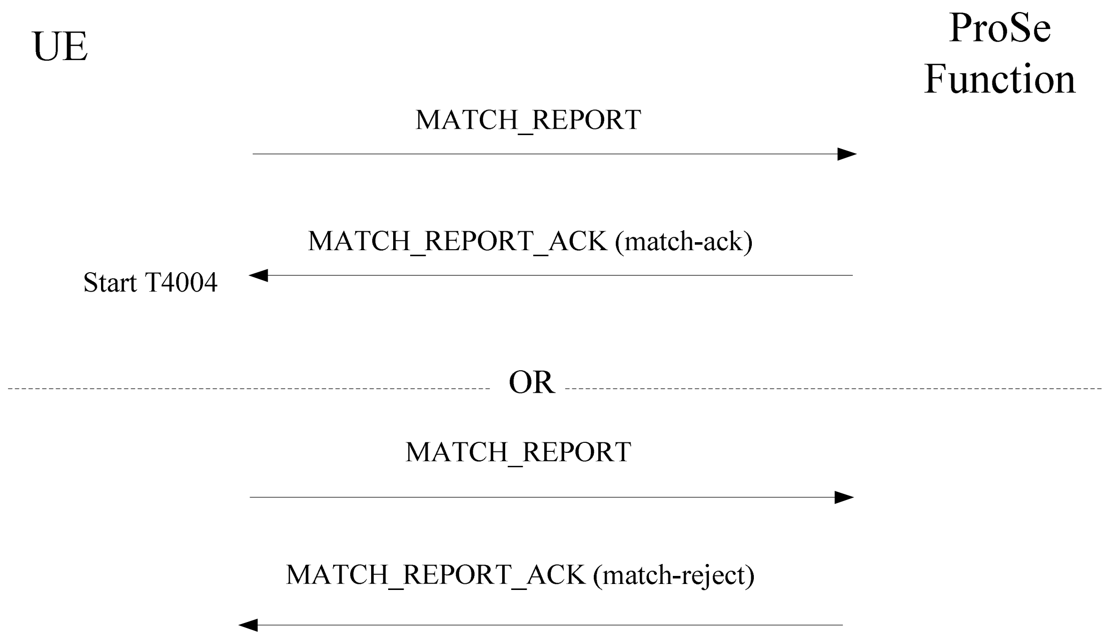

Figure 6.2.4.2.1: Match report procedure

6.2.4.3 Match report procedure accepted by the ProSe Function

Upon receiving a MATCH_REPORT message, the ProSe Function shall check whether there is an existing context for the UE identified by its IMSI. If there is no associated UE context, the ProSe Function checks with the HSS whether the UE is authorised for ProSe direct discovery monitoring as described in 3GPP TS 29.344 [3]. If the check indicates that the UE is authorised, the ProSe Function shall check whether the received ProSe Application Code is authorised to be transmitted on the monitored PLMN indicated in the Monitored PLMN ID in the received message.

The ProSe Function shall also check the PLMN ID in the ProSe Application Code received from the UE.If the PLMN ID in the ProSe Application Code is not the same of that of the PLMN to which the ProSe Function belongs, the ProSe Function executes the procedures defined in 3GPP TS 29.345 [5].

If the ProSe Application Code is PLMN-specific, the ProSe Function shall verify if the monitored PLMN indicated in the received MATCH_REPORT message is the same as the PLMN of the ProSe Function. If so, the ProSe Function shall map the ProSe Application Code to the corresponding ProSe Application ID from the PLMN-specific database. If the ProSe Application Code is country-specific, as specified in subclause 24.3 of 3GPP TS 23.003 [4], the ProSe Function shall check whether the MCC of the PLMN ID part of the ProSe Application Code corresponds to the country of the ProSe Function. If so, the ProSe Function shall map the ProSe Application Code to the corresponding ProSe Application ID from the country-specific database. If the ProSe Application Code is global as specified in subclause 24.3 of 3GPP TS 23.003 [4], the ProSe Function shall map the ProSe Application Code to the corresponding ProSe Application ID from the global database.

The ProSe Function shall analyse the ProSe Application Code received from the UE and determine the validity of the ProSe Application Code.

NOTE: This might require the ProSe Function to execute procedures defined in 3GPP TS 29.345 [5].

The ProSe Function shall check if the MIC value and its corresponding time parameter are valid, as defined in 3GPP TS 33.303 [6].

If the outcome of above processing is successful, the ProSe Function shall send a MATCH_REPORT_ACK message containing a <match-ack> element with the transaction ID set to the value of the transaction ID received in the MATCH_REPORT message from the UE, the ProSe Application ID set to the ProSe Application ID provided by the ProSe Function and corresponding to the ProSe Application Code contained in the MATCH_REPORT message, the Validity Timer T4004 set to indicate for how long this ProSe Application Code is valid. If the UE has set the Metadata Flag to indicate that it wishes to receive metadata information associated with the ProSe Application ID, the ProSe Function shall set the Metadata to the metadata information associated with the ProSe Application Code received in the MATCH_REPORT message.

6.2.4.4 Match report procedure completion by the UE

Upon receipt of the MATCH_REPORT_ACK message, if the transaction ID contained in the <match-ack> element matches the value sent by the UE in a MATCH_REPORT message, the UE may store the mapping between the ProSe Application Code and ProSe Application ID locally, start timer T4004, and may inform the upper layers of this match of the ProSe Application ID. Otherwise the UE shall discard the MATCH_REPORT_ACK message.

NOTE 1: It is an implementation specific choice whether the UE informs the upper layers every time a ProSe Application ID triggers a match event, or only the first time this match occurs.

NOTE 2: The UE may also inform the upper layers if a ProSe Application ID is no longer matched, because the validity timer T4004 of the corresponding ProSe Application Code expires.

6.2.4.5 Match report procedure not accepted by the ProSe Function

If the MATCH_REPORT message is not accepted by the ProSe Function, the ProSe Function sends a MATCH_REPORT_ACK message with a <match-reject> element to the UE including an appropriate PC3 Control Protocol cause value.

If the ProSe Application Code contained in the MATCH_REPORT message is unknown by the ProSe Function, the ProSe Function shall send the MATCH_REPORT_ACK message with a <match-reject> element with PC3 Control Protocol cause value #4 "Unknown ProSe Application Code".

If the check of the MIC contained in the MATCH_REPORT message fails, the ProSe Function shall send the MATCH_REPORT_ACK message with a <match-reject> element with PC3 Control Protocol cause value #5 "Invalid MIC".

If the UE is not authorised for ProSe direct discovery monitoring, the ProSe Function shall send the MATCH_REPORT_ACK message with a <match-reject> element with PC3 Control Protocol cause value #3 "UE authorisation failure".

6.2.4.6 Abnormal cases

6.2.4.6.1 Abnormal cases in the UE

The following abnormal cases can be identified:

a) Indication from the transport layer of transmission failure of MATCH_REPORT message (e.g. after TCP retransmission timeout)

   The UE shall close the existing secure connection to the ProSe Function, establish a new secure connection and then restart the match report procedure.

b) No response from the ProSe Function after the MATCH_REPORT message has been successfully delivered (e.g. TCP ACK has been received for the MATCH_REPORT message)

The UE shall retransmit the MATCH_REPORT message.

Editor's note: the timer to trigger retransmission and the maximum number of allowed retransmission are FFS.

c) Change of PLMN

If a PLMN change occurs before the match report procedure is completed, the procedure shall be aborted.

7 EPC-level ProSe discovery

7.1 Overview

This clause describes the PC3 Control Protocol procedures between the UE and the ProSe Function for EPC-level ProSe discovery.

7.1.1 Transport protocol for PC3 Control Protocol messages for EPC-level ProSe discovery

The UE and the ProSe Function shall use HTTP 1.1 as specified in IETF RFC 2616 [18] as the transport protocol for EPC-level ProSe discovery messages over the PC3 interface. The ProSe messages described here shall be included in the body of either an HTTP request message or an HTTP response message.

7.1.2 Handling of UE-initiated procedures

The following rules apply for UE-initiated procedures:

- The UE initiates ProSe transactions with an HTTP request message containing the PC3 request(s); and
- The ProSe Function responds to the requests with an HTTP response message containing the PC3 response(s) for the PC3 request(s).

7.1.3 Handling of network-initiated procedures

The network-initiated messages for EPC ProSe-level discovery over the PC3 interface shall be contained in an HTTP response.

Editor's note: Whether SMS push, or HTTP long polling, or another mechanism is used to trigger network-initiated transactions with HTTP transport protocol is FFS.

7.2 Procedures

7.2.1 Types of EPC-level ProSe discovery procedures

The following PC3 Control Protocol procedures are defined:

- UE registration;
- application registration;
- proximity request;
- proximity request validation;
- proximity alert;
- UE deregistration; and
- proximity request cancellation.

EPC support for WLAN direct discovery and communication may be requested as part of the EPC-level ProSe discovery procedure.

7.2.2 UE registration procedure

7.2.2.1 General

The purpose of the UE registration procedure is for the UE to register with the ProSe Function to obtain EPC-level ProSe discovery services as defined in 3GPP TS 23.303 [2]. The UE registers with the ProSe Function residing in the HPLMN.

7.2.2.2 UE registration procedure initiation

Based on pre-configuration, if the UE is authorised to perform EPC-level ProSe discovery in the registered PLMN, it shall initiate the UE registration procedure when the UE is triggered by upper layers to obtain EPC-level ProSe discovery services and the UE has no corresponding EPC ProSe User ID.

The UE initiates the UE registration procedure by sending a UE_REGISTRATION_REQUEST message with the UE identity set to the UE's IMSI. If the UE intends to use EPC support for WLAN direct discovery and communication and if the UE uses a permanent WLAN link layer identifier, then the UE also includes the WLAN link layer identifier in the UE_REGISTRATION_REQUEST message.

Figure 7.2.2.2.1 illustrates the interaction of the UE and the ProSe Function in the UE registration procedure.

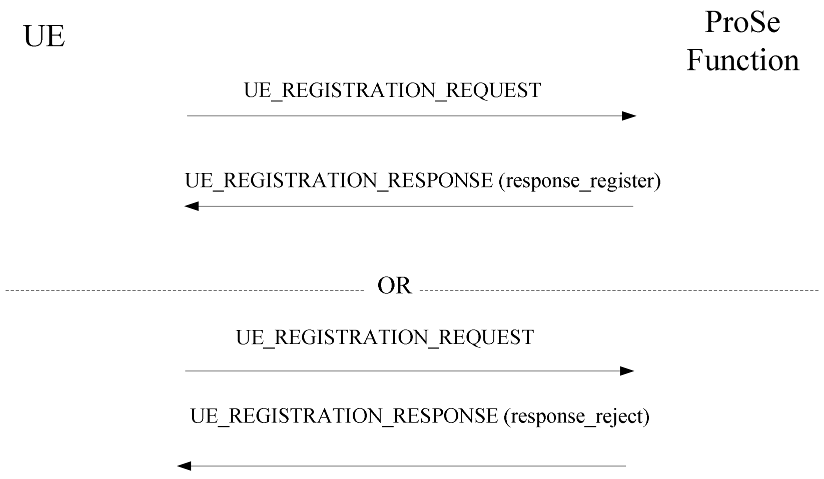
UE
ProSe Function
UE_REGISTRATION_REQUEST
UE_REGISTRATION_RESPONSE (response_register)
OR
UE_REGISTRATION_REQUEST
UE_REGISTRATION_RESPONSE (response_reject)

Figure 7.2.2.2.1: UE registration procedure

7.2.2.3 UE registration procedure accepted by the ProSe Function

Upon receiving a UE_REGISTRATION_REQUEST message, the ProSe Function interacts with the HSS in order to authenticate the user and check whether the user is authorised to use EPC-level ProSe discovery services corresponding to the IMSI contained in the UE_REGISTRATION_REQUEST message in the registered PLMN.

If the ProSe Function contains all the settings related to authentication and authorisation for the user corresponding to the IMSI contained in the UE_REGISTRATION_REQUEST message then the ProSe function need not interact with the HSS and the ProSe Function checks locally if the user is authorised to use EPC-level ProSe discovery services.

If the UE is authorised to use EPC-level ProSe discovery services, the ProSe Function generates an EPC ProSe User ID corresponding to the IMSI contained in the UE_REGISTRATION_REQUEST message and shall send a UE_REGISTRATION_RESPONSE message containing a <response-register> element to the UE with the EPC ProSe User ID. The EPC ProSe User ID is a number generated by the ProSe Function that is unique within the ProSe Function on a per UE basis.

7.2.2.4 UE registration procedure completion by the UE

Upon receipt of the UE_REGISTRATION_RESPONSE message containing a <response-register> element the UE stores the EPC ProSe User ID and may start the application registration procedure.

7.2.2.5 UE registration procedure not accepted by the ProSe Function

If the UE_REGISTRATION_REQUEST message is not accepted by the ProSe Function, the ProSe Function shall send a UE_REGISTRATION _RESPONSE message containing a <response-reject> element to the UE including an appropriate PC3 EPC Control Protocol cause value.

If the UE is not authorised for EPC-level ProSe discovery, the ProSe Function shall send the UE_REGISTRATION_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #3 "UE authorisation failure".

7.2.3 Application registration procedure

7.2.3.1 General

The purpose of the application registration procedure is for the UE to activate EPC-level Prose discovery for a specific application as defined in 3GPP TS 23.303 [2]. The UE registers the specific application with the ProSe Function residing in the HPLMN.

7.2.3.2 Application registration procedure initiation

When the user uses applications on the UE, an Application ID is used to identify the corresponding application server platform. When the user registers an application with the application server, the user is designated an Application Layer User ID. If the application requires EPC-level ProSe discovery, the UE is configured with the data structure of the Application IDs and the Application Layer User ID. This step is performed using mechanisms outside of the scope of 3GPP. The user may have multiple Application Layer User IDs for an application, but may choose to register only one of these to activate EPC-level ProSe discovery. The UE shall initiate the application registration procedure after successfully completing the UE registration procedure.

If the UE is authorised to perform EPC-level ProSe discovery in the registered PLMN, it shall initiate the application registration procedure when the UE is triggered by upper layers to activate EPC-level Prose discovery for a specific application and the application is not registered.

The UE initiates the application registration procedure by sending an APPLICATION_REGISTRATION_REQUEST message by including a new transaction ID, the UE's EPC ProSe User ID, the Application ID for the application that is to be registered and the user's Application Layer User ID for the application that is to be registered.

NOTE: A UE can include one or multiple transactions in one APPLICATION_REGISTRATION_REQUEST message for different Application IDs, and receive corresponding <response-register> element or <response-reject> element in the APPLICATION_REGISTRATION_RESPONSE message for each respective transaction. In the following description of the application registration procedure, only one transaction is included.

Figure 7.2.3.2.1 illustrates the interaction of the UE and the ProSe Function in the application registration procedure.

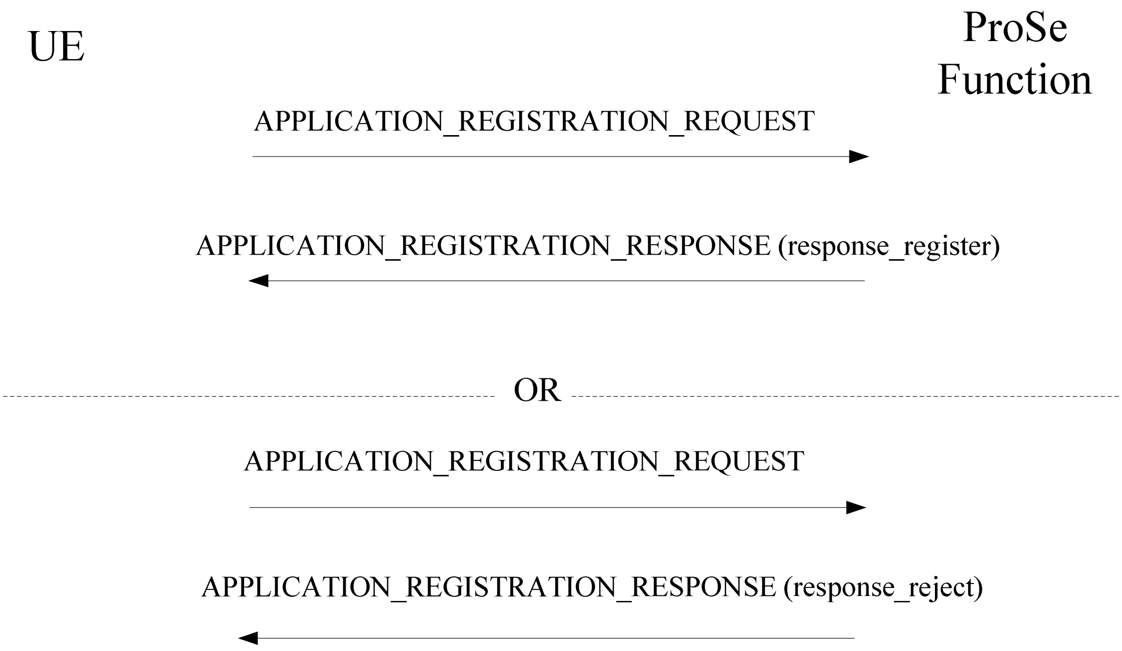
UE
ProSe Function
APPLICATION_REGISTRATION_REQUEST
APPLICATION_REGISTRATION_RESPONSE (response_register)
OR
APPLICATION_REGISTRATION_REQUEST
APPLICATION_REGISTRATION_RESPONSE (response_reject)

Figure 7.2.3.2.1: Application registration procedure

7.2.3.3 Application registration procedure accepted by the ProSe Function

Upon receiving an APPLICATION_REGISTRATION_REQUEST message, the ProSe Function retrieves the user profile based on the UE's EPC ProSe User ID included in the APPLICATION_REGISTRATION_REQUEST message. The ProSe Function then checks if the list of authorised applications in the user's profile includes the requested application based on the Application ID in the APPLICATION_REGISTRATION_REQUEST message. The user's profile is either stored locally in the ProSe Function or if that is not the case then it is retrieved from the HSS.

If the check is successful then the ProSe Function sends a request to the application server so that the user of this application identified by Application Layer User ID in the APPLICATION_REGISTRATION_REQUEST message can use EPC-level ProSe discovery for that application.

If the user is authorised to use EPC-level ProSe discovery for the specified application, the ProSe Function generates Allowed Range which contains the set of allowed range classes corresponding to the Application ID contained in the APPLICATION_REGISTRATION_REQUEST message. The ProSe Function shall send an APPLICATION_REGISTRATION_RESPONSE message containing a <response-register> element to the UE with transaction ID set to the value of the transaction ID received in the APPLICATION_REGISTRATION_REQUEST message from the UE and the Allowed Range. The allowed range of classes for each Application ID is stored in the ProSe Function.

7.2.3.4 Application registration procedure completion by the UE

Upon receipt of the UE_REGISTRATION_RESPONSE message, if the transaction ID contained in the <response-register> element matches the value sent by the UE in an APPLICATION_REGISTRATION_REQUEST message the UE stores the set of Allowed Range classes for this Application ID and may start the proximity request procedure.

7.2.3.5 Application registration procedure not accepted by the ProSe Function

If the APPLICATION_REGISTRATION_REQUEST message is not accepted by the ProSe Function, the ProSe Function shall send an APPLICATION_REGISTRATION_RESPONSE message containing a <response-reject> element to the UE including an appropriate PC3 EPC Control Protocol cause value.

If the application corresponding to the Application ID contained in the APPLICATION_REGISTRATION_REQUEST message is not authorised for EPC-level ProSe Discovery in the registered PLMN, the ProSe Function shall send the

APPLICATION_REGISTRATION_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #1 "Invalid Application".

If the UE is not authorised for EPC-level ProSe Discovery, the ProSe Function shall send the APPLICATION_REGISTRATION_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #3 "UE authorisation failure".

7.2.4 Proximity request procedure

7.2.4.1 General

The purpose of the proximity request procedure is to allow a UE (UE A) to request to be alerted when it enters in proximity with a targeted UE (UE B) as defined in 3GPP TS 23.303 [2]. UE A performs the proximity request procedure with the ProSe Function residing in the HPLMN.

7.2.4.2 Proximity request procedure initiation

Before initiating the proximity request procedure, UE A needs to register the user's Application Layer User ID A with ProSe Function A as described in subclause 7.2.3. UE A shall initiate the proximity request procedure when triggered by upper layers to activate EPC-level Prose discovery for a specific application and for a specific targeted user identified via its Application Layer User ID B.

UE A initiates the proximity request procedure by sending a PROXIMITY_REQUEST message to ProSe Function A by including a new transaction ID, UE A's EPC ProSe User ID, the Application ID for the application for which the request is made, UE A's Application Layer User ID (Application Layer User ID A), the Application Layer User ID of UE B (Application Layer User ID B), a Range Class value selected from the set of allowed range classes for this application, UE A's Current Location with the best known accuracy and a Time Window indicating the time interval during which the request is valid.

NOTE: A UE can include one or multiple transactions in one PROXIMITY_REQUEST message for different Application IDs, and receives corresponding <response-accept> element or <response-reject> element in the PROXIMITY_REQUEST_RESPONSE message for each respective transaction. In the following description of the Proximity Request procedure, only one transaction is included.

If UE A, subsequent to successful proximity detection with UE B, wishes to engage in WLAN direct discovery and communication, UE A also includes a WLAN Indication in the PROXIMITY_REQUEST message.

Figure 7.2.4.2.1 illustrates the interaction of the UE and the ProSe Function in the rroximity request procedure.

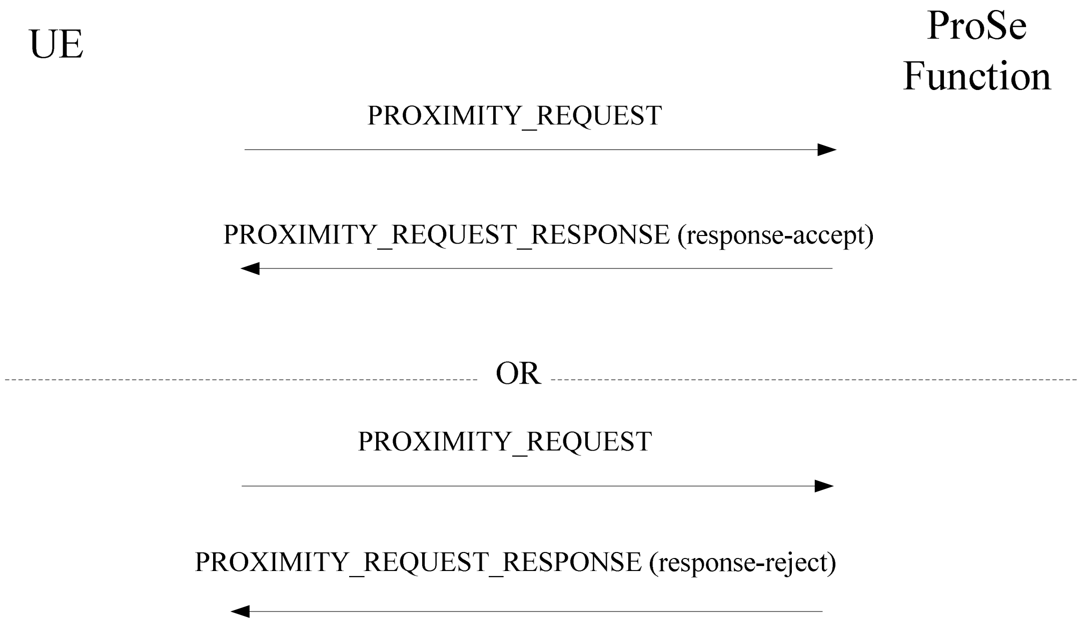
UE
ProSe Function
PROXIMITY_REQUEST
PROXIMITY_REQUEST_RESPONSE (response-accept)
OR
PROXIMITY_REQUEST
PROXIMITY_REQUEST_RESPONSE (response-reject)

Figure 7.2.4.2.1: Proximity request procedure

7.2.4.3 Proximity request procedure accepted by the ProSe Function

Upon receiving a PROXIMITY_REQUEST message from UE A, ProSe Function A retrieves the user profile based on the UE's EPC ProSe User ID included in the PROXIMITY_REQUEST message. ProSe Function A then checks that UE A has previously registered the application identified with the Application ID in the PROXIMITY_REQUEST message and that the Range Class value belongs to the set of allowed Range Classes for this application.

If the check is successful then ProSe Function A interacts with the Application Server to obtain the identifier of ProSe Function B that owns the user profile of the targeted user, as well as its EPC ProSe User ID (EPC ProSe User ID B). ProSe Function A then propagates the proximity request to ProSe Function B on the targeted UE side (the B-side). The Current Location of UE A included in the PROXIMITY_REQUEST message may be used by ProSe Function B to determine whether the proximity request is accepted or not.

NOTE: The mechanism used by ProSe Function B to determine acceptance or non-acceptance of PROXIMITY_REQUEST messages is outside the scope of this specification.

If the proximity request is accepted by the B-side, ProSe Function A stores Application Layer User ID A, Application Layer User ID B, EPC ProSe User ID B, Range Class and Time Window in the UE A's context identified with EPC ProSe User ID A. The WLAN Indication is also stored, if it was included in the PROXIMITY_REQUEST message. Then ProSe Function A shall initiate location reporting for UE A and send a PROXIMITY_REQUEST_RESPONSE message containing a <response-accept> element to UE A with transaction ID set to the value of the transaction ID received in the PROXIMITY_REQUEST message from UE A.

7.2.4.4 Proximity request procedure completion by the UE

Upon receipt of the PROXIMITY_REQUEST_RESPONSE message, if the transaction ID contained in the <response-accept> element matches the value sent by the UE in a PROXIMITY_REQUEST message, the proximity request procedure is successfully completed.

7.2.4.5 Proximity request procedure not accepted by the ProSe Function

If the PROXIMITY_REQUEST message is not accepted by the ProSe Function, the ProSe Function shall send a PROXIMITY_REQUEST_RESPONSE message containing a <response-reject> element to the UE including an appropriate PC3 EPC Control Protocol cause value.

If the application corresponding to the Application ID contained in the PROXIMITY_REQUEST message is not authorised for EPC-level ProSe discovery, the ProSe Function shall send the PROXIMITY_REQUEST_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #1 "Invalid Application".

If the application corresponding to the Application ID contained in the PROXIMITY_REQUEST message has not been registered for EPC-level ProSe discovery, the ProSe Function shall send the PROXIMITY_REQUEST_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #4 "Application not registered".

If the requested Range Class value is not allowed for this application, the ProSe Function shall send the PROXIMITY_REQUEST_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #5 "Range Class not allowed for this application".

If based on the Current Location the B-side determines that UE A and targeted UE B are unlikely to enter proximity within the requested Time Window, the ProSe Function shall send the PROXIMITY_REQUEST_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #6 "Proximity detection unlikely within requested time window".

If the ProSe Function determines that the targeted UE has not registered the application identified with Application ID, the ProSe Function shall send the PROXIMITY_REQUEST_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #7 "Targeted user not registered for this application".

If the B-side rejects to validate the proximity request, the ProSe Function shall send the PROXIMITY_REQUEST_RESPONSE message containing a <response-reject> element with PC3 EPC Control Protocol cause value #8 "Proximity validation rejected by B side".

7.2.5 Proximity alert procedure

7.2.5.1 General

The purpose of the proximity alert procedure is to inform the UE (UE A) that it has been determined to be in proximity with the targeted UE (UE B) as defined in 3GPP TS 23.303 [2]. If UE A has indicated in the proximity request procedure that it wishes to engage in WLAN direct discovery and communication with UE B, the proximity alert procedure is also used to provide Assistance Information that expedites the WLAN direct discovery and communication to both UE A and UE B. The proximity alert procedure is initiated by the ProSe Function residing in the HPLMN.

7.2.5.2 Proximity alert procedure initiation by the network

When ProSe Function A on the A-side determines that UE A and UE B are in proximity, it cancels the location reporting for UE A with the SUPL Location Platform and sends a PROXIMITY_ALERT message to UE A including the Application ID, Application Layer User ID A and Application Layer User ID B.

UE A may have registered multiple proximity requests for applications with different Application IDs. In this case the ProSe Function may combine the multiple alerts for each of the different Application IDs and send a combined PROXIMITY_ALERT message to the UE.

If UE A's context contains a WLAN Indication, the ProSe Function generates Assistance Information for WLAN direct discovery and communication according to the underlying WLAN technology, includes the Assistance Information in the PROXIMITY ALERT message and forwards the alert towards the B-side.

If the context of UE A does not contain WLAN Indication then ProSe Function A sends a cancellation request towards ProSe Function B.

After transmitting the PROXIMITY_ALERT message to UE A and alerting (or sending a cancellation request to) the B-side, the ProSe Function deletes the information related to this specific Proximity Request in UE A's context.

NOTE: If UE A has signalled a permanent WLAN Link Layer ID during UE Registration procedure as described in subclause 7.2.2, the WLAN Link Layer ID for UE A is retrieved from UE A's context; otherwise a random WLAN Link Layer ID is generated for UE A by the ProSe Function. Similarly, if during the Proximity Request procedure the ProSe Function has received a permanent WLAN Link Layer ID for UE B, the WLAN Link Laye ID for UE B is retrieved from UE A's context; otherwise a random WLAN Link Layer ID is generated for UE B by the ProSe Function.

When ProSe Function B is alerted that UE A and UE B are in proximity, it cancels the location reporting for UE B and shall send a PROXIMITY_ALERT message to UE B including the Application ID, Application Layer User ID A and Application Layer User ID B.

Figure 7.2.5.2.1 illustrates the interaction of the UE and the ProSe Function in the proximity alert procedure.

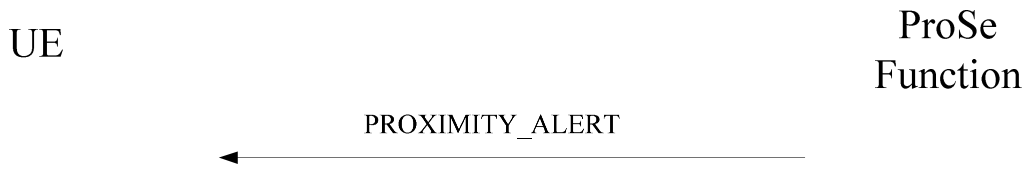

Figure 7.2.5.2.1: Proximity alert procedure

7.2.5.3 Proximity alert procedure completion by the UE

Upon receipt of the PROXIMITY_ALERT message the UE shall inform the application identified via the Application ID in the PROXIMITY_ALERT message including Application Layer User ID A and Application Layer User ID B. If the Assistance Information for WLAN direct discovery and communication is included in the PROXIMITY_ALERT message, the UE uses this information to engage in WLAN direct discovery and communication with the peer UE.

7.2.6 UE deregistration procedure

7.2.6.1 General

The UE deregistration procedure is used to deregister the UE for EPC-level ProSe discovery services. It can be initiated at any time by the UE or by the ProSe Function residing in the HPLMN.

7.2.6.2 UE-initiated UE deregistration procedure

7.2.6.2.1 UE-initiated UE deregistration procedure initiation

When the UE decides to deregister for EPC-level ProSe discovery services, it shall send the UE_DEREGISTRATION_REQUEST message to the ProSe Function residing in the HPLMN. The message includes the EPC ProSe User ID.

Figure 7.2.6.2.1.1 illustrates the interaction of the UE and the ProSe Function in the UE-initiated UE deregistration procedure.

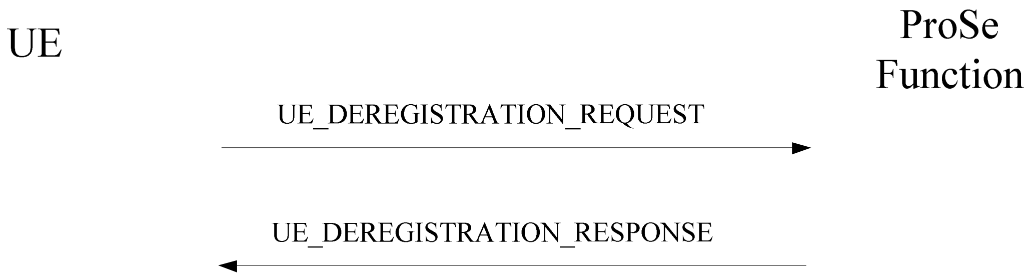

Figure 7.2.6.2.1.1: UE-initiated UE deregistration procedure

7.2.6.2.2 UE-initiated UE deregistration procedure accepted by the ProSe Function

Upon receiving the UE_DEREGISTRATION_REQUEST message, the ProSe Function retrieves the user profile based on the UE's EPC ProSe User ID included in the UE_DEREGISTRATION_REQUEST message, cancels any ongoing proximity alert procedures for this UE, clears the UE context and shall send a UE_DEREGISTRATION_RESPONSE message to the UE.

7.2.6.2.3 UE-initiated UE deregistration procedure completion by the UE

Upon receipt of the UE_DEREGISTRATION_RESPONSE message by the UE, the UE deregistration procedure is complete.

7.2.6.3 Network-initiated UE deregistration procedure

7.2.6.3.1 Network-initiated UE deregistration procedure initiation

When the ProSe Function residing in the HPLMN decides to deregister the UE for EPC-level ProSe discovery services, it shall send the UE_DEREGISTRATION_REQUEST to the UE.

Figure 7.2.6.3.1.1 illustrates the interaction of the UE and the ProSe Function in the network-initiated UE deregistration procedure.

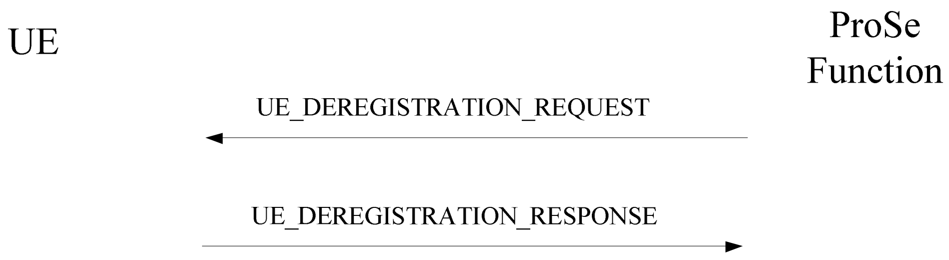

Figure 7.2.6.3.1.1: Network-initiated UE deregistration procedure

7.2.6.3.2 Network-initiated UE deregistration procedure in the UE

Upon receiving a UE_DEREGISTRATION_REQUEST message, the UE deletes all context information related to EPC-level ProSe discovery and shall send a UE_DEREGISTRATION_RESPONSE message to the network.

7.2.6.3.3 Network-initiated UE deregistration procedure completion by the network

Upon receipt of the UE_DEREGISTRATION_RESPONSE message by the ProSe Function the UE deregistration procedure is complete.

7.2.7 Proximity request cancellation procedure

7.2.7.1 General

The proximity request cancellation procedure is used by the UE to cancel an ongoing proximity request that was sent earlier as defined in 3GPP TS 23.303 [2]. The UE initiates the proximity request cancellation procedure due to occurrence of certain event (e.g.termination of the corresponding application).

7.2.7.2 Proximity request cancellation procedure initiation

The UE initiates the proximity request cancellation procedure by sending a CANCEL_PROXIMITY_REQUEST message to the ProSe Function including a new transaction ID, the UE's EPC ProSe User ID, the Application ID for the application for which the cancellation is being made and the targeted user's Application Layer User ID B.

NOTE: A UE can include one or multiple transactions in one CANCEL_PROXIMITY_REQUEST message for different Application IDs, and receive corresponding APPLICATION_REGISTRATION_RESPONSE message for each respective transaction. In the following description of the proximity request cancellation procedure, only one transaction is included.

Figure 7.2.7.2.1 illustrates the interaction of the UE and the ProSe Function in the proximity request cancellation procedure.

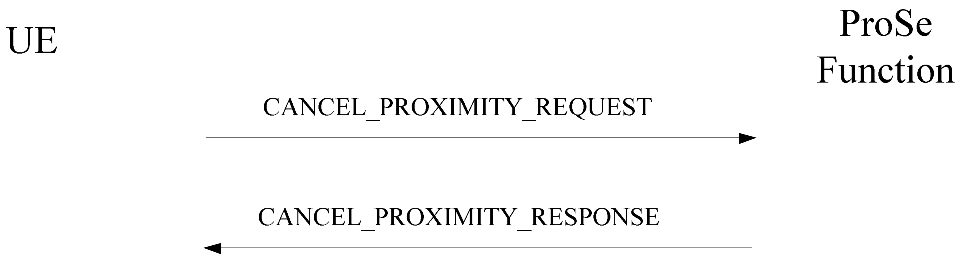

Figure 7.2.7.2.1: Proximity request cancellation procedure

7.2.7.3 Proximity request cancellation procedure handling by the ProSe Function

Upon receiving a CANCEL_PROXIMITY_REQUEST message from UE A, ProSe Function A retrieves the user profile of UE A based on UE A's EPC ProSe User ID included in the CANCEL_PROXIMITY_REQUEST message. ProSe Function A then uses the Application ID and Application Layer User ID B to identify the ProSe Function identifier of ProSe Function B which owns the context of the targeted user and forwards the cancellation request towards ProSe Function B.

If UE A has no other ongoing proximity requests then ProSe Function A cancels the location reporting for UE A and shall send a CANCEL_PROXIMITY_RESPONSE message to UE A with transaction ID set to the value of the transaction ID received in the CANCEL_PROXIMITY_REQUEST message from UE A.

7.2.7.4 Proximity request cancellation procedure completion by the UE

Upon receipt of the CANCEL_PROXIMITY_RESPONSE message with transaction ID set to the value of the transaction ID received in the CANCEL_PROXIMITY_REQUEST message the cancel proximity request procedure is complete.

7.2.8 Proximity request Validation procedure

7.2.8.1 General

If the targeted UE's profile indicates that the proximity requests for the UE need to be explicitly validated then the network uses the proximity request validation procedure to request the targeted UE (UE B) to confirm permission for the proximity requests (e.g. user B may have temporarily disabled the ProSe functionality on UE B). It is initiated by the ProSe Function residing in the HPLMN as part of the overall proximity request procedure defined in 3GPP TS 23.303 [2].

7.2.8.2 Initiation of the proximity request validation procedure

Upon reception of a proximity request from UE A, the ProSe Function on the targeted UE side (B-side) retrieves the stored profile of UE B. If UE B's profile indicates that the proximity requests for UE B need to be explicitly validated, ProSe Function B shall send the PROXIMITY_REQUEST_VALIDATION message to UE B including the Application ID of the application for which the proximity request is being validated.

Figure 7.2.8.2.1 illustrates the interaction of the targeted UE and the ProSe Function in the proximity request validation procedure.

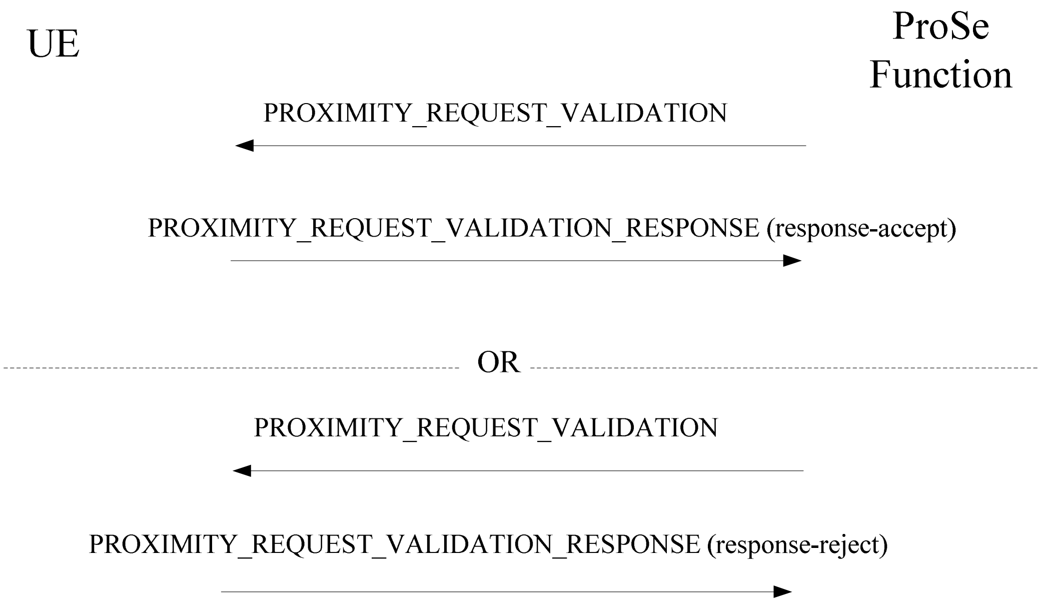
UE
ProSe Function
PROXIMITY_REQUEST_VALIDATION
PROXIMITY_REQUEST_VALIDATION_RESPONSE (response-accept)
OR
PROXIMITY_REQUEST_VALIDATION
PROXIMITY_REQUEST_VALIDATION_RESPONSE (response-reject)

Figure 7.2.8.2.1: Proximity request validation procedure

7.2.8.3 Proximity request validation procedure in the UE

Upon receiving a PROXIMITY_REQUEST_VALIDATION message, UE B checks whether the application corresponding to the Application ID included in the message is ready for accepting a proximity request from another user, UE A.

If the application corresponding to the Application ID contained in the PROXIMITY_REQUEST_VALIDATION message is ready for accepting proximity requests from other users, the targeted UE shall send the PROXIMITY_REQUEST_VALIDATION_ACCEPT message to the ProSe Function.

If the application corresponding to the Application ID contained in the PROXIMITY_REQUEST_VALIDATION message is not ready for accepting proximity requests from other users, the targeted UE shall send the PROXIMITY_REQUEST_VALIDATION_RESPONSE message with PC3 EPC Control Protocol cause value #9 "Application disabled temporarily".

7.2.8.4 Proximity request validation procedure completion by the network

Upon receipt of the PROXIMITY_REQUEST_VALIDATION_RESPONSE message containing a <response-accept> element, ProSe Function B initiates location reporting for UE B and acknowledges the proximity request towards ProSe Function A.

Upon receipt of the PROXIMITY_REQUEST_VALIDATION_RESPONSE message containing a <response-reject> element, ProSe Function B forwards an indication towards ProSe Function A that the proximity request is rejected.

7.2.9 Abnormal cases

7.2.9.1 Abnormal cases in the UE

In case of the messages listed below:

- UE_REGISTRATION_REQUEST;
- APPLICATION_REGISTRATION_REQUEST;
- PROXIMITY_REQUEST;

- UE_DEREGISTRATION_REQUEST;
- CANCEL_PROXIMITY_REQUEST; and
- PROXIMITY_REQUEST_VALIDATION.

the following abnormal cases can be identified.

a) Indication from transport layer of transmission failure of a message (e.g. after TCP retransmission timeout)

   The UE shall close the existing connection to the ProSe Function, establish a new connection and then restart the appropriate procedure.

b) No response from the ProSe Function after a message has been successfully delivered (e.g. TCP ACK has not been received)

   The UE shall retransmit the message.

7.2.9.2 Abnormal cases in the ProSe Function

In case of the messages listed below:

- UE_REGISTRATION_RESPONSE;
- APPLICATION_REGISTRATION_RESPONSE;
- PROXIMITY_REQUEST_RESPONSE;
- PROXIMITY_ALERT;
- UE_DEREGISTRATION_RESPONSE;
- CANCEL_PROXIMITY_RESPONSE; and
- PROXIMITY_REQUEST_VALIDATION_RESPONSE.

the following abnormal cases can be identified.

a) Indication from the lower layer of transmission failure of a message

   After receiving an indication from lower layer that a message has not been successfully acknowledged (e.g. TCP ACK is not received), the ProSe Function shall abort the procedure.

8 EPC support for WLAN direct discovery and communication

Editor's note: This clause will describe the procedures for EPC support for WLAN direct discovery and communication based on 3GPP TS 23.303 [2].

9 Handling of unknown, unforeseen, and erroneous protocol data

Editor's note: This clause will specify handling of error cases.

10 ProSe direct communication

10.1 General

This clause describes the procedures at the UE, and between UEs, for ProSe direct communication over the PC5 interface.

When served by E-UTRAN, the UE shall be authorised for ProSe direct communication in the registered PLMN based on the service authorisation procedure as specified in clause 5, before initiating one-to-many ProSe direct communication.

When not served by E-UTRAN, the UE shall be authorised for ProSe direct communication when not served by E-UTRAN based on the service authorisation procedure as specified in subclause 5, before initiating one-to-many ProSe direct communication.

10.2 One-to-many ProSe direct communication

One-to-many ProSe direct communication is applicable only to ProSe-enabled Public Safety UEs and when authorised, can apply when the UE is served by E-UTRAN and when the UE is outside of E-UTRAN coverage.

Upon receiving a request from upper layers to send data for ProSe direct communication in a given group, the UE shall initiate the procedure for ProSe direct communication transmission.

The UE shall obtain the ProSe direct communication policy parameters for that group as specified in subclause 5.

If the ProSe direct communication policy parameters indicate that the UE is configured to use IPv6 for that group, the UE shall auto-configures a link local IPv6 Address following procedures defined in RFC 4862 [15]. This address can only be used as the source IP address for one-to-many ProSe direct communication.

If the ProSe Direct communication policy parameters group indicate that the UE is configured to use IPv4 for that group, then the UE shall:

- use the configured IPv4 address for that group as source address; or
- if there is no configured IPv4 address for that group, use Dynamic Configuration of IPv4 Link-Local Addresses as specified in IETF RFC 3927 [16].

When the UE is served by E-UTRAN, the UE requests the ProSe direct communication parameters from the lower layers (see 3GPP TS 36.331 [12]). If the lower layers indicate that ProSe direct communication is not supported by the network, the UE shall not perform direct communication. Else if the lower layers indicate that ProSe direct communication is supported by the network:

- if the lower layers indicate that the transmission resources for direct communication are already provided by the eNodeB and UE is not required to explicitly request transmission resources from the eNodeB, the UE shall start transmission of direct communication;
- if the lower layers indicate that the UE is required to request transmission resources from the eNodeB for ProSe direct communication and the UE is in EMM-IDLE mode, the UE shall perform a Service Request procedure to allow the UE to request transmission resources from the eNodeB as specified in 3GPP TS 24.301 [11] to transfer to EMM-CONNECTED mode. Upon successful completion of the Service Request procedure, the UE shall trigger the lower layers to request transmission resources from the eNodeB as specified in 3GPP TS 36.331 [12]. Upon indication from the lower layers that the transmission resources have been obtained successfully from the eNodeB, the UE shall start transmission of direct communication; or
- if the lower layers indicate that the UE is required to request transmission resources from the eNodeB for ProSe direct communication and the UE is in EMM-CONNECTED mode, the UE shall trigger the lower layers to request transmission resources from the eNodeB as specified in 3GPP TS 36.331 [12]. Upon indication from lower layers that the transmission resources have been obtained successfully from the eNodeB, the UE shall start transmission of direct communication.

When the UE is not served by E-UTRAN, the UE shall initiate ProSe direct communication based on the radio parameters provisioned in the service authorisation procedure as specified in subclause 5.

11 Message functional definitions and contents

11.1 Overview

This clause contains the definition and contents of the messages used in the procedures described in the present document.

11.2 ProSe discovery messages

11.2.1 General

This subclause defines the XML schema and MIME type related to ProSe direct discovery messages and EPC-level ProSe discovery messages.

This subclause also defines the format of the PC5_DISCOVERY message transmitted over the PC5 interface.

11.2.2 application/3gpp-prose+xml

The MIME type is used to carry information related to the ProSe discovery operation. It shall be coded as an XML document containing one of the following ProSe discovery messages:

- DISCOVERY_REQUEST;
- DISCOVERY_RESPONSE;
- MATCH_REPORT;
- MATCH_REPORT_ACK;
- UE_REGISTRATION_REQUEST;
- UE_REGISTRATION_RESPONSE;
- APPLICATION_REGISTRATION_REQUEST;
- APPLICATION_REGISTRATION_RESPONSE;
- PROXIMITY_REQUEST;
- PROXIMITY_REQUEST_RESPONSE;
- PROXIMITY_ALERT;
- UE_DEREGISTRATION_REQUEST;
- UE_DEREGISTRATION_RESPONSE;
- CANCEL_PROXIMITY_REQUEST;
- CANCEL_PROXIMITY_RESPONSE;
- PROXIMITY_REQUEST_VALIDATION; or
- PROXIMITY_REQUEST_VALIDATION_RESPONSE.

Each of those messages is presented in the XML document as an XML element named after the corresponding message.

Editor's note: The name space for XML schema "urn:3GPP:ns:ProSe:Discovery:2014" has not been registered with 3GPP yet and needs to be confirmed.

11.2.3 XML Schema

Implementations in compliance with the present document shall implement the XML schema defined below for messages used in ProSe direct discovery procedures over PC3 interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="urn:3GPP:ns:ProSe:Discovery:2014"
           elementFormDefault="qualified"
           targetNamespace="urn:3GPP:ns:ProSe:Discovery:2014">
        <xs:annotation>
            <xs:documentation>
                Info for ProSe Discovery Control Messages Syntax
            </xs:documentation>
        </xs:annotation>


  <!-- Complex types defined for parameters with complicate structure -->

  <xs:complexType name="AppID-info">
    <xs:sequence>
      <xs:element name="OS-ID">
        <xs:simpleType>
          <xs:restriction base="xs:hexBinary">
            <xs:length value="16"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="OS-App-ID" type="xs:string"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PLMN-info">
    <xs:sequence>
      <xs:element name="mcc" type="xs:integer"/>
        <xs:element name="mnc" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="DiscFilter-info">
    <xs:sequence>
      <xs:element name="filter-ID" type="xs:integer"/>
      <xs:element name="ProSe-Application-Code" type="xs:hexBinary"/>
      <xs:element name="ProSe-Application-Mask" type="xs:hexBinary" maxOccurs="unbounded"/>
      <xs:element name="TTL-timer-T4002" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="WLANAssistance-info">
    <xs:sequence>
      <xs:element name="ssid" type="xs:string"/>
      <xs:element name="WLAN-secret-key" type="xs:string"/>
      <xs:element name="group-owner-indication" type="xs:boolean"/>
      <xs:element name="P2P-device-address-self" type="xs:hexBinary" minOccurs="0"/>
      <xs:element name="P2P-device-address-peer" type="xs:hexBinary" minOccurs="0"/>
      <xs:element name="operation-channel" type="xs:integer"/>
      <xs:element name="validity-time" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="TimeParameter-info"/>
  <!-- Need to be further specified depending on RAN2 and SA3 specificaiton -->
  <xs:complexType name="metadata-Info"/>
  <!-- Metadata structure in XML is FFS -->

  <!-- Complex types defined for transaction-level -->

  <xs:complexType name="AnnounceRsp-info">
    <xs:sequence>
```

```
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="ProSe-Application-Code" type="xs:hexBinary"/>
      <xs:element name="validity-timer-T4000" type="xs:integer"/>
      <xs:element name="discovery-type" type="xs:integer"/>
      <xs:element name="discovery-key" type="xs:hexBinary"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="MonitorRsp-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="discovery-filter" type="DiscFilter-info" maxOccurs="unbounded"/>
      <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>


  <xs:complexType name="DiscReq-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="command" type="xs:integer"/>
      <xs:element name="UE-identity" type="xs:hexBinary"/>
      <xs:element name="ProSe-Application-ID" type="xs:string"/>
      <xs:element name="application-identity" type="AppID-info"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="RejectRsp-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="PC3-control-protocol-cause-value" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="MatchRep-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="ProSe-Application-Code" type="xs:hexBinary"/>
      <xs:element name="UE-identity" type="xs:hexBinary"/>
      <xs:element name="Monitored-PLMN-ID" type="PLMN-info"/>
      <xs:element name="VPLMN-ID" type="PLMN-info" minOccurs="0"/>
      <xs:element name="MIC" type="xs:hexBinary"/>
      <xs:element name="time-parameter" type="TimeParameter-info"/>
      <xs:element name="Metadata-flag" type="xs:boolean"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="MatchAck-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="ProSe-Application-ID" type="xs:string"/>
      <xs:element name="validity-timer-T4004" type="xs:integer"/>
      <xs:element name="metadata" type="metadata-Info" minOccurs="0"/>
      <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="MatchReject-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="PC3-control-protocol-cause-value" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>
  <!-- Complex types defind for Message-level -->

  <xs:complexType name="prose-direct-discovery-request">
    <xs:sequence>
     <xs:element name="discovery-request" type="DiscReq-info" maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-direct-discovery-response">
    <xs:sequence>
      <xs:element name="response-announce" type="AnnounceRsp-info" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="response-monitor" type="MonitorRsp-info" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="response-reject" type="RejectRsp-info" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-direct-discovery-match-report">
    <xs:sequence>
     <xs:element name="match-report" type="MatchRep-info" maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-direct-discovery-match-report-ack">
    <xs:sequence>
      <xs:element name="match-ack" type="MatchAck-info" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="match-reject" type="MatchReject-info" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--  extension allowed -->
  <xs:complexType name="DiscMsgExtType">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--  XML attribute for any future extensions  -->
  <xs:complexType name="anyExtType">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>


<!-- Complex types defined for EPC-level Discovery transaction-level -->

  <xs:complexType name="UeRegReq-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="UE-identity" type="xs:hexBinary" />
      <xs:element name="WLAN-Link-Layer-ID" type="xs:hexBinary" minOccurs="0"/>
      <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="UeRegRsp-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="EPC-ProSe-User-ID" type="xs:string"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="UeRegReject-info">
```

```
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="PC3-EPC-control-protocol-cause-value" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="AppRegReq-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="EPC-ProSe-User-ID" type="xs:string"/>
      <xs:element name="application-identity" type="AppID-info"/>
      <xs:element name="Application-Layer-User-ID" type="xs:string"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="AppRegRsp-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="allowed-range" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="AppRegReject-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="PC3-EPC-control-protocol-cause-value" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="ProximityReq-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="EPC-ProSe-User-ID" type="xs:string"/>
      <xs:element name="application-identity" type="AppID-info"/>
      <xs:element name="Application-Layer-User-ID-A" type="xs:string"/>
      <xs:element name="Application-Layer-User-ID-B" type="xs:string"/>
      <xs:element name="range-class" type="xs:integer"/>
      <xs:element name="UE-A-location" type="xs:string"/>
      <xs:element name="time-window" type="xs:integer"/>
      <xs:element name="WLAN-indication" type="xs:boolean" minOccurs="0"/>
      <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="ProximityReqAccept-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="ProximityReqReject-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="PC3-EPC-control-protocol-cause-value" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="ProximityAlert-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="application-identity" type="AppID-info"/>
      <xs:element name="Application-Layer-User-ID-A" type="xs:string"/>
      <xs:element name="Application-Layer-User-ID-B" type="xs:string"/>
      <xs:element name="assistance-information" type="WLANAssistance-info" minOccurs="0" />
      <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="UeDeregReq-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="EPC-ProSe-User-ID" type="xs:string"/>
      <xs:element name="PC3-EPC-control-protocol-cause-value" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="UeDeregRsp-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="CancelProximityReq-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="EPC-ProSe-User-ID-A" type="xs:string"/>
      <xs:element name="application-identity" type="AppID-info"/>
      <xs:element name="EPC-ProSe-User-ID-B" type="xs:string"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="CancelProximityRsp-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="ProximityReqValidation-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="application-identity" type="AppID-info"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="ProximityReqValidationRspAccept-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>

  <xs:complexType name="ProximityReqValidationRspReject-info">
    <xs:sequence>
      <xs:element name="transaction-ID" type="xs:integer"/>
      <xs:element name="PC3-EPC-control-protocol-cause-value" type="xs:integer"/>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
  </xs:complexType>


<!-- Complex types defined for EPC level Discovery Message-level -->

  <xs:complexType name="prose-epc-level-ue-registration-request">
    <xs:sequence>
     <xs:element name="ue-registration-request" type="UeRegReq-info" maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-ue-registration-response">
    <xs:sequence>
     <xs:element name="response-register" type="UeRegRsp-info" minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="response-reject" type="UeRegReject-info" minOccurs="0"
maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-application-registration-request">
    <xs:sequence>
     <xs:element name="application-registration-request" type="AppRegReq-info"
maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-application-registration-response">
    <xs:sequence>
     <xs:element name="response-register" type="AppRegRsp-info" minOccurs="0"
maxOccurs="unbounded"/>
     <xs:element name="response-reject" type="AppRegReject-info" minOccurs="0"
maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-proximity-request">
    <xs:sequence>
     <xs:element name="proximity-request" type="ProximityReq-info" maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-proximity-request-response">
    <xs:sequence>
     <xs:element name="response-accept" type="ProximityReqAccept-info" minOccurs="0"
maxOccurs="unbounded"/>
     <xs:element name="response-reject" type="ProximityReqReject-info" minOccurs="0"
maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-proximity-alert">
    <xs:sequence>
     <xs:element name="proximity-alert" type="ProximityAlert-info" maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-ue-deregistration-request">
    <xs:sequence>
     <xs:element name="ue-deregistration-request" type="UeDeregReq-info" maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-ue-deregistration-response">
    <xs:sequence>
     <xs:element name="ue-deregistration-response" type="UeDeregRsp-info" maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-cancel-proximity-request">
    <xs:sequence>
```

```
     <xs:element name="cancel-proximity-request" type="CancelProximityReq-info"
maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-cancel-proximity-response">
    <xs:sequence>
     <xs:element name="cancel-proximity-response" type="CancelProximityRsp-info"
maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-proximity-request-validation">
    <xs:sequence>
     <xs:element name="proximity-request-validation" type="ProximityReqValidation-info"
maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="prose-epc-level-proximity-request-validation-response">
    <xs:sequence>
     <xs:element name="response-accept" type="ProximityReqValidationRspAccept-info" minOccurs="0"
maxOccurs="unbounded"/>
     <xs:element name="response-reject" type="ProximityReqValidationRspReject-info" minOccurs="0"
maxOccurs="unbounded"/>
     <xs:element name="anyExt" type="anyExtType" minOccurs="0"/>
     <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>


<!--  Top levelDiscovery Message definition  -->
  <xs:element name="prose-discovery-message">
    <xs:complexType>
      <xs:choice>
        <xs:element name="DISCOVERY_REQUEST" type="prose-direct-discovery-request"/>
        <xs:element name="DISCOVERY_RESPONSE" type="prose-direct-discovery-response"/>
        <xs:element name="MATCH_REPORT" type="prose-direct-discovery-match-report"/>
        <xs:element name="MATCH_REPORT_ACK" type="prose-direct-discovery-match-report-ack"/>
        <xs:element name="UE_REGISTRATION_REQUEST" type="prose-epc-level-ue-registration-request"/>
        <xs:element name="UE_REGISTRATION_RESPONSE" type="prose-epc-level-ue-registration-
response"/>
        <xs:element name="APPLICATION_REGISTRATION_REQUEST" type="prose-epc-level-application-
registration-request"/>
        <xs:element name="APPLICATION_REGISTRATION_RESPONSE" type="prose-epc-level-application-
registration-response"/>
        <xs:element name="PROXIMITY_REQUEST" type="prose-epc-level-proximity-request"/>
        <xs:element name="PROXIMITY_REQUEST_RESPONSE" type="prose-epc-level-proximity-request-
response"/>
        <xs:element name="PROXIMITY_ALERT" type="prose-epc-level-proximity-alert"/>
        <xs:element name="UE_DEREGISTRATION_REQUEST" type="prose-epc-level-ue-deregistration-
request"/>
        <xs:element name="UE_DEREGISTRATION_RESPONSE" type="prose-epc-level-ue-deregistration-
response"/>
        <xs:element name="CANCEL_PROXIMITY_REQUEST" type="prose-epc-level-cancel-proximity-
request"/>
        <xs:element name="CANCEL-PROXIMITY_REQUEST" type="prose-epc-level-cancel-proximity-
response"/>
        <xs:element name="PROXIMITY_REQUEST_VALIDATION" type="prose-epc-level-proximity-request-
validation"/>
        <xs:element name="PROXIMITY_REQUEST_VALIDATION_RESPONSE" type="prose-epc-level-proximity-
request-validation-response"/>
        <xs:element name="message-ext" type="DiscMsgExtType"/>
        <xs:any namespace="##other" processContents="lax"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

An entity receiving the XML body ignores any unknown XML element and any unknown XML attribute.

11.2.4 Semantics

Editor's note: The description of the data semantics in subclause 11.2.4 can be removed if they are deemed duplicated information already described by the XML schema.

11.2.4.1 General

The <prose-discovery-message> element is the root element of this XML document and it can be one of the following elements:

- <DISCOVERY_REQUEST>;
- <DISCOVERY_RESPONSE>;
- <MATCH_REPORT>;
- <MATCH_REPORT_ACK>;
- <UE_REGISTRATION_REQUEST>;
- <UE_REGISTRATION_RESPONSE>;
- <APPLICATION_REGISTRATION_REQUEST>;
- <APPLICATION_REGISTRATION_RESPONSE>;
- <PROXIMITY_REQUEST>;
- <PROXIMITY_REQUEST_RESPONSE>;
- <PROXIMITY_ALERT>;
- <UE_DEREGISTRATION_REQUEST>;
- <UE_DEREGISTRATION_RESPONSE>;
- <CANCEL_PROXIMITY_REQUEST>;
- <CANCEL_PROXIMITY_RESPONSE>;
- <PROXIMITY_REQUEST_VALIDATION>; or
- <PROXIMITY_REQUEST_VALIDATION_RESPONSE>.

11.2.4.2 Semantics of <DISCOVERY_REQUEST>

The <DISCOVERY_REQUEST> element contains one or more of the following elements:

1) One or more <discovery-request> element which contains transactions sent from the UE to the ProSe Function as announcing or monitoring requests. Each <discovery-request> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.2.2.1;

   b) a <command> element containing the parameter defined in subclause 12.2.2.2;

   c) a <UE-identity> element containing the parameter defined in subclause 12.2.2.3;

   d) a <Prose-Application-ID> element containing the parameter defined in subclause 12.2.2.4; and

   e) an <application-identity> element containing the parameter defined in subclause 11.2.2.5.

11.2.4.3 Semantics of <DISCOVERY_RESPONSE>

The <DISCOVERY_RESPONSE> element contains one or more of the following elements:

1) One or more <response-announce> element which contains transactions sent from the ProSe Function to the UE as a response to an announcing request if the ProSe Function accepts the request. Each <response-announce> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.2.2.1;

   b) a <ProSe-Application-Code> element containing the parameter defined in subclause 12.2.2.6;

   c) a <validity-timer-T4000> element containing the parameter defined in 12.2.2.7;

   d) a <discovery-type> element containing the parameter defined in subclause 12.2.2.10; and

   e) a <discovery-key> element containing the parameter defined in subclause 12.2.2.9;

2) One or more <response-monitor> element which contains transactions sent from the ProSe Function to the UE as a response to an announcing request if the ProSe Function accepts the request. Each <response-monitor> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.2.2.1; and

   b) a <discovery-filter> element containing the parameter defined in subclause 12.2.2.12; and

3) One or more <response-reject> element which contains transactions sent from the ProSe Function to the UE as a response to an announcing or monitoring requests if the ProSe Function cannot accept the request. Each <response-reject> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.2.2.1; and

   b) a <PC3-control-protocol-cause-value> element containing the parameter defined in subclause 12.2.2.8.

11.2.4.4 Data semantics of <MATCH_REPORT>

The <MATCH_REPORT> element contains one or more of the following element:

1) One or more <match-report> element which contains transactions sent from the UE to the ProSe Function to report a matching of the direct discovery. Each <match-report> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.2.2.1;

   b) a <ProSe-Application-Code> element containing the parameter defined in subclause 12.2.2.6;

   c) a <UE-identity> element containing the parameter defined in subclause 12.2.2.3;

   d) a <Monitored-PLMN-id> element containing the parameter defined in subclause 12.2.2.16;

   e) a <MIC> element containing the parameter defined in subclause 12.2.2.11;

   f) a <time-parameter> element containing the parameter defined in subclause 12.2.2.18;

   g) a <metadata-flag> element containing the parameter defined in subclause 12.2.2.20; and optionally

   h) a <VPLMN-id> element containing the parameter defined in subclause 12.2.2.17.

Editor's note: The content and format of <time-parameter> is to be defined by 3GPP TSG RAN WG2.

11.2.4.5 Data semantics of <MATCH_REPORT_ACK>

The <MATCH-REPORT_ACK> element contains one or more of the following elements:

1) One or more <match-ack> element which contains transactions sent from the ProSe Function to the UE as a response to a match report if the ProSe Function accepts the report. Each <match-ack> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.2.2.1;

   b) a <ProSe-Application-ID> element containing the parameter defined in subclause 12.2.2.4;

   c) a <validity-timer-T4004> element containing the parameter defined in subclause 12.2.2.19; and optionally

   d) a <metadata> element containing the parameter defined in subclause 12.2.2.21; and

2) One or more <match-reject> element which contains transactions sent from the ProSe Function to the UE as a response to an announcing or monitoring requests if the ProSe Function cannot accept the match report. Each <match-reject> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.2.2.1; and

   b) a <PC3-control-protocol-cause-value> element containing the parameter defined in subclause 12.2.2.8.

Editor's note: The type definition of the "metadata-info" is FFS.

11.2.4.6 Semantics of <UE_REGISTRATION_REQUEST>

The <UE_REGISTRATION_REQUEST> element contains one or more of the following elements:

1) One or more <UE-register-request> element which contains transactions sent from the UE to the ProSe Function to register the UE. Each <UE-register-request > consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1;

   b) a <UE-identity> element containing the parameter defined in subclause 12.3.2.2; and

   c) a <WLAN-link-layer-ID> element containing the parameter defined in subclause 12.3.2.6.

11.2.4.7 Semantics of <UE_REGISTRATION_RESPONSE>

The <UE_REGISTRATION_RESPONSE> element contains one or more of the following elements:

1) One or more <response-register> element which contains transactions sent from the ProSe Function to the UE as a response to the UE_REGISTRATION_REQUEST message if the ProSe Function accepts the request. Each <response-register> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 11.3.2.1; and

   b) an <EPC-ProSe-User-ID> element containing the parameter defined in subclause 12.3.2.7.

2) One or more <response-reject> element which contains transactions sent from the ProSe Function to the UE as a response to the UE_REGISTRATION_REQUEST message if the ProSe Function cannot accept the request. Each <response-reject> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1; and

   b) a <PC3-EPC-control-protocol-cause-value> element containing the parameter defined in subclause 12.3.2.5.

11.2.4.8 Semantics of <APPLICATION_REGISTRATION_REQUEST>

The <APPLICATION_REGISTRATION_REQUEST> element contains one or more of the following elements:

1) One or more <application-register-request> element which contains transactions sent from the UE to the ProSe Function to activate EPC-level ProSe discovery for a specific application. Each < application-register-request > consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1;

   b) an <EPC-ProSe-User-ID> element containing the parameter defined in subclause 12.3.2.7.

   c) an <application-identity> element containing the parameter defined in subclause 12.3.2.3; and

   d) an <Application-Layer-User-ID> element containing the parameter defined in subclause 12.3.2.4.

11.2.4.9 Semantics of <APPLICATION_REGISTRATION_RESPONSE>

The <APPLICATION_REGISTRATION_RESPONSE> element contains one or more of the following elements:

1) One or more <response-register> element which contains transactions sent from the ProSe Function to the UE as a response to the APPLICATION_REGISTRATION_REQUEST message if the ProSe Function accepts the request. Each <response-register> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1; and

   b) an <allowed-range> element containing the parameter defined in subclause 12.3.2.8.

2) One or more <response-reject> element which contains transactions sent from the ProSe Function to the UE as a response to the APPLICATION_REGISTRATION_REQUEST message if the ProSe Function cannot accept the request. Each <response-reject> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1; and

   b) a <PC3-EPC-control-protocol-cause-value> element containing the parameter defined in subclause 12.3.2.5.

11.2.4.10 Semantics of <PROXIMITY_REQUEST>

The <PROXIMITY_REQUEST> element contains one or more of the following elements:

1) One or more <proximity-request> element which contains transactions sent from the UE to the ProSe Function to request to be alerted when it enters in proximity with a targeted UE. Each < proximity-request > consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1;

   b) a <EPC-ProSe-User-ID-A> element containing the parameter defined in subclause 12.3.2.7;

   c) an <application-identity> element containing the parameter defined in subclause 12.3.2.3;

   d) an <Application-Layer-User-ID-A> element containing the parameter defined in subclause 12.3.2.4;

   e) an <Application-Layer-User-ID-B> element containing the parameter defined in subclause 12.3.2.4;

   f) a <range-class> element containing the parameter defined in subclause 12.3.2.10;

   g) a <UE-A-Location> element containing the parameter defined in subclause 12.3.2.11;

   h) a <time-window> element containing the parameter defined in subclause 12.3.2.9; and

   i) a <WLAN-indication> element containing the parameter defined in subclause 12.3.2.12.

11.2.4.11 Semantics of <PROXIMITY_REQUEST_RESPONSE>

The <PROXIMITY_REQUEST_RESPONSE> element contains one or more of the following elements:

1) One or more <response-accept> element which contains transactions sent from the ProSe Function to the UE as a response to the PROXIMITY_REQUEST message if the ProSe Function accepts the request. Each <response-register> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1.

2) One or more <response-reject> element which contains transactions sent from the ProSe Function to the UE as a response to the PROXIMITY_REQUEST message if the ProSe Function cannot accept the request. Each <response-reject> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1; and

   b) a <PC3-EPC-control-protocol-cause-value> element containing the parameter defined in subclause 12.3.2.5.

11.2.4.12 Semantics of <PROXIMITY_ALERT>

The <PROXIMITY_ALERT> element contains one or more of the following elements:

1) One or more <proximity-alert> element which contains transactions sent from the ProSe Function to the (UE A) and optionally to targeted (UE B) to alert them that they have entered in proximity. Each <proximity-alert> consists of:

   a) <transaction-ID> element containing the parameter defined in subclause 12.3.2.1.

   b) an <application-identity> element containing the parameter defined in subclause 12.3.2.3;

   c) an <Application-Layer-User-ID-A> element containing the parameter defined in subclause 12.3.2.4;

   d) an <Application-Layer-User-ID-B> element containing the parameter defined in subclause 12.3.2.4; and

   e) an <assistance-information> element containing the parameter defined in subclause 12.3.2.13.

11.2.4.13 Semantics of <UE_DEREGISTRATION_REQUEST>

The <UE_DEREGISTRATION_REQUEST> element contains one or more of the following elements:

1) One or more <UE-deregister-request> element which contains transactions sent either from the UE to the ProSe Function or from the ProSe Function to the UE to deregister the UE. Each < UE-deregister-request > consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1;

   b) an <EPC-ProSe-User-ID> element containing the parameter defined in subclause 12.3.2.8; and

   c) a <PC3-EPC-control-protocol-cause-value> element containing the parameter defined in subclause 12.3.2.6.

11.2.4.14 Semantics of <UE_DEREGISTRATION_RESPONSE>

The <UE_DEREGISTRATION_RESPONSE> element contains one or more of the following elements:

1) One or more <UE-deregister-response> element which contains transactions sent either from the UE to the ProSe Function or from the ProSe Function to the UE to complete the UE deregistration. Each < UE-deregister-response > consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1.

11.2.4.15 Semantics of <CANCEL_PROXIMITY_REQUEST>

The <CANCEL_PROXIMITY_REQUEST> element contains one or more of the following elements:

1) One or more <cancel-proximity-request> element which contains transactions sent from the UE to the ProSe Function to request cancellation of an ongoing proximity request. Each < cancel_proximity-request > consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1;

   b) an <EPC-ProSe-User-ID-A> element containing the parameter defined in subclause 12.3.2.7;

   c) an <application-identity> element containing the parameter defined in subclause 12.3.2.3; and

   d) an <Application-Layer-User-ID-B> element containing the parameter defined in subclause 12.3.2.4;

11.2.4.16 Semantics of <CANCEL_PROXIMITY_RESPONSE>

The <CANCEL_PROXIMITY_RESPONSE> element contains one or more of the following elements:

1) One or more <cancel-proximity-responset> element which contains transactions sent from the ProSe Function to the UE as a response to CANCEL_PROXIMITY_REQUEST message. Each < cancel_proximity-response > consists of:

a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1;

11.2.4.17 Semantics of <PROXIMITY_REQUEST_VALIDATION>

The <PROXIMITY_REQUEST_VALIDATION> element contains one or more of the following elements:

1) One or more <proximity-request-validation> element which contains transactions sent by the ProSe Function to the targeted UE (UE B) to request confirmation of permission for proximity request. Each <proximity-request-validation> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1; and

   b) an <application-identity> element containing the parameter defined in subclause 12.3.2.3.

11.2.4.18 Semantics of <PROXIMITY_REQUEST_VALIDATION_RESPONSE>

The <PROXIMITY_REQUEST_VALIDATION_RESPONSE> element contains one or more of the following elements:

1) One or more <response-accept> element which contains transactions sent from the UE to the ProSe Function as a response to the PROXIMITY_REQUEST_VALIDATION message if the application in the UE accepts the request. Each <response-accept> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1.

2) One or more <response-reject> element which contains transactions sent from the UE to the ProSe Function as a response to the PROXIMITY_REQUEST_VALIDATION message if the application in the UE does not accept the request. Each <response-reject> consists of:

   a) a <transaction-ID> element containing the parameter defined in subclause 12.3.2.1; and

   b) a <PC3-EPC-control-protocol-cause-value> element containing the parameter defined in subclause 12.3.2.5.

11.2.5 PC5_DISCOVERY

11.2.5.1 Message definition

This message is sent by the UE over the PC5 interface. See table 11.2.5.1.1.

Message type: PC5_DISCOVERY

Direction: UE to UE

Table 11.2.5.1.1: PC5_DISCOVERY message content

Information Element	Type/Reference	Presence	Length (bits)
Discovery Type	Discovery Type 12.2.2.10	M	8
ProSe Application Code	Binary 12.2.2.6	M	184
MIC	Binary 12.2.2.11	M	32

12 General message format and information elements coding

12.1 Overview

This clause contains general message format and information elements coding for the messages used in the procedures described in the present document.

12.2 ProSe direct discovery message formats

12.2.1 Data types format in XML schema

To exchange structured information over the transport protocol, XML text format/notation is introduced.

The corresponding XML data types for the data types used in ProSe messages are provided in table 12.2.1.1.

Table 12.2.1.1: Primitive or derived types for ProSe Parameter Type

ProSe Parameter Type	Type in XML Schema
Integer	decimal
String	string
Boolean	boolean
Binary	hexBinary
Date and Time	dateTime

For complex data types described in subclause 12.2.2, an XML "complexType" can be used.

Message construction shall be compliant with W3C REC-xmlschema-2-
20041028: "XML Schema Part 2: Datatypes" [7]

12.2.2 Parameters in ProSe direct discovery messages

12.2.2.1 Transaction ID

This parameter is used to uniquely identify a PC3 Control Protocol for ProSe direct discovery transaction when it is combined with other PC3 Control Protocol for ProSe direct discovery transactions in the same transport message. The UE shall set this parameter to a new number for each outgoing new discovery request. The transaction ID is an integer in the 0-255 range.

12.2.2.2 Command

This parameter is used to indicate the type of discovery request (announce or monitor) contained in a DISCOVERY_REQUEST message. It is an integer in the 0-255 range encoded as follows:

0	Reserved
1	announce
2	monitor
3-255	Unused

12.2.2.3 UE Identity

This parameter is used to indicate the requesting UE's identity and is set to the IMSI. The IMSI consists of three parts:

- MCC: 3-digit integer;

- MNC: 2-digit integer; and

- MSIN: 10-digit or shorter integer.

Editor's note: XML parser will be used to extract MCC and MNC information from the UE Identity parameter.

12.2.2.4 Prose Application ID

This parameter is used to carry an identity used for ProSe direct discovery, identifying application related information for the ProSe-enabled UE. It is coded as specified in 3GPP TS 23.003 [4].

12.2.2.5 Application Identity

This parameter is used to identify the particular application that triggers the DISCOVERY_REQUEST message. The format of the Application Identity consists of two parts:

- OS ID: operating system identifier. The format of the OS ID is a Universally Unique IDentifier (UUID) as specified in IETF RFC 4122 [8]; and

- OS App ID: a string containing the OS specific application identifier.

NOTE: Further definition of the format of OS App ID is beyond the scope of this specification.

12.2.2.6 ProSe Application Code

This parameter is used to carry a ProSe Application Code. It is a bit string coded as specified in 3GPP TS 23.003 [4].

12.2.2.7 Validity Timer T4000

This parameter is used to carry the value of validity timer T4000 associated with a ProSe Application Code. It is an integer representing the timer value in unit of minutes.

Editor's note: The range of the timer value is FFS.

12.2.2.8 PC3 Control Protocol cause value

This parameter is used to indicate the particular reason why a DISCOVERY_REQUEST or MATCH_REPORT message from the UE has been rejected by the ProSe Function. It is an integer in the 0-255 range encoded as follows:

0	Reserved
1	Invalid Application
2	Unknown ProSe Application ID
3	UE authorisation failure
4	Unknown ProSe Application Code
5	Invalid MIC
6	Invalid Time Parameter
7	Invalid Message Format
8	Scope violation in ProSe Application ID
9-255	Unused

12.2.2.9 Discovery Key

This parameter is used to carry a Discovery Key allocated by the ProSe Function. This key is used by the UE to compute the MIC that is included in the PC5_DISCOVERY message.

12.2.2.10 Discovery Type

This parameter is used to indicate the type of ProSe direct discovery.

This parameter is coded as shown in figure 12.2.2.10.1 and table 12.2.2.10.1.

8	7	6	5	4	3	2	1	
Message type		0	0	0	0	Discovery model		octet 1
		spare						

Figure 12.2.2.10.1: Discovery Type parameter

Table 12.2.2.10.1: Discovery Type parameter

Message type value (octet 1)		
Bit		
8	**7**	
0	0	Reserved
0	1	PC5_DISCOVERY message for Open discovery
1	0	Reserved
1	1	Reserved
Discovery model value (octet 1)		
Bit		
2	1	
0	0	Reserved
0	1	Model A
1	0	Reserved
1	1	Reserved
Bits 3 to 6 of octet 1 are spare and shall be coded as zero.		

Editor's note: The coding of the Discovery Type parameter is still FFS.

12.2.2.11 MIC

This parameter is used to carry the MIC (Message Integrity Check) associated with the ProSe Application Code contained in a PC5_DISCOVERY message.

12.2.2.12 Discovery Filter

The elements in the Discovery Filter parameter are listed below.

- Filter ID: an identifier assigned by the ProSe Function, having local significance only to the ProSe Function, in order for it to identify the Discovery Filter it has created based on a ProSe Application ID;

- ProSe Application Code: This ProSe Application Code is used by a monitoring UE for full or partial matching of PC5_DISCOVERY messages received on the PC5 interface. Only one code is allowed in a Discovery Filter;

- ProSe Application Mask: a bitmask provided by the ProSe Function in order to allow the monitoring UE to perform full or partial matching of PC5_DISCOVERY messages received on the PC5 interface. Note that multiple Masks may be included in a Discovery Filter. The length of the mask is as same as the length of ProSe Application Code; and

- TTLTimer T4002: Time-to-live duration for which the specified Discovery Filter is valid, after which it shall not be used.

12.2.2.13 ProSe Application Masks

This parameter is used to carry a bitmask provided by the ProSe Function in order to allow the monitoring UE to perform full or partial matching of PC5_DISCOVERY messages received on the PC5 interface. Note that multiple Masks may be included in a Discovery Filter. The length of the mask is the same as the length of the ProSe Application Code.

12.2.2.14 TTL Timer T4002

This parameter is used to carry the value of TTL timer T4002 associated with a Discovery Filter. It is an integer representing the timer value in unit of minutes.

12.2.2.15 Filter ID

This parameter is used to identify the filter used by a monitoring UE to match the ProSe Application Code contained in a MATCH_REPORT message. It is an integer in the 0-65535 range.

12.2.2.16 Monitored PLMN ID

This parameter is used to indicate the PLMN ID of the PLMN in which the PC5_DISCOVERY message containing a ProSe Application Code for which there was a match event was received. It is coded as specified in 3GPP TS 23.003 [4].

12.2.2.17 VPLMN ID

This parameter is used to indicate the PLMN ID of the PLMN in which the requesting UE is registered. It is coded as specified in 3GPP TS 23.003 [4].

12.2.2.18 Time Parameter

This parameter is used to indicate the time parameter associated with the PC5_DISCOVERY message containing a ProSe Application code for which there was a match event.

Editor's note: The format of this parameter is to be decided by 3GPP TSG RAN WG2

12.2.2.19 Validity Timer T4004

This parameter is used to carry the value of Validity Timer T4004 associated with a ProSe Application Code for which there was a match event. It is an integer representing the timer value in unit of minutes.

12.2.2.20 Metadata Flag

This parameter is used to indicate whether the UE wishes to receive metadata information associated with the ProSe Application ID in the MATCH_REPORT_ACK from the ProSe Function. It is a Boolean value coded as follows:

False the UE does not wishes to receive metadata information associated with the ProSe Application ID in the MATCH_REPORT_ACK from the ProSe Function

True the UE wishes to receive metadata information associated with the ProSe Application ID in the MATCH_REPORT_ACK message from the ProSe Function

12.2.2.21 Metadata

This parameter is used to carry the metadata that is associated with the ProSe Application ID contained in the MATCH_REPORT_ACK message. The purpose of the metadata is to carry additional application-layer information associated with a particular ProSe Application ID. Examples of such information are postal address, phone number, URL etc. The contents of the metadata are out of scope of 3GPP.

Editor's note: The format of the metadata is FFS.

12.3 EPC-level ProSe discovery message formats

12.3.1 Data types format in XML schema

To exchange structured information over the transport protocol, XML text format/notation is introduced.

The corresponding XML data types for the data types used in ProSe messages are provided in table 12.3.1.

Table 12.3.1: Primitive or derived types for ProSe Parameter Type

ProSe Parameter Type	Type in XML Schema
Integer	decimal
String	string
Boolean	boolean
Binary	hexBinary
Date and Time	dateTime

For complex data types described in subclause 12.2.2, an XML "complexType" can be used.

Message construction shall be compliant with W3C REC-xmlschema-2-
20041028: "XML Schema Part 2: Datatypes" [7]

12.3.2 Information elements in EPC-level ProSe discovery messages

12.3.2.1 Transaction ID

This parameter is used to uniquely identify a PC3 Control Protocol for EPC-level ProSe discovery transaction when it is combined with other PC3 Control Protocol for EPC-level ProSe discovery transactions in the same transport message. The UE shall set this parameter to a new number for each outgoing new discovery request. The transaction ID is an integer in the 0-255 range.

12.3.2.2 UE Identity

This parameter is used to indicate the requesting UE's identity and is set to the IMSI. The IMSI consists of three parts:

- MCC: 3-digit integer;
- MNC: 2-digit integer; and
- MSIN: 10-digit or shorter integer.

12.3.2.3 Application Identity

This parameter is used to identify the particular application that triggers the APPLICATION_REGISTRATION_REQUEST message. The format of the Application Identity consists of two parts:

- OS ID: operating system identifier. The format of the OS ID is a Universally Unique IDentifier (UUID) as specified in IETF RFC 4122 [8]; and
- OS App ID: a string containing the OS specific application identifier.

NOTE: Further definition of the format of OS App ID is beyond the scope of this specification.

12.3.2.4 Application Layer User ID

This parameter is used to carry an Application Layer User ID that identifies the user in the context of specific application. It is encoded as a bit string.

12.3.2.5 PC3 EPC Control Protocol cause value

This parameter is used to indicate the particular reason why a UE_REGISTRATION_REQUEST or APPLICATION_REGISTRATION_REQUEST messages from the UE have been rejected by the ProSe Function. It is an integer in the 0-255 range encoded as follows:

0 Reserved

1 Invalid Application

2 UE authorisation failure

3 Invalid Message Format

4 Application not registered

5 Range class not allowed for this application

6 Proximity detection unlikely within requested time window

7 Targeted user not registered for this application

8 Proximity validation rejected by B-side

9 Application disabled temporarily

10-255 Unused

12.3.2.6 WLAN Link Layer ID

This parameter is used to carry WLAN link layer identifier.

12.3.2.7 EPC ProSe User ID

This parameter is used to carry an EPC ProSe User ID. It is a bit string coded as specified in 3GPP TS 23.003 [4].

12.3.2.8 Allowed Range

This parameter is used to carry set of range classes allowed for an application. It is an integer in the 0-255 range encoded as follows:

0 Reserved

1 50 m

2 100 m

3 200 m

4 500 m

5 1000 m

6-255 Unused

12.3.2.9 Time Window

This parameter is used to specify a time interval in minutes during which a proximity request is valid. The Time Window is an integer in the range of 1 – 1440 minutes.

12.3.2.10 Range Class

This parameter is used to carry a range class for a specific proximity request. It is selected from the set of allowed range classes carried in the Allowed Range parameter.

12.3.2.11 UE Location

This parameter is used to carry the UE location with the best known accuracy (e.g. Cell ID or geo-location coordinates). The UE Location is set to the cell identity part of the Evolved Cell Global Identifier, as described in 3GPP TS 36.331 [12] and obtained from the lower layers of the UE. The value of UE Location is coded as a hexabinary value with fixed length of 28 bits.

12.3.2.12 WLAN Indication

This parameter is used to carry an indication of whether the searching UE wishes to engage in WLAN direct discovery and communication subsequent to successful proximity detection. It is a Boolean value coded as follows:

False the searching UE does not wishe to engage in WLAN direct discovery and communication subsequent to successful proximity detection

True the searching UE wishes to engage in WLAN direct discovery and communication subsequent to successful proximity detection

12.3.2.13 Assistance Information

This parameter is used to carry information for expediting WLAN direct discovery and communication. The content of this parameter depends on the WLAN technology.

Wi-Fi Peer-to-Peer (P2P) specification [13] defines an architecture and set of protocols that facilitate direct discovery and communication using the IEEE 802.11 technology [14]. To assist WLAN direct discovery and communication as required by the Wi-Fi P2P technology, the Assistance Information includes the following parameters.

- SSID: The SSID to use for Wi-Fi P2P operation. To be compliant with the Wi-Fi P2P specification [13] the SSID should be in the form "DIRECT-ab" where a, b are two random characters;
- WLAN Secret Key: The pre-shared key to be used by UEs to secure their Wi-Fi P2P communication. This is used by UEs as the Pairwise Master Key (PMK);
- Group Owner indication: If set, the UE should implement the Group Owner (GO) functionality specified in the Wi-Fi P2P specification [13]. The UE implementing this functionality essentially becomes an AP that transmits Beacons with the P2P Information Element and accepts associations from other Wi-Fi P2P devices or from legacy Wi-Fi devices (those not implementing the Wi-Fi P2P functionality). If not set, the UE should behave as a Wi-Fi P2P client that attempts to discover and associate with a GO;
- P2P Device Address of self: This is the WLAN Link Layer ID to be used by UE to advertise itself. A UE implementing the Group Owner and indicates the WLAN Direct device from which the GO should accept WLAN association requests. Association requests from all other WLAN devices should be rejected by GO.
- P2P Device Address of peers: This is the WLAN Link Layer ID to be used by UE to discover peer UEs. A UE implementing the Group Owner should accept WLAN association requests only from devices that are in this list;
- Operation channel: The channel on which Wi-Fi P2P discovery and communication should take place; and
- Validity time: The time period during which the content provided in the assistance information is valid.

13 List of system parameters

13.1 General

The description of timers in table 13.2.1 and table 13.2.2 should be considered a brief summary. The complete descriptions of the timers are in the procedures defined in subclause 5 and subclause 6.

13.2 Timers of ProSe direct services procedures

Table 13.2.1: ProSe direct services timers – UE side

TIMER NUM.	TIMER VALUE	CAUSE OF START	NORMAL STOP	ON EXPIRY
T4000	NOTE 1	Upon receiving a ProSe Application Code with an associated T4000 timer in a DISCOVERY_RESPONSE message whose transaction ID contained in the <response-announce> element matches the value sent by the UE in a DISCOVERY_REQUEST message with the command set to "announce", as described in subclause 6.2.2.4.	Upon receiving a new T4000 timer value for the same ProSe Application Code or receiving a new Timer associated with a new ProSe Application Code for the same ProSe Application ID in a DISCOVERY_RESPONSE message. When the UE selects a new PLMN.	Stop announcing the associated ProSe Application Code over the PC5 interface and re-initiate the announce request procedure if the request from upper layers to announce the ProSe Application ID corresponding to the associated ProSe Application Code is still in place.
T4002	NOTE 2	Upon receiving a Discovery Filter with an associated T4002 timer in a DISCOVERY_RESPONSE message whose transaction ID contained in the <response-monitor> element matches the value sent by the UE in a DISCOVERY_REQUEST message with the command set to "monitor", as described in subclause 6.2.3.4.	Upon receiving a new T4002 timer value for the same Discovery Filter in a DISCOVERY_RESPONSE message.	Stop using the associated Discovery Filter for ProSe direct discovery monitoring over the PC5 interface and re-initiate the monitor request procedure, if the request from upper layers to monitor the ProSe Application ID corresponding to the associated Discovery Filter is still in place.
T4004	NOTE 3	Upon receiving a T4004 timer in a MATCH_REPORT_ACK message whose transaction ID contained in the <match-ack> element matches the value sent by the UE in a MATCH_REPORTmessage, as described in subclause 6.2.4.4.	Upon receiving a new T4004 timer value for the same ProSe Application Code in a MATCH_REPORT_ACK message.	The UE may inform the upper layers that the corresponding ProSe Application ID is not longer matched.
T4005	NOTE 4	Upon receiving a monitoring, announcing or communication policy for a given PLMN with an associated T4005 value in the ProSe service authorisation as described in subclause 5.1.3 and subclause 5.1.4.	When the service authorisation for the corresponding PLMN is revoked by the ProSe Function. Upon receiving a new T4005 timer value for the same operation (monitoring, announcing or communication) in the same PLMN.	Stop the monitoring, announcing or communication operation in the corresponding PLMN and re-initiate the service authorisation procedure if the UE wants to continue performing announcing, monitoring or communication in that PLMN.
NOTE 1: The value of this timer is provided by the ProSe Function during the announce request procedure. NOTE 2: The value of this timer is provided by the ProSe Function during the monitor request procedure. NOTE 3: The value of this timer is provided by the ProSe Function during the match report procedure. NOTE 4: The value of this timer is provided by the ProSe Function during service authorisation procedure.				

Table 13.2.2: ProSe direct services timers – ProSe Function side

TIMER NUM.	TIMER VALUE	CAUSE OF START	NORMAL STOP	ON EXPIRY
T4001	NOTE 1	Upon assigning a ProSe Application Code with an associated T4000 value to the UE, as described in subclause 6.2.2.3.	Upon receiving a new DISCOVERY_REQUEST message from the UE with the command set to "announce" for the same ProSe Application ID.	Delete the association between the UE, the requested ProSe Application ID and the corresponding ProSe Application Code allocated by the ProSe Function.
T4003	NOTE 2	Upon assigning a Discovery Filter with an associated T4002 value to the UE, as described in subclause 6.2.3.3.	Upon receiving a new DISCOVERY_REQUEST message from the UE with the command set to "monitor" for the same ProSe Application ID	Delete the association between the UE, the requested ProSe Application ID and the corresponding Discovery Filter allocated by the ProSe Function.
NOTE 1: The value of this timer is assigned by the ProSe Function during the announce request procedure. NOTE 2: The value of this timer is assigned by the ProSe Function during the monitor request procedure.				

NOTE: Multiple timers T4001 and T4003 can run simultaneously in the ProSe Function.

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2014-03					TS skeleton generated for submission at CT1#86bis	-	0.0.0
2014-04	CT1#86 bis				Implementation of C1-141576, C1-141623, C1-141533 and C1-141534.	0.0.0	0.1.0
2014-05	CT1#87				Implementation of C1-142123, C1-142152, C1-142153, C1-142187, C1-142188, C1-142189, C1-142190, C1-142191, C1-142192, C1-142195, C1-142198 and C1-142498.	0.1.0	0.2.0
2014-06	CT-64	CP-140279			Version 1.0.0 created for presentation for information to CT plenary	0.2.0	1.0.0
2014-07	CT1#88				Implementation of C1-142652, C1-142674, C1-142700, C1-142833, C1-142851, C1-142852, C1-142998, C1-143022, C1-143027, C1-143238, C1-143240, C1-143241, C1-143243, C1-143246, C1-143247, C1-143248, C1-143249, C1-143251, C1-143319, C1-143347, C1-143348, C1-143349, C1-143359 and C1-143360. Editorial corrections.	1.0.0	1.1.0
2014-09	CT-65	CP-140630			Version 2.0.0 created for presentation for approval to CT plenary	1.1.0	2.0.0
2014-09	CT-65	CP-140717			Plenary tdoc revised to include missing cover sheet	1.1.0	2.0.0
2014-09	Post CT-65				Version 12.0.0 created after approval	2.0.0	12.0.0

History

Document history		
V12.0.0	October 2014	Publication