INTERNATIONAL STANDARD

ISO/IEC 21559-3

First edition
2022-03

Telecommunications and information exchange between systems — Future network protocols and mechanisms —

Part 3:
Networking of everything

Télécommunications et échange d'informations entre systèmes — Futurs protocoles et mécanismes de réseau —

Partie 3: Réseautique universelle

Reference number
ISO/IEC 21559-3:2022(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword v
Introduction vi
1 Scope 1
2 Normative references 1
3 Terms and definitions 1
4 Abbreviated terms 4
5 FN-NoE functional description 5
5.1 General 5
5.2 Thing-user social community organization procedure 5
5.3 Thing-user social community maintenance procedure 6
5.4 Thing-user experiences sharing procedure 8
5.5 Coordinated thing-user discovery procedure 8
5.6 Proximal path establishment procedure 9
6 FN-NoE services 10
6.1 General 10
6.2 NoE SAP interface 11
6.2.1 Thing-user centric communication service primitive 11
6.2.2 Thing-user autonomous coordination 15
6.3 TSN SAP interface 15
6.3.1 Thing-user social community organization primitives 15
6.3.2 Thing-user social community join primitives 17
6.3.3 Thing-user social community maintenance primitives 18
6.3.4 Thing-user subscribe/publish primitives 19
6.4 PDN SAP interface 20
6.4.1 Thing-user discovery primitives 20
6.4.2 Proximal path management primitives 22
7 FN-NoE protocols 23
7.1 General 23
7.2 TSN protocols 24
7.2.1 Search a social community request message 24
7.2.2 Search a social community report message 25
7.2.3 Setup a new social group request message 26
7.2.4 Setup a new social group report message 26
7.2.5 Join a thing-user network request message 27
7.2.6 Join a thing-user network report message 27
7.2.7 Join a cluster request message 28
7.2.8 Join a cluster report message 29
7.2.9 Leave a TSN request message 29
7.2.10 Leave a TSN report message 30
7.2.11 Renew a coordinator request message 31
7.2.12 Renew a coordinator report message 31
7.2.13 Vote for a new coordinator request message 31
7.2.14 Vote for a new coordinator report message 32
7.2.15 Accept as a new coordinator message 33
7.2.16 Subscribe thing-users request message 33
7.2.17 Subscribe thing-users report message 33
7.2.18 Publish to subscriber report message 34
7.3 PDN protocols 34
7.3.1 Set proximity measure request message 34
7.3.2 Set proximity measure report message 35
7.3.3 Discover a peer request message 36
7.3.4 Discover a peer report message 36

7.3.5 Maintain proximal path to a peer request message 37
7.3.6 Maintain proximal path to a peer report message 37

Bibliography **39**

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

A list of all parts in the ISO 21559 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

ISO/IEC TR 29181 is a series of standards on Future Network (FN). The subject of ISO/IEC TR 29181-9 is Networking of Everything (NoE).

ISO/IEC 21558-3[1)] specifies the architecture of the FN-NoE to provide further advanced NoE services identified in ISO/IEC TR 29181-9.

This document focuses on the FN-NoE protocols and mechanisms, consisting of functional procedures, service interfaces, and protocols, to provide the thing-user centric communication service upon the FN-NoE architecture defined in ISO/IEC 21558-3.

[1)] Under preparation. Stage at the time of publication: ISO/IEC PRF 21558-3.

Telecommunications and information exchange between systems — Future network protocols and mechanisms —

Part 3: Networking of everything

1 Scope

This document specifies:

— functional procedures to provide thing-user social networking and proximity defined networking;

— service access interfaces and protocols to support the FN-NoE functional procedure.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 21558-3:2022, *Telecommunications and information exchange between systems — Future network architecture — Part 3: Networking of everything*

ISO/IEC/TR 29181-9:2017, *Information technology — Future Network — Problem statement and requirements — Part 9: Networking of everything*

3 Terms and definitions

For the purposes of this document, the terms and definitions specified in ISO/IEC 21558-3 and ISO/IEC TR 29181-9 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
object
intrinsic representation of an entity that is described at an appropriate level of abstraction in terms of its attributes and functions

[SOURCE: ISO/IEC TR 29181-9:2017, 3.12]

3.2
identifier
series of digits, characters and symbols or any other form of data used to identify subscriber(s), user(s), network element(s), function(s), network entity(ies) providing services/applications, or other entities (e.g. physical or logical objects)

[SOURCE: ISO/IEC TR 29181-9:2017, 3.7]

3.3
context
information that can be used to characterize the environment of a user

[SOURCE: ISO/IEC TR 29181-9:2017, 3.4]

3.4
thing
object (3.1) of the physical world (physical things) or of the information world (virtual thing), which is capable of being identified and integrated into communication networks

Note 1 to entry: Physical things are capable of being sensed, actuated and connected to things such as robots, goods and electrical equipment. Virtual things are capable of being stored, processed and accessed by things such as multimedia content and application software.

[SOURCE: ISO/IEC TR 29181-9:2017, 3.16]

3.5
collaborative work group
group of thing-users that can perform job planning, thing-user recruiting and coordination without human intervention

[SOURCE: ISO/IEC TR 29181-9:2017, 3.2]

3.6
proximity defined network
PDN
network configured among devices in close proximity, using conventional wired/wireless LAN or WAN technologies, which are in not only physically close proximity, but also closely related, or logically close proximity.

Note 1 to entry: PDN is an instantaneous network which is formed during the networking of everything.

[SOURCE: ISO/IEC TR 29181-9:2017, 3.13, modified — Note 1 to entry was added.]

3.7
Future Network for Networking of Everything
FN-NoE
network that is capable of providing *thing-user social networking* (3.12) and *thing-user centric communication* (3.13) service to the thing-users

[SOURCE: ISO/IEC 21558-3:2022, 3.8]

3.8
thing-user
thing that uses the *Future Network for Networking of Everything (FN-NoE)* (**3.7**) network service or the FN-NoE services provided by other things

Note 1 to entry: The thing-user in FN-NoE is defined as a concept corresponding to a human-user. Thing-user has an *NoE terminal* (3.11) function.

[SOURCE: ISO/IEC 21558-3:2022, 3.10, modified — Note 1 to entry was added.]

3.9
everything
equipment that is capable of performing Networking of Everything (NoE)

Note 1 to entry: "Everything" can be regarded as anything which can perform NoE in Future Network (FN). Everything is limited to the NoE related documents if there are no other comments.

[SOURCE: ISO/IEC 21558-3:2022, 3.11, modified — The last sentence of Note 1 to entry was added.]

3.10
Networking of Everything
NoE
process that is capable of providing *Future Network for Networking of Everything (FN-NoE)* (**3.7**) services

[SOURCE: ISO/IEC 21558-3:2022, 3.12]

3.11
Networking of Everything (NoE) terminal
thing that can perform the process in the network capable of providing *thing-user social networking* (3.12) and *thing-user centric communication* (3.13) service to the thing users

[SOURCE: ISO/IEC 21558-3:2022, 3.13]

3.12
thing-user social network
social network among *thing-users* (3.8) which automatically shares its capabilities, *context* (3.3), communicative motivation, experiences and intentions of collaboration for delivering the intelligent super-realistic service

Note 1 to entry: As the thing-user expands the social network, it may expand its knowledge.

Note 2 to entry: Thing-user social networking service can be a web application that thing-users use to build a social network provided by Networking of Everything (NoE). For web implementations, please see references [2], [3], [4], [5], and [6].

[SOURCE: ISO/IEC 21558-3:2022, 3.14, modified — The last sentence of Note 1 to entry was added.]

3.13
thing-user centric communication
process of conveying intended meanings from one *thing-user* (3.8) to another thing-user or thing-user group through the use of mutually understood language

[SOURCE: ISO/IEC 21558-3:2022, 3.15]

3.14
thing-user centric network
network that allows a *thing-user* (3.8) to discover another thing-user or thing-user group who understands its intention conveyed from the thing-user and supports the thing-user in achieving its mission

[SOURCE: ISO/IEC 21558-3:2022, 3.16]

3.15
Profile
all or some of the following information statements about a *thing-user* (3.8): (1) basic statements including Name, Identity, Address, URL, Account, Contract, Security; (2) motivation statements describing why the thing-user joins the social network; (3) mission statements describing what the thing-user plans to accomplish; and/or (4) capacity statements which describe its predications, knowledge, resources

Note 1 to entry: The statements in the Profile can be represented by semantic web language, e.g. RDF, RDFS or OWL, which can be standardized in W3C.

Note 2 to entry: In this document, semantic web technologies are used for explanation.

3.16
Networking of Everything (NoE) registry
resource which maintains and manages *thing-user* (3.8) social communities and *clusters* (3.17) by keeping a list of all or some of the following characteristics of social communities and social clusters: (1) types of social networks including category, place, means to connect with others, or a recommendation system linked to trust; (2) coordinator information and status of social networks; (3) physical characteristics, e.g. geographical locations, owner information, type, size; (4) network characteristics, e.g. name, address, URI, capacity including QoS or security

3.17
cluster
one of the social groups of *community* (3.18) based on cultural and geographical relations

Note 1 to entry: Each cluster shall be built on characteristics of category place, communication type, or system type. The cluster is organized and maintained by the cluster coordinator, who the cluster members elect.

3.18
community
one of the social groups of a *thing-user* (3.8) social network to share experiences and collaborate with other thing-users to achieve a specific mission.

Note 1 to entry: The thing-users are autonomous and have equal rights to govern the thing-user social community. A social community can be composed of one or more *thing-user social clusters* (3.17).

Note 2 to entry: The thing-user social community may have multi-tiered communities.

3.19
home thing-user
thing-user (3.8) that initially requests any action for a thing-user social network or a *thing-user centric network* (3.14)

4 Abbreviated terms

dTU-ID	destination Thing-User ID
FN	Future Network
FN-NoE	Future Network for Networking of Everything
ICT	Information and Communication Technology
ID	Identifier
NoE	Networking of Everything
PDN	Proximity Defined Network
SAP	Service Access Point
sTU-ID	source Thing-User ID
TSC	Thing-User Service Control
TSN	Thing-User Social Network
TU	Thing-User
TU-ID	Thing-User ID
TUn	Thing-User n

TUn-ID	Thing-User n ID
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

5 FN-NoE functional description

5.1 General

The thing-user representing a NoE terminal or a virtual switch in the FN-NoE organizes the thing-user social community to share experiences or collaborate on problem solving. The thing-user social community provides intelligent collaboration services to accomplish mission described by a human-user or by a thing-user. The thing-users configure the thing-user social community related to a specific domain task, share an experience with another thing-user who has joined the thing-user social community, and perform specific domain tasks autonomously without human operator intervention.

The thing-user social community has a procedure for consensus decision-making and for distributing shared information. By discovering coordinated peers, the thing-user will establish a proximal path. Then the thing-user is provided with the thing-user centric communication service.

5.2 Thing-user social community organization procedure

The thing-user of an NoE terminal or a virtual switch is initially specified with the Profile items including identities, capability skill sets and mission statement to be accomplished.

After initiation, the thing-user visits well-known resources, like FN-NoE registry, to check whether a thing-user social community exists for collaborating to accomplish the mission. If the thing-user finds a coordinator of the thing-user social community with whom to collaborate, the thing-user sends a JOIN message to the thing-user social community, as shown in Figure 2.

If the thing-user fails to find a thing-user social community, the thing-user organizes its own new thing-user social community as shown in Figure 1. In this case, the thing-user becomes a community coordinator. The TSN coordinator specifies the range of TSN and configures the structure of TSN.

The community coordinator searches the FN-NoE registry for the identities of the coordinator and mission statement. The first community coordinator serves as a cluster coordinator as well, until the cluster is renewed after joining another thing-user.

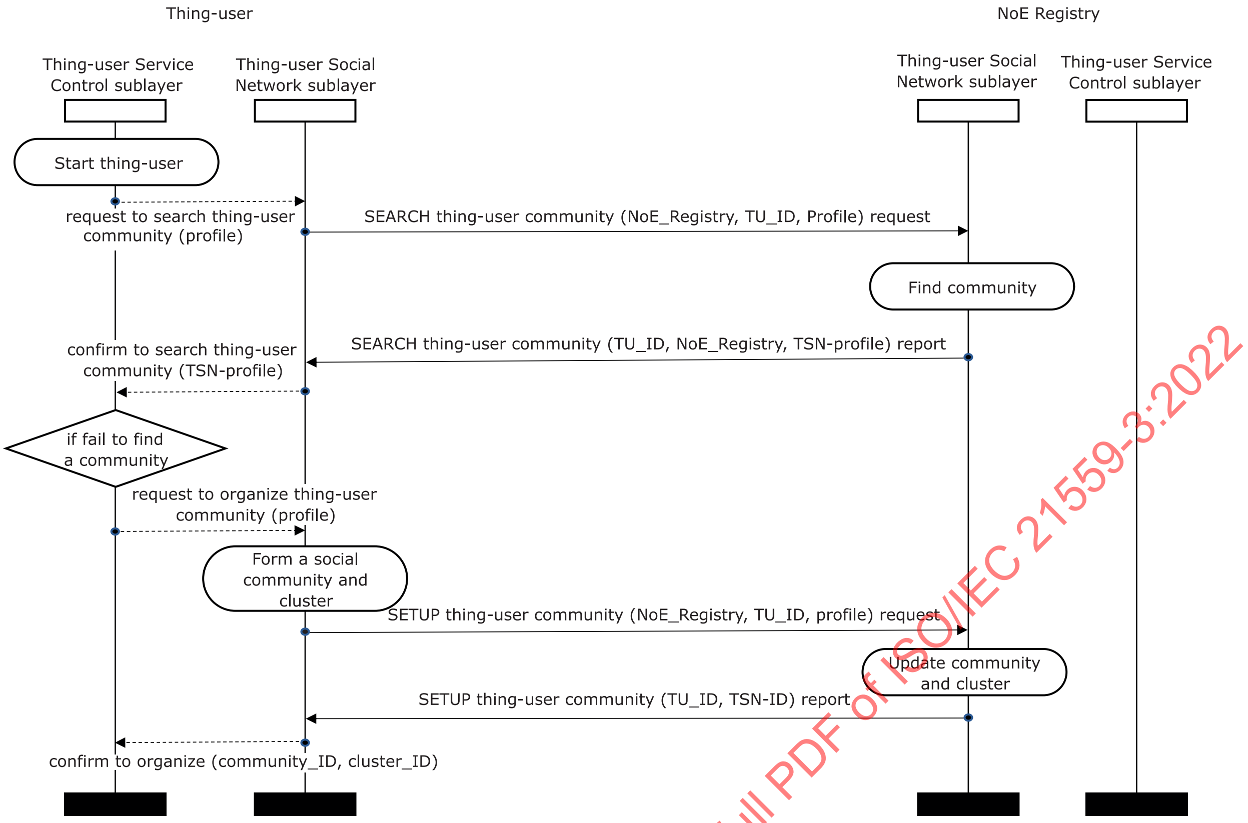
Thing-user
NoE Registry
Thing-user Service Control sublayer
Thing-user Social Network sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
Start thing-user
request to search thing-user community (profile)
SEARCH thing-user community (NoE_Registry, TU_ID, Profile) request
Find community
confirm to search thing-user community (TSN-profile)
SEARCH thing-user community (TU_ID, NoE_Registry, TSN-profile) report
if fail to find a community
request to organize thing-user community (profile)
Form a social community and cluster
SETUP thing-user community (NoE_Registry, TU_ID, profile) request
Update community and cluster
SETUP thing-user community (TU_ID, TSN-ID) report
confirm to organize (community_ID, cluster_ID)

Figure 1 — Sequence chart for thing-user social community organization

5.3 Thing-user social community maintenance procedure

If the thing-user finds a social community coordinator to collaborate on its mission, the thing-user sends a message to join the social community. The social community coordinator validates the thing-user and assigns a cluster to join. If the assigned cluster coordinator is not located in the social community coordinator, the social community coordinator requests the cluster coordinator to accept the thing-user.

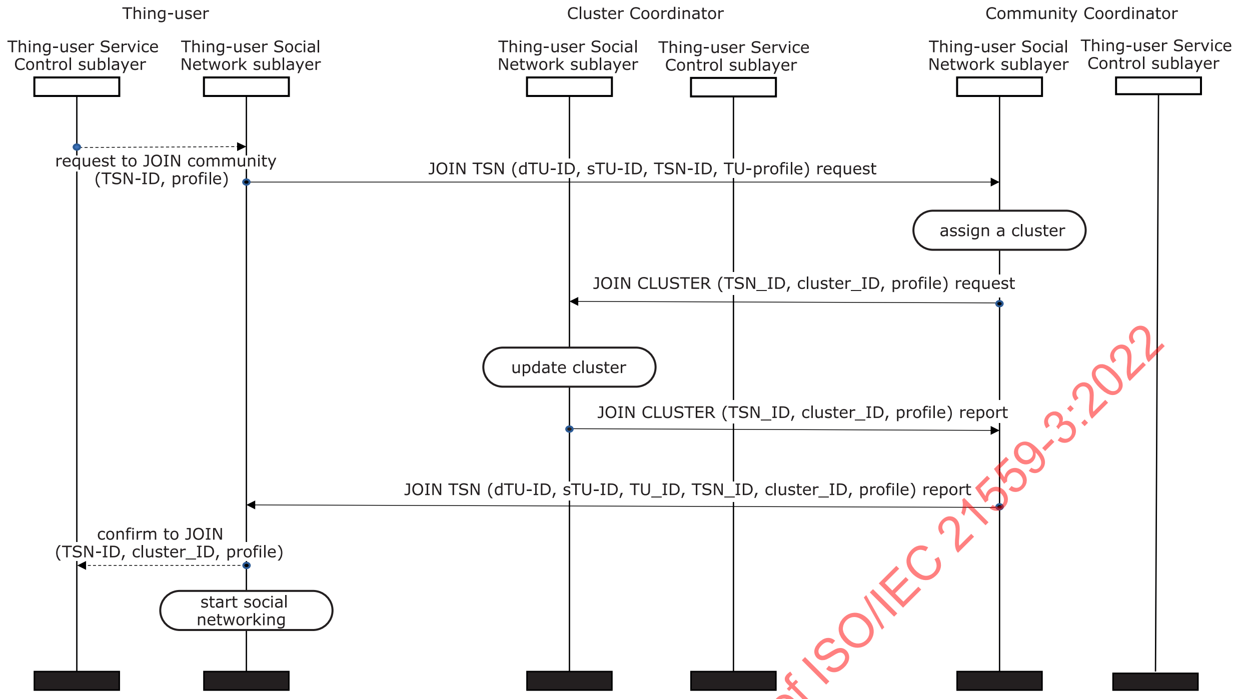
Thing-user
Cluster Coordinator
Community Coordinator
Thing-user Service Control sublayer
Thing-user Social Network sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
request to JOIN community (TSN-ID, profile)
JOIN TSN (dTU-ID, sTU-ID, TSN-ID, TU-profile) request
assign a cluster
JOIN CLUSTER (TSN_ID, cluster_ID, profile) request
update cluster
JOIN CLUSTER (TSN_ID, cluster_ID, profile) report
JOIN TSN (dTU-ID, sTU-ID, TU_ID, TSN_ID, cluster_ID, profile) report
confirm to JOIN (TSN-ID, cluster_ID, profile)
start social networking

Figure 2 — Sequence chart for thing-user social community join

The community coordinator maintains the clusters and their coordinators. The periodic election of a cluster coordinator renews the clusters of a social community. Similarly, the periodic election of a community tier coordinator renews the community tier of a social community. The incumbent cluster coordinator collects the ballots from the thing-users and selects a new coordinator and reports the voting result to the new coordinator. The newly elected coordinator announces the voting results to other thing-users. The incumbent cluster coordinator reports to the community coordinator that a new cluster coordinator was elected.

When a thing-user leaves a social community, the cluster coordinator updates the list of cluster members. If the departing thing-user is the last remaining thing-user of the social community, the thing-user should check out of the FN-NoE registry.

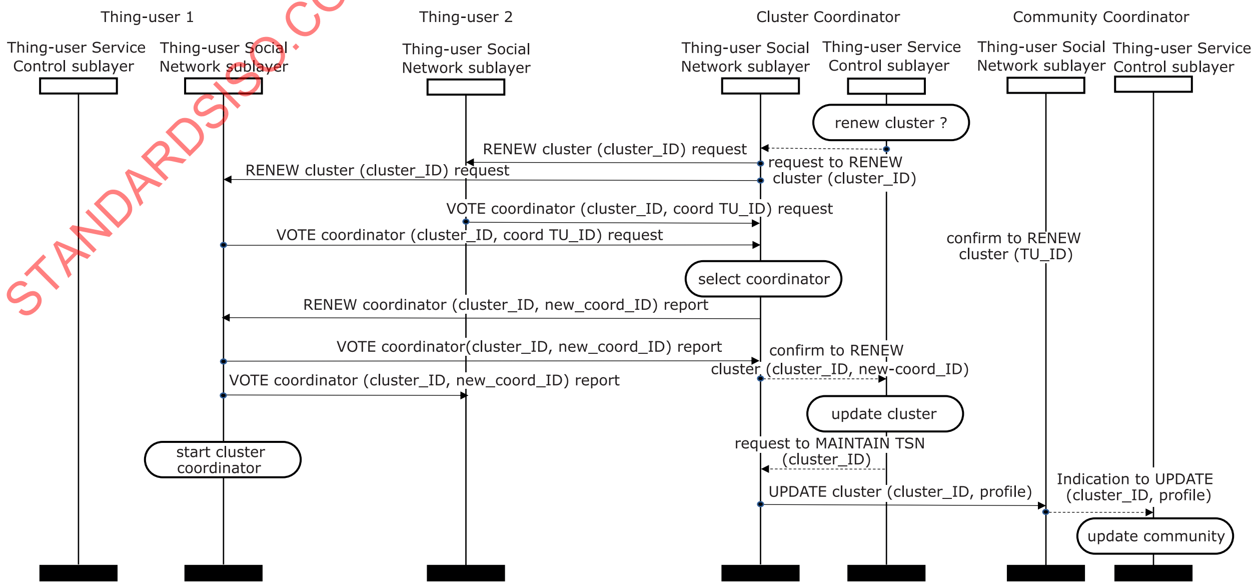
Thing-user 1
Thing-user 2
Cluster Coordinator
Community Coordinator
Thing-user Service Control sublayer
Thing-user Social Network sublayer
Thing-user Social Network sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
renew cluster ?
RENEW cluster (cluster_ID) request
request to RENEW cluster (cluster_ID)
RENEW cluster (cluster_ID) request
VOTE coordinator (cluster_ID, coord TU_ID) request
VOTE coordinator (cluster_ID, coord TU_ID) request
confirm to RENEW cluster (TU_ID)
select coordinator
RENEW coordinator (cluster_ID, new_coord_ID) report
VOTE coordinator(cluster_ID, new_coord_ID) report
confirm to RENEW cluster (cluster_ID, new-coord_ID)
VOTE coordinator (cluster_ID, new_coord_ID) report
update cluster
request to MAINTAIN TSN (cluster_ID)
start cluster coordinator
UPDATE cluster (cluster_ID, profile)
Indication to UPDATE (cluster_ID, profile)
update community

Figure 3 — Sequence chart for thing-user social community renewal

5.4 Thing-user experiences sharing procedure

The experience of a thing-user is captured and shared with the thing-user social community in a distributed manner. A cluster member may subscribe to an experience report from a certain cluster member who publishes a captured experience to the subscribed thing-user according to its own schedule.

The subscription request from a thing-user is reviewed and accepted based on the evaluation of the subscribing thing-user's social capital.

The subscription and publishing of experiences is managed at the cluster level.

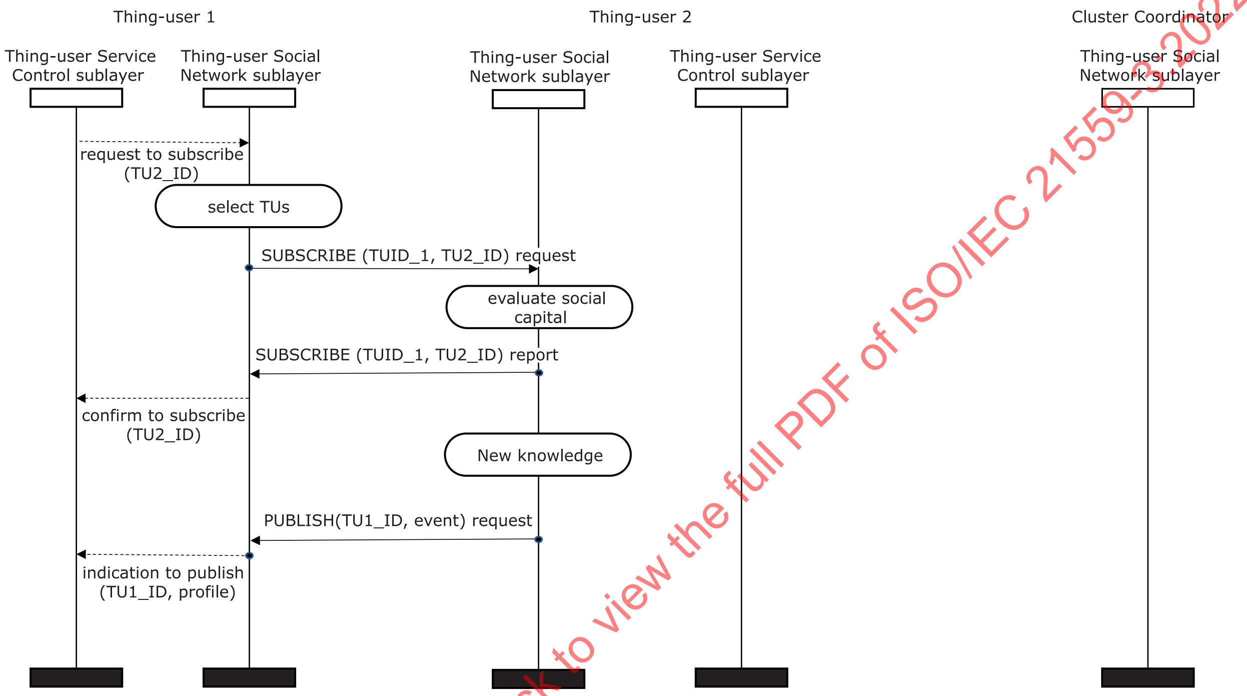
Thing-user 1
Thing-user 2
Cluster Coordinator
Thing-user Service Control sublayer
Thing-user Social Network sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
Thing-user Social Network sublayer
request to subscribe (TU2_ID)
select TUs
SUBSCRIBE (TUID_1, TU2_ID) request
evaluate social capital
SUBSCRIBE (TUID_1, TU2_ID) report
confirm to subscribe (TU2_ID)
New knowledge
PUBLISH(TU1_ID, event) request
indication to publish (TU1_ID, profile)

Figure 4 — Sequence chart for thing-user experience sharing

5.5 Coordinated thing-user discovery procedure

The thing-user starts to search a coordinated peer or peer group to help solve the problem of the thing-user described in the problem statement. The measure of proximity to a coordinated peer is defined in the motive statement, mission statement, or capacity statement in the profile. If the thing-user does not have enough information to specify the proximity measure, it may request the measure to the cluster coordinator or the TSN coordinator.

The thing-user checks the proximity to subscribed thing-users in the cluster. If the thing-user finds an appropriate peer thing-user, it sends a request for collaboration to solve a problem. If the thing-user receives an acceptance of the collaboration request, the thing-user requests to establish the proximal path to the coordinated thing-user.

If the thing-user fails to find a coordinated thing-user in the cluster, the thing-user starts to search for a peer in another cluster in the same level of community tier. If the thing-user fails to find a coordinated thing-user in the community tier In this case, the thing-user attempts to search for a peer in clusters in the higher tier of community until it discovers a peer.

According to the profile, the proximal path will be established either to the single coordinated thing-user or the sequentially ordered coordinated thing-users may be established.

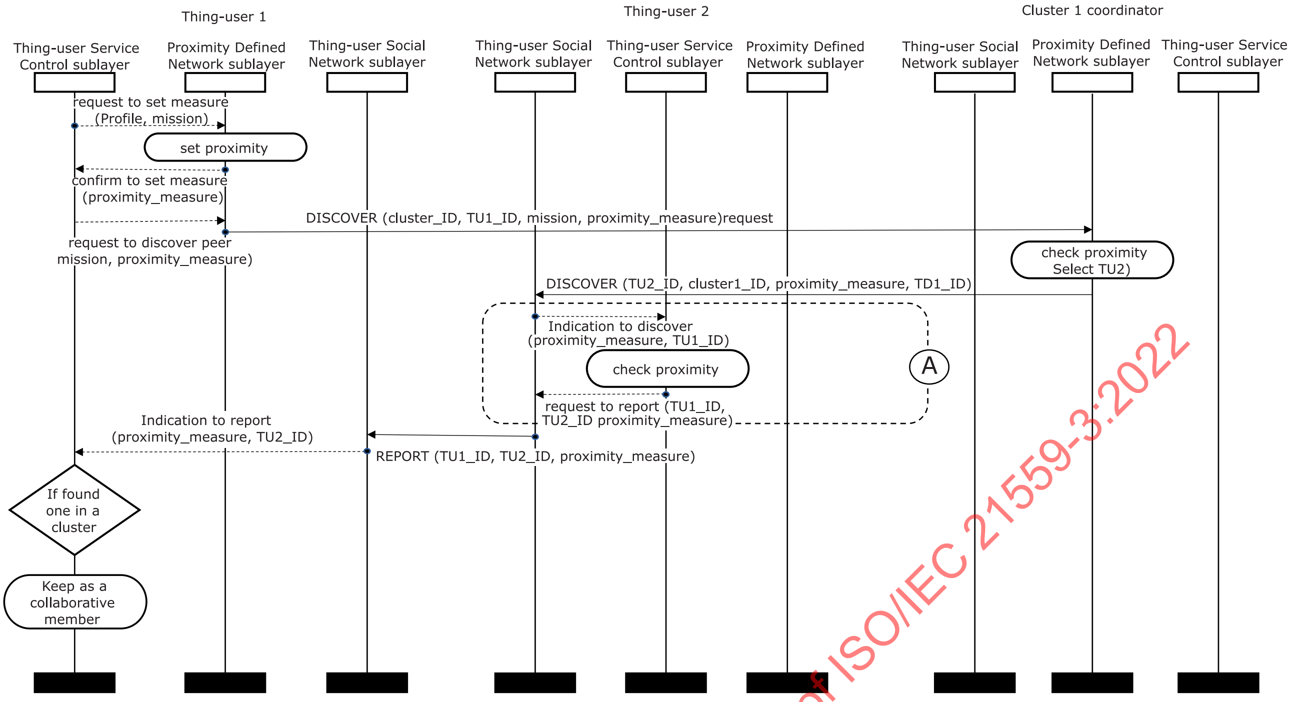
Thing-user 1
Thing-user 2
Cluster 1 coordinator
Thing-user Service Control sublayer
Proximity Defined Network sublayer
Thing-user Social Network sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
Proximity Defined Network sublayer
Thing-user Social Network sublayer
Proximity Defined Network sublayer
Thing-user Service Control sublayer
request to set measure (Profile, mission)
set proximity
confirm to set measure (proximity_measure)
DISCOVER (cluster_ID, TU1_ID, mission, proximity_measure)request
request to discover peer mission, proximity_measure)
check proximity Select TU2)
DISCOVER (TU2_ID, cluster1_ID, proximity_measure, TD1_ID)
Indication to discover (proximity_measure, TU1_ID)
check proximity
A
request to report (TU1_ID, TU2_ID proximity_measure)
Indication to report (proximity_measure, TU2_ID)
REPORT (TU1_ID, TU2_ID, proximity_measure)
If found one in a cluster
Keep as a collaborative member

Figure 5 — Sequence chart for thing-user discovery in case no appropriate thing-user was found

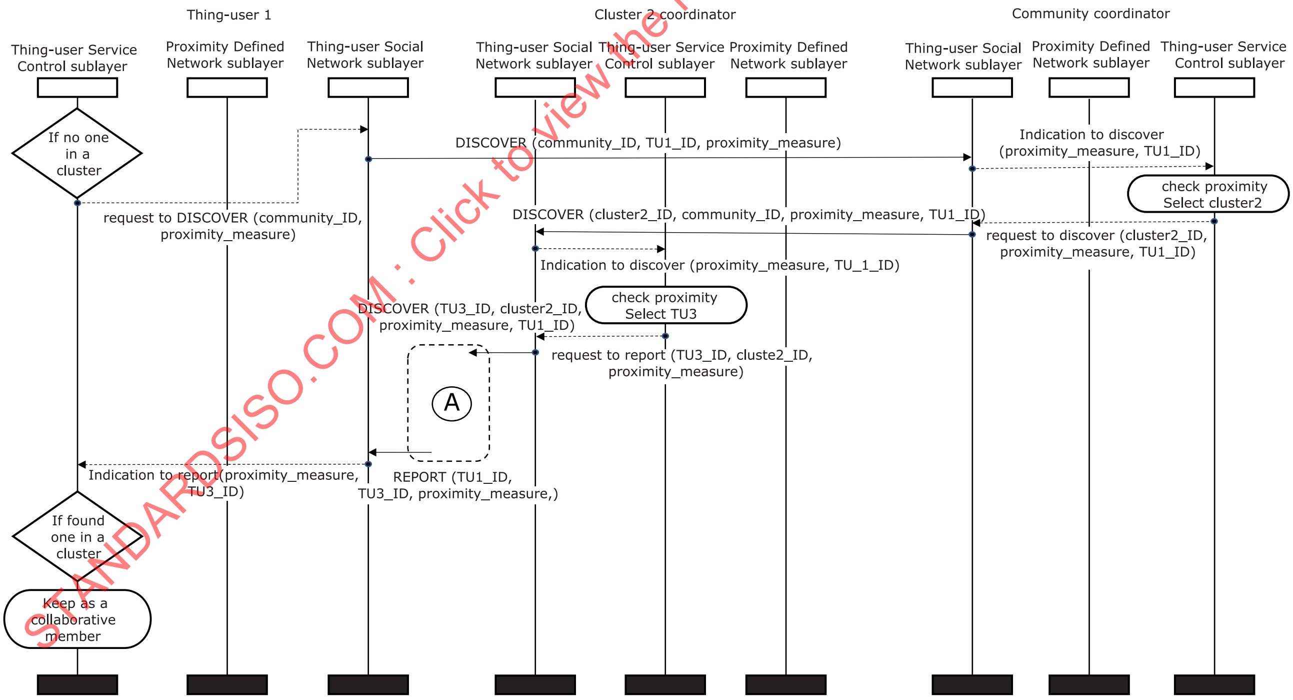
Thing-user 1
Cluster 2 coordinator
Community coordinator
Thing-user Service Control sublayer
Proximity Defined Network sublayer
Thing-user Social Network sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
Proximity Defined Network sublayer
Thing-user Social Network sublayer
Proximity Defined Network sublayer
Thing-user Service Control sublayer
If no one in a cluster
DISCOVER (community_ID, TU1_ID, proximity_measure)
Indication to discover (proximity_measure, TU1_ID)
check proximity Select cluster2
request to DISCOVER (community_ID, proximity_measure)
DISCOVER (cluster2_ID, community_ID, proximity_measure, TU1_ID)
request to discover (cluster2_ID, proximity_measure, TU1_ID)
Indication to discover (proximity_measure, TU_1_ID)
check proximity Select TU3
DISCOVER (TU3_ID, cluster2_ID, proximity_measure, TU1_ID)
request to report (TU3_ID, cluste2_ID, proximity_measure)
A
Indication to report(proximity_measure, TU3_ID)
REPORT (TU1_ID, TU3_ID, proximity_measure,)
If found one in a cluster
Keep as a collaborative member

Figure 6 — Sequence chart for thing-user discovery in case one or more appropriate thing-user(s) were found

5.6 Proximal path establishment procedure

To accomplish a mission collaboratively, when a thing-user finds a thing-user or a thing-user group belonging to the coordinated peers, the thing-user establishes proximal paths between them depending on the type of collaboration group and the characteristics of collaborative communication. The type of collaboration group can be peer-to-peer, peer-to-multi-peer, or multi-peer-to-multi-peer.

The thing-user who requests a search for the peers becomes the manager of the proximal paths between peers. The manager of the proximal path has a responsibility to establish, maintain and release the proximal paths. The manager of the proximal path monitors the status of the proximal path.

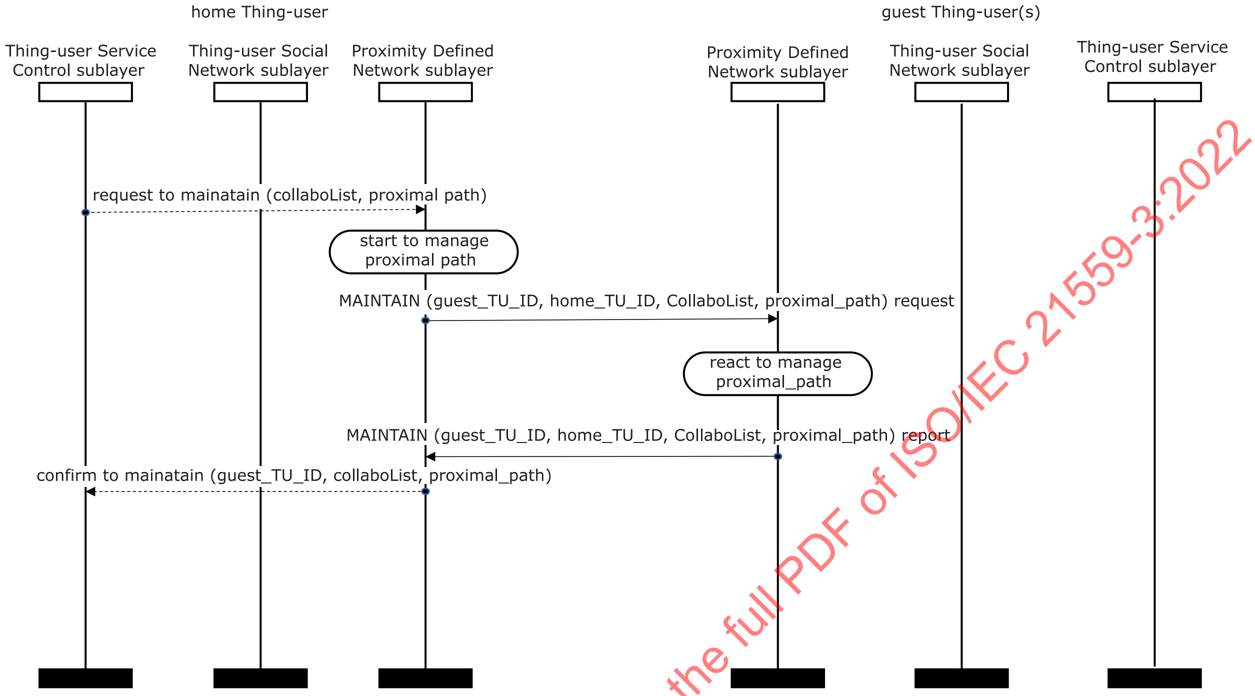
home Thing-user
guest Thing-user(s)
Thing-user Service Control sublayer
Thing-user Social Network sublayer
Proximity Defined Network sublayer
Proximity Defined Network sublayer
Thing-user Social Network sublayer
Thing-user Service Control sublayer
request to mainatain (collaboList, proximal path)
start to manage proximal path
MAINTAIN (guest_TU_ID, home_TU_ID, CollaboList, proximal_path) request
react to manage proximal_path
MAINTAIN (guest_TU_ID, home_TU_ID, CollaboList, proximal_path) report
confirm to mainatain (guest_TU_ID, collaboList, proximal_path)

Figure 7 — Sequence chart for proximal path establishment

6 FN-NoE services

6.1 General

The coordinated networking layer consists of a thing-user service control (TSC) sublayer, thing-user social network (TSN) sublayer, and proximity defined network (PDN) sublayer, as shown in Figure 8.

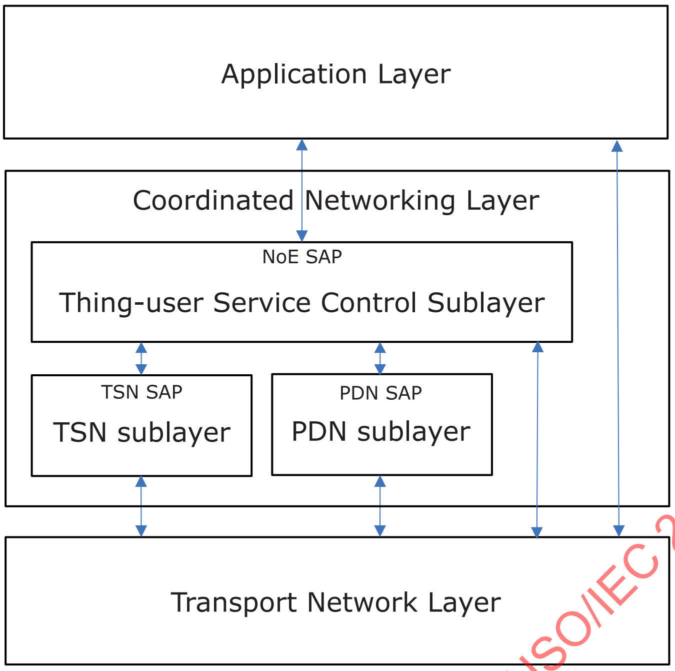

Figure 8 — FN-NoE terminal architecture

6.2 NoE SAP interface

The TSC sublayer provides NoE services to the application service layer through the NoE service access point (NoE SAP). This sublayer maintains the profile of the thing-user and controls the execution of the thing-user centric communication service.

6.2.1 Thing-user centric communication service primitive

The thing-user uses the thing-user centric communication service primitives to manage the thing-user profile, maintain the thing-user social community to which it subscribed, and discover a peer thing-user or a thing-user group that can accomplish a mission.

6.2.1.1 Managing thing-user profile

The MANAGE_TU_REQUEST and the MANAGE_TU_CONFIRM primitive are used to get the thing-user profile or to set the thing-user profile. The structure of the thing-user profile is defined by a thing-user. The name of an object in the thing-user profile shall be registered in the vocabulary set of the NoE Registry.

The thing-user profile may consist of identities, thing-user descriptions, capabilities, mission statements, social group descriptions and status of the thing-user. As identities, a thing-user may use combined objects of the profile: name, location, URI, thing-user description objects, capability objects and social group description objects. The thing-user description objects may consist of mobility, communication access, processing, power, thing-user motivation, and thing-user limitation. The thing-user capability description objects may consist of sensing capability skill sets and actuating capability skill sets. The social group description objects may consist of social group identity, social group structure, role in a social group, social group limitation and trustworthiness metrics.

NOTE The specification of the thing-user profile is outside the scope of this document. The object of the thing-user profile can be described by semantic web language. The structure of the thing-user profile depends on the domain ontologies related to the mission of the thing-user. In this document, the thing-user profile shall be written in JSON text [1].

The MANAGE_TU_REQUEST primitive shall have two attributes: action and profile.

- MANAGE_TU_REQUEST (action, profile)
  - action: "GET," "SET"
  - profile: JSON text

The MANAGE_TU_CONFIRM primitive shall have the following attributes: result, profile, and reason.

- MANAGE_TU_CONFIRM (result, profile, reason)
  - result: "SUCCESS," "FAIL"
  - profile: JSON text
  - reason: error comment

Upon receiving the MANAGE_TU_REQUEST primitive, the TSC sublayer starts to manage the thing-user profile.

When the action is set to "GET" and the profile attribute is a blank object, the TSC sublayer shall respond with the MANAGE_TU_CONFIRM by setting the result attribute to "SUCCESS" and the profile attribute to the entire object of the thing-user profile. If the profile attribute indicates the list of selected objects, the TSC sublayer shall respond with the MANAGE_TU_CONFIRM, by setting the result attribute to "SUCCESS" and the profile attribute to the requested objects of the thing-user profile.

If the TSC sublayer is inactive or fails to identify the object, the TSC sublayer shall respond with MANAGE_TU_CONFIRM, by setting the result attribute as "FAIL" and the reason attribute to error comment using the words registered in the vocabulary set of the NoE Registry.

When the action is set to "SET," the TSC sublayer identifies the objects of the profile attribute. If the object is matched to the stored thing-user profile, the TSC sublayer changes the object of the thing-user profile with the profile attribute and shall respond with MANAGE_TU_CONFIRM, by setting the result attribute to "SUCCESS" and the profile attribute to the changed objects of the thing-user profile.

If the object is not found, the TSC sublayer shall respond with MANAGE_TU_CONFIRM, by setting the result attribute to "FAIL" and the reason attribute to error comment.

6.2.1.2 Maintain thing-user social network (TSN)

The MANAGE_TSN_REQUEST and the MANAGE_TSN_CONFIRM primitives are used to maintain the TSNs.

The MANAGE_TSN_REQUEST primitive shall have two attributes: action and profile.

- MANAGE_TSN_REQUEST (action, profile)
  - action: "GET," "JOIN," "LEAVE," "MAINTAIN"
  - profile: JSON text

The MANAGE_TSN_CONFIRM primitive shall have the following attributes: result, profile, and reason.

- MANAGE_TSN_CONFIRM (result, profile, reason)
  - result: "SUCCESS," "FAIL"
  - profile: JSON text
  - reason: error comment

Upon receiving the MANAGE_TSN_REQUEST primitive, the TSC sublayer starts to manage the TSN.

When the action attribute is set to "GET," and if the profile attribute indicates the list of the selected social group, the TSC sublayer shall respond with MANAGE_TSN_CONFIRM by setting the result attribute to "SUCCESS" and the profile attribute to the requested social group description objects of the thing-user profile.

If the profile attribute is a blank object; In this case, the TSC sublayer shall respond with MANAGE_TSN_CONFIRM by setting the result attribute to "SUCCESS" and the profile attribute to the entire social group description that the thing-user joined.

If the thing-user is the community coordinator or the cluster coordinator, the structure of the community or the structure of the cluster is contained in the profile attribute.

If the TSC sublayer fails to identify the social group or the TSN sublayer is not active; In this case, the TSC sublayer shall respond with MANAGE_TSN_CONFIRM by setting the result attribute to "FAIL" and the reason attribute to error comment text.

When the action attribute is set to "JOIN," and if the profile attribute is not a blank object, the TSC sublayer checks whether the thing-user already joined the TSN indicated in the profile attribute.

If the thing-user already joined, the TSC sublayer shall respond with MANAGE_TSN_CONFIRM by setting the result attribute to "SUCCESS" and the profile attribute to the structure of the TSN joined.

If the thing-user did not already join, or if the profile attribute is a blank object, the TSC sublayer shall search for a TSN by using the SEARCH_TSN_REQUEST primitive, as described in 6.3.1.1.

If the TSN sublayer reports to the TSC sublayer that it found a social group, the TSC sublayer shall join the social group using the JOIN_TSN_REQUEST primitive, as described in 6.3.2.1.

If the TSN sublayer reports to the TSC sublayer that it did not find any social group, the TSC sublayer shall organize a social group using the SETUP_TSN_REQUEST primitive, as described in 6.3.1.2.

When the TSN has social group organization, the TSC sublayer shall respond with MANAGE_TSN_CONFIRM by setting the result attribute to "SUCCESS" and the profile attribute to the newly formed social group description objects.

If the thing-user fails to join a social group, the TSC sublayer shall respond with MANAGE_TSN_CONFIRM by setting the result attribute to "FAIL" and the reason attribute to the error comment text.

When the action attribute is set to "LEAVE," and the profile attribute is not an empty object, the TSC sublayer checks whether the thing-user still joining the TSN indicated in the profile attribute or not.

If the thing-user is still joining, the TSC sublayer shall leave a social community using the LEAVE_TSN_REQUEST primitive, as described in 6.3.2.2.

Upon receiving the LEAVE_TSN_CONFIRM primitive with the result attribute set to "SUCCESS," the TSC sublayer shall confirm using MANAGE_TSN_CONFIRM with the result attribute set to "SUCCESS."

If the thing-user did not join, the TSC sublayer confirms with MANAGE_TSN_CONFIRM by setting the result attribute to "FAIL" and the reason attribute to the error comment text.

When the action attribute is set to "MAINTAIN," and if the profile attribute is not a blank object, the TSC sublayer checks whether the thing-user is the coordinator of the TSN.

If the thing-user is the coordinator of the TSN or the coordinator of the cluster, the TSC sublayer shall maintain the social group using the MAINTAIN_TSN_REQUEST primitive by setting the action attribute to "UPDATE," "RENEW_TSN" or "RENEW_CLUSTER," as described in 6.3.3.1.

Upon receiving the MAINTAIN_TSN_CONFIRM primitive with the result attribute set to "SUCCESS," the TSC sub-layer shall confirm using MANAGE_TSN_CONFIRM with the result attribute set to "SUCCESS."

If the thing-user is not the coordinator, the TSC sublayer shall respond with MANAGE_TSN_CONFIRM by setting the result attribute to "FAIL" and the reason attribute to the error comment text.

6.2.1.3 Discover peer thing-user

The MANAGE_PEER_REQUEST and the MANAGE_PEER_CONFIRM primitives are used to discover a peer thing-user or peer thing-user group that collaborates to accomplish a mission by establishing proximal paths to the thing-user.

The MANAGE_PEER_REQUEST primitive shall have the following attributes: action, profile, and path ID.

- MANAGE_PEER_REQUEST (action, profile, path-ID)
  - action: "DISCOVER," "MAINTAIN," "RELEASE"
  - profile: JSON text
  - path-ID: proximal path ID

The MANAGE_PEER_CONFIRM primitive shall have the following attributes: result, path-ID, peer-ID, TU-profile, and reason.

- MANAGE_PEER_CONFIRM (result, path ID, peer ID, TU-profile, reason)
  - result: "SUCCESS," "FAIL"
  - path-ID: proximal path ID
  - peer-ID: a thing-user ID or the ID list of thing-users who collaborate
  - TU-profile: profile of the peer thing-user who collaborates
  - reason: error comment

Upon receiving the MANAGE_PEER_REQUEST primitive, the TSC sublayer starts to manage the peer thing-user.

When the action attribute is set to "DISCOVER," the TSC sublayer shall discover a peer thing-user or peer thing-user group according to the mission statement, thing-user description, and social group description objects included in the profile attribute.

The TSC sublayer shall specify the proximity measure by using the SET_PROXIMITY_REQUEST primitive, as described in 6.4.1.1.

When the TSC sublayer is reported to have specified a proximity measure successfully by receiving the SET_PROXIMITY_CONFIRM primitive with the result as attribute to "SUCCESS," the TSC sublayer shall start to discover a peer thing-user or peer thing-user group by using the DSICOVER_PEER_REQUEST primitive, as described in 6.4.1.2.

Upon receiving the DISCOVER_PEER_CONFIRM primitive, the TSC sublayer completes peer discovery by setting the resulting attribute to "SUCCESS," and setting the profile attribute to the profile of the discovered peer thing-user or the profile list of peer thing-users. If the TSC sublayer cannot find the peer, the TSC sublayer shall respond with MANAGE_PEER_CONFIRM by setting the result attribute set to "FAIL" and the reason attribute to the error annotation text.

When the action attribute is set to "MAINTAIN," the TSC sublayer shall maintain a proximal path and peer thing-user by using the MAINTAIN_PEER_REQUEST primitives by setting the action attribute to "HEARTBEAT," as described in 6.4.2.1.

When the action is set to "RELEASE," the TSC sublayer shall release a proximal path by using the MAINTAIN_PEER_REQUEST primitives by setting the action attribute to "RELEASE," as described in 6.4.2.1.

6.2.2 Thing-user autonomous coordination

The TSC sublayer provides autonomous coordination services to the thing-user.

The TSC sublayer will autonomously manage the thing-user profile, maintain the thing-user social community, discover a peer thing-user, and share the experiences by using the TSN SAP primitives and/or the PDN SAP primitives.

6.3 TSN SAP interface

The TSN sublayer provides the social network service to the Thing-user Service Control sublayer through the TSN SAP. This sublayer provides the thing-user social community organizing service, the thing-user experience sharing service and the thing-user collaboration service.

6.3.1 Thing-user social community organization primitives

The TSC sublayer uses the thing-user social community organization primitives for searching the thing-user social community to share experiences or for organizing a new social group.

6.3.1.1 Searching thing-user social community

The SEARCH_TSN_REQUEST primitive and the SEARCH_TSN_CONFIRM are used to search a TSN, which will be used to share the experiences for the mission accomplishment.

The SEARCH_TSN_REQUEST primitive shall have the following attribute: profile.

- SEARCH_TSN_REQUEST (profile)
  - profile: thing-user's profile

The SEARCH_TSN_CONFIRM primitive shall have the following attributes: result, TSN-ID, TSN-profile, and reason.

- SEARCH_TSN_CONFIRM (result, TSN-profile, reason)
  - result: "SUCCESS," "FAIL"
  - TSN-ID: identifier of TSN discovered
  - TSN-profile: social group description objects
  - reason: error comment

Upon receiving the SEARCH_TSN_REQUEST primitive, the TSN sublayer of the home thing-user starts to search a social community and to match up a TSN that is closely related to accomplishing the mission stated in the profile attribute.

The TSN sublayer of the home thing-user shall send the SearchTSN request message to the NoE Registry first, as described in 7.2.1.

When the TSN sublayer of the NoE Registry receives the SearchTSN request message, the TSN sublayer of the NoE Registry looks up the social community database and finds a TSN related to the mission of the home thing-user.

If the TSN sublayer finds a TSN, the TSN sublayer shall respond with the SearchTSN report message containing a TSN-profile to the home thing-user, as described in 7.2.2.

If the TSN sublayer of the NoE Registry fails to find a TSN to collaborate with the home thing-user, the TSN sublayer shall respond with the SearchTSN report message containing a blank profile object and its reason.

When the TSN sublayer of the home thing-user receives the SearchTSN report message, the TSN sublayer responds with the SEARCH_TSN_CONFIRM primitive by setting the TSN-profile attribute to the TSN-profile field of the SearchTSN report.

If the TSN sublayer of the home thing-user receives a blank object in the TSN-profile, the TSN sublayer responds with the SEARCH_TSN_CONFIRM primitive by setting the result attribute to “FAIL” and the reason attribute to the reason field of the SearchTSN report message.

6.3.1.2 Setup a social community and new cluster

The SETUP_TSN_REQUEST and the SETUP_TSN_CONFIRM primitives are used to form a TSN in which users may join based on their capabilities and experiences.

The SETUP_TSN_REQUEST primitive shall have the following attribute: profile.

- SETUP_TSN_REQUEST (profile)
  - profile: thing-user’s profile

The SETUP_TSN_CONFIRM primitive shall have the following attributes: result, TSN-ID, TSN-profile, and reason.

- SETUP_TSN_CONFIRM (result, TSN-ID, TSN-profile, reason)
  - result: “SUCCESS,” “FAIL”
  - TSN-ID: identifier of TSN registered
  - TSN-profile: social group description objects of profile
  - reason: error comment

Upon receiving the SETUP_TSN_REQUEST primitive, the TSN sublayer of the home thing-user organizes a thing-user social network and designates the TSN coordinator.

The range of the TSN is specified according to the mission statement object of the profile attribute. The mission statement may be ontology written in semantic web language, or keywords described by using the words registered in the vocabulary set of the NoE Registry.

The identifier of the TSN may be the combined objects of the profile attribute and may be written as serialized JSON text.

The TSN coordinator specifies the range of TSN and assigns the TSN-ID. The TSN coordinator configures the structure of TSN by forming a cluster or disbanding a cluster. The TSN coordinator maintains the structure of the TSN.

After designating the TSN coordinator and configuring the TSN ID and structure of the TSN, the TSN sublayer shall report new TSN establishment by sending the SetupTSN request message containing the social group description profile to the NoE Registry, as described in 7.2.3.

When the TSN sublayer of the NoE Registry receives the SetupTSN request message, the TSN sublayer shall validate the new social group configuration, update the social community database and respond with the SetupTSN report message containing the TSN-ID and result, as described in 7.2.4.

When the TSN sublayer of the home thing-user receives the SetupTSN report message, the TSN sublayer checks the TSN-ID field. If it is the TSN ID requesting to register, the TSN sublayer lets the TSN coordinator form the first cluster of the TSN and starts to maintain the TSN.

The TSN sublayer shall confirm the TSN establishment to the TSC sublayer using the SETUP_TSN_CONFIRM primitive with the TSN-ID attribute and the TSN-profile attribute.

If the NoE Registry refused to register the TSN, the TSN sublayer of the home thing-user releases the TSN coordinator and reports to the TSC sublayer using the SETUP_TSN_CONFIRM primitive by setting the result attribute to “FAIL” and reason attribute to error comment text.

6.3.2 Thing-user social community join primitives

The TSC sublayer uses the thing-user social community join primitives for the thing-user to join a TSN.

6.3.2.1 Joining a TSN and a cluster

The JOIN_TSN_REQUEST primitive is used for a thing-user to join a TSN that is selected by the NoE Registry or the TSC sublayer of the TSN coordinator.

The JOIN_TSN_REQUEST primitive shall have the following attributes: TSN-ID and TU-profile.

- JOIN_TSN_REQUEST (TSN-ID, TU-profile)
  - TSN-ID: identifier of the community to join
  - TU-profile: Thing-user’s profile

The JOIN_TSN_CONFIRM primitive shall have the following attributes: result, TSN-ID, CLUSTER_ID, TSN-profile, and reason.

- JOIN_TSN_CONFIRM (result, TSN-ID, CLUSTER_ID, TSN-profile, reason)
  - result: “SUCCESS,” “FAIL”
  - TSN-ID: identifier of the TSN(community) to join
  - CLUSTER-ID: cluster coordinator ID
  - TSN-profile: social group profile of the TSN to join
  - reason: error comment

Upon receiving the JOIN_TSN_REQUEST primitive, the TSN sublayer sends the JoinTSN request message to the TSN coordinator, as described in 7.2.5.

When the TSN sublayer of the TSN coordinator receives the JoinTSN request message, the TSN sublayer validates the thing-user and assigns a cluster to be joined.

If the assigned cluster coordinator is under the jurisdiction of the TSN coordinator, the TSN sublayer lets the cluster coordinator update the cluster profile and register the thing-user profile to the cluster database.

If the assigned cluster coordinator is not under the jurisdiction of the TSN coordinator, the TSN sublayer sends the JoinCLUSTER request message to the cluster coordinator, as described in 7.2.7.

When the TSN sublayer of the TSN coordinator is confirmed by the cluster coordinator or by receiving the JoinCLUSTER report message, as described in 7.2.8, the TSN sublayer shall respond by sending the JoinTSN report message containing cluster coordinator ID and TSN-profile to the home thing-user, as described in 7.2.6.

If the TSN sublayer of the TSN coordinator fails to join a TSN, the TSN sublayer shall respond by sending the JoinTSN report message containing an error comment.

When the TSN sublayer of the home thing-user receives the JoinTSN report message, the TSN sublayer confirms the join request to the TSC sublayer by using the JOIN_TSN_CONFIRM primitive, which fills TSN-ID, CLUSTER_ID and TSN-profile.

If the TSN sublayer of the home thing-user receives a blank object in the TSN-profile, the TSN sublayer reports the join request to the TSC sublayer by using the JOIN_TSN_CONFIRM primitive, which sets the result attribute as "FAIL" and fills the reason attribute with error comment text.

6.3.2.2 Leave a TSN and a cluster

The LEAVE_TSN_REQUEST and the LEAVE_TSN_CONFIRM primitives are used to leave a thing-user from a TSN.

The LEAVE_TSN_REQUEST primitive shall have the following attribute: TSN-ID.

- LEAVE_TSN_REQUEST (TSN-ID)
  - TSN-ID: identifier of the community to leave

The LEAVE_TSN_CONFIRM primitive shall have the following attributes: result, TSN-ID, and reason.

- LEAVE_TSN_CONFIRM (result, TSN-ID, reason)
  - result: "SUCCESS," "FAIL"
  - TSN-ID: identifier of the community to leave
  - reason: error comment

On receiving the LEAVE_TSN_REQUEST primitive, the TSN sublayer sends the LeaveTSN request message to the cluster coordinator, as described in 7.2.9.

When the TSN sublayer of the cluster coordinator receives the LeaveTSN request message, the TSN sublayer validates the thing-user.

The TSN sublayer of the cluster coordinator deletes the thing-user profile from the cluster database and updates the cluster profile.

The TSN sublayer of the cluster coordinator shall respond by sending the LeaveTSN report message containing TSN ID to the home thing-user, as described in 7.2.10.

When the TSN sublayer of the home thing-user receives the LeaveTSN report message, the TSN sublayer confirms the leave request using the LEAVE_TSN_CONFIRM primitive with appropriate attributes; the result attribute is set to 'SUCCESS' and the TSN-ID attribute is set to TSN-ID value obtained from the LeaveTSN report message.

6.3.3 Thing-user social community maintenance primitives

The TSC sublayer uses the thing-user social community maintenance primitives for maintaining the TSN and the cluster by providing a ballot procedure to select the TSN coordinator or cluster coordinator by its own rule.

6.3.3.1 Maintain TSN and cluster

The MAINTAIN_TSN_REQUEST primitive is used to elect a coordinator of the community or cluster by balloting.

The MAINTAIN_TSN_REQUEST primitive shall have the following attributes: action and TSN ID.

- MAINTAIN_TSN_REQUEST (action, TSN-ID)
  - action: "UPDATE," "RENEW_TSN," "RENEW_CLUSTER"
  - TSN-ID: identifier of the TSN or a cluster to maintain

The MAINTAIN_TSN_CONFIRM primitive shall have the following attributes: result, TSN ID and reason.

- MAINTAIN_TSN_CONFIRM (result, TSN-ID, reason)
  - result: “SUCCESS,” “FAIL”
  - TSN-ID: identifier of the TSN or cluster to maintain
  - reason: error comment

On receiving the MAINTAIN_TSN_REQUEST primitive with the action attribute set to “UPDATE,” the TSC sublayer of a cluster coordinator reports the status of the cluster to the community coordinator.

If the action attribute in the MAINTAIN_TSN_REQUEST primitive is “RENEW_TSN” or “RENEW_CLUSTER,” the TSN sublayer of the TSN coordinator or the cluster coordinator starts to renew the community or the cluster. The TSN sublayer sends the RenewCORD request message to the thing-users, as described in 7.2.11.

When the thing-user receives the RenewCORD request message, the TSN sublayer of the thing-user votes for a coordinator through responding to the VoteCORD request message, as described in 7.2.12.

The TSN sublayer of the coordinator collects the ballots from the received VoteCORD request messages. The vote result is sent to the elected coordinator by the RenewCORD report message.

When the TSN sublayer of the coordinator receives the AcceptCORD report message, the TSN sublayer advertises the new coordinator to the thing-users via sending the VoteCORD report message.

6.3.4 Thing-user subscribe/publish primitives

The TSC sublayer uses the thing-user subscribe/publish primitives for sharing the thing-user experiences with the thing-users who are in the TSN.

6.3.4.1 Subscribe to other thing-users

The SUBSCRIBE_TU_REQUEST and the SUBSCRIBE_TU_CONFIRM primitives are used for a thing-user to subscribe to other thing-users to share its knowledge.

The SUBSCRIBE_TU_REQUEST primitive shall have the following attribute: TU-ID.

- SUBSCRIBE_TU_REQUEST (TU-ID)
  - TU-ID: identifier of thing-user subscribed

The SUBSCRIBE_TU_CONFIRM primitive shall have the following attributes: result, TU-ID, and reason.

- SUBSCRIBE_TU_CONFIRM (result, TU-ID, reason)
  - result: “SUCCESS,” “FAIL”
  - TU-ID: identifier of thing-user subscribed
  - reason: error comment

On receiving the SUBSCRIBE_TU_REQUEST primitive, the TSN sublayer sends the SubscribeTU request message to the selected thing-user, as described in 7.2.16.

When the TSN sublayer of the subscribed thing-user receives the SubscribeTU request message, the TSN sublayer updates the subscription list of events sharing, and responds by sending the SubscribeTU report message, as described in 7.2.17.

When the TSN sublayer of the home thing-user receives the SubscribeTU report message, the TSN sublayer confirms to the TSC sublayer by using the SUBSCRIBE_TU_CONFIRM primitive by setting the result attribute to ‘SUCCESS’ and TU-ID attribute to the TU-ID value in the SubscribeTU report message.

6.3.4.2 Publish the event

The PUBLISH_TU_REPORT and the PUBLISH_TU_INDICAITON primitives are used for the publisher to announce its newly updated knowledge to its subscriber.

The PUBLISH_TU_ REPORT primitive shall have the following attributes: event-ID and event.

- PUBLISH_TU_ REPORT (event-ID, event);
  - event-ID: identifier of event published
  - event: new_Knowledge

The PUBLISH_TU_INDICAITON primitive shall have the following attributes: TU-ID, event.

- PUBLISH_TU_INDICATON (TU-ID, event)
  - TU-ID: identifier of thing-user published
  - event: new_Knowledge

On receipt of the PUBLISH_TU_ REPORT primitive, the TSN sublayer sends the PublishTU report message to the subscribed thing-user identified by the event-ID, as described in 7.2.18.

When the TSN sublayer of the subscribed thing-user receives the PublishTU report message, the TSN sublayer indicates the reception of the PublishTU report message by using the PUBLISH_TU_INDICATON primitive.

6.4 PDN SAP interface

The PDN sublayer provides the PDN service to the Thing-user Service Control sublayer through PDN SAP. This sublayer provides the proximal discovery service and proximal path management service.

6.4.1 Thing-user discovery primitives

The TSC sublayer uses the thing-user discovery primitives to search for a peer thing-user or peer thing-user group that can collaborate to accomplish a mission of the thing-user.

6.4.1.1 Specify proximity measure

The SET_PROXIMITY_REQUEST primitive and the SET_PROXIMITY_CONFIRMs are used to specify the proximity measure that is applied to choose a specific peer thing-user out of the pool of candidate peers discovered to accomplish a mission.

The SET_PROXIMITY_REQUEST primitive shall have the following attributes: mission, TU-profile and constraints.

- SET_PROXIMITY_REQUEST (mission, TU-profile, constraints)
  - mission: mission statement described by using the words registered in the vocabulary set of the NoE Registry
  - TU-profile: profile of the thing-user who requests to discover a peer thing-user
  - constraints: constraints on peers described by using the words registered in the vocabulary set of the NoE Registry

The SET_PROXIMITY_CONFIRM primitive shall have the following attributes: result, measure and reason.

- SET_PROXIMITY_CONFIRM (result, measure, reason)
  - result: “SUCCESS,” “FAIL”
  - measure: proximity measure ontology written in semantic web language using the words registered in the vocabulary set of the NoE Registry
  - reason: error comment

The TSC sublayer requests for the PDN sublayer to specify the proximity measure; It will be used to find thing-users to collaborate with according to the mission statement.

On receiving the SET_PROXIMITY_REQUEST primitive, the PDN sublayer generates the proximity measure ontology based on the mission statement and the thing-user capability information. If the facility of the thing-user is not enough to generate the proximity measure ontology; In this case, the PDN sublayer may request the cluster coordinator or the TSN coordinator for the proximity measure ontology using the SetPROXIMITY request message, as described in 7.3.1.

The PDN sublayer confirms the generation of proximity measure by using the SET_PROXIMITY_CONFIRM primitive, which contains the proximity measure.

6.4.1.2 Discover a peer thing-user or peer thing-user group

The DISCOVER_PEER_REQUEST and the DISCOVER_PEER_CONFIRM primitives are used to discover a peer or peer group with whom to collaborate.

The DISCOVER_PEER_REQUEST primitive shall have the following attributes: mission and measure.

- DISCOVER_PEER_REQUEST (mission, measure)
  - mission: mission statement described by using the words registered in the vocabulary set of the NoE Registry
  - measure: proximity measure ontology written in semantic web language using the words registered in the vocabulary set of the NoE Registry

The DISCOVER_PEER_CONFIRM primitive shall have the following attributes: result, path ID, peer ID and TU-profile and reason.

- DISCOVER_PEER _CONFIRM (result, path ID, peer ID, TU-profile, reason)
  - result: “SUCCESS,” “FAIL”
  - path-ID: proximal path ID
  - peer-ID: a thing-user ID or the ID list of thing-user who collaborates
  - TU-profile: profile of the peer thing-user who collaborates
  - reason: error comment

On receiving the DISCOVER_PEER_REQUEST primitive, the PDN sublayer starts to search for a thing-user from its neighbours.

The PDN sublayer sends the DiscoverPEER request message to the neighbour candidate peer thing-user, as described in 7.3.3.

The PDN sublayer of the candidate peer thing-user checks the possibility of collaboration based on the social capital of the home thing-user and the reciprocal advantage derived from the mission and proximity measure. The PDN sublayer of the candidate peer thing-user responds by sending the DiscoverPEER report message to the home thing-user.

When the PDN sublayer of the home thing-user receives the DiscoverPEER report message, the PDN sublayer checks the reason field to see whether or not the candidate thing-user accepts the collaboration request.

If accepted, the PDN sublayer confirms the peer thing-user discovery using the DISCOVER_PEER_CONFIRM primitive with appropriate attributes. The result attribute is set to “SUCCESS,” the peer-ID attribute is the value of the peer-ID field in the DiscoverPEER report message and the TU-profile attribute is the value of the peer-profile field in the DiscoverPEER report message.

It not accepted, the PDN sublayer sends the DiscoverPEER request message to the cluster coordinator.

The PDN sublayer of the cluster coordinator selects another candidate peer thing-users based on the mission and proximity measure and forwards the DiscoverPEER request message. The PDN sublayer of the candidate peer thing-user checks the social capital of the home thing-user and the possibility of collaboration and responds by sending the DiscoverPEER report message to the home thing-user.

If the PDN sublayer of the home thing-user discovers a peer thing-user, the PDN sublayer confirms the discovery of the peer thing-user. If the PDN sublayer of the home thing-user fails to discover a peer thing-user, the PDN sublayer sends the DiscoverPEER request message to the TSN coordinator.

The PDN sublayer of the TSN coordinator forwards the DiscoverPEER request message to the cluster coordinators. The PDN sublayer of the cluster coordinator selects candidate peer thing-users based on the mission and proximity measure and forwards the DiscoverPEER request message. The PDN sublayer of the candidate peer thing-user responds by sending the DiscoverPEER report message to the home thing-user.

If the PDN sublayer of the home thing-user discovers a peer thing-user, the PDN sublayer confirms that it has successfully discovered the peer thing-user. If the PDN sublayer of the home thing-user fails to discover a peer thing-user, the PDN sublayer reports that it has been unsuccessful in discovering the peer thing-user by using the DISCOVER_PEER_CONFIRM primitive, which sets the result attribute to “FAIL.”

6.4.2 Proximal path management primitives

The TSC sublayer uses the proximal path management primitives for maintaining the proximal path to a peer thing-user or peer thing-user group.

6.4.2.1 Maintaining a proximal path

The MAINTAIN_PEER_REQUEST and the MAINTAIN_PEER_CONFIRM primitives are used to manage those members of the coordinated group who participated in the PDN to provide thing-user centric communication services to a thing-user.

The MAINTAIN_PEER_REQUEST primitive shall have the following attributes: action and path-ID.

- MAINTAIN_PEER_REQUEST (action, path-ID)
  - action: “HEARTBEAT,” “RELEASE”
  - path-ID: proximal path ID

The MAINTAIN_PEER_CONFIRM primitive shall have the following attributes: result, path-ID and peer-ID.

- MAINTAIN_PEER _CONFIRM (result, path-ID, peer ID)
  - result: “SUCCESS,” “FAIL”

- path-ID: proximal path ID
- peer-ID: a thing-user ID or the ID list of thing-user who collaborates
- reason: error comment

On receiving the MAINTAIN_PEER_REQUEST primitive with action attribute set to “HEARTBEAT,” the PDN sublayer sends the MaintainPEER request message with action field set as “HEARTBEAT” to each of the peer thing-users coordinated in the proximal path group, as described in 7.3.5.

When the PDN sublayer of the peer thing-user receives a MaintainPEER request message, the PDN sublayer shall respond by sending the MaintainPEER report message, as described in 7.3.6.

The PDN sublayer of the home thing-user collects the MaintainPEER report messages from the peer thing-users and confirms using the MAINTAIN_PEER_CONFIRM primitive containing the collection of peer thing-user IDs.

On receiving the MAINTAIN_PEER_REQUEST primitive with action attribute set to “RELEASE,” the PDN sublayer sends the MaintainPEER request message with action field set as “RELEASE” to each of the peer thing-users coordinated in the proximal path group, as described in 7.3.5.

When the PDN sublayer of the peer thing-user receives a MaintainPEER request message, the PDN sublayer updates the thing-user profile and shall respond by sending the MaintainPEER report message.

The PDN sublayer of the home thing-user collects the MaintainPEER report message from the peer thing-users and reports to the TSC sublayer using the MAINTAIN_PEER_CONFIRM primitive with the collection of peer thing-user ID, as described in 7.3.6.

7 FN-NoE protocols

7.1 General

The coordinated networking layer of the FN-NoE is composed of capability blocks, as shown in Figure 9. These blocks include the NoE terminal thing-user management, thing-user centric networking control, thing-user social networking, coordinated experience management, coordinated peer discovery and proximal path management.

The NoE terminal thing-user management block maintains the profile and the status of the NoE terminal’s resources and capability skill set. This block manages thing-users such as the NoE terminal, regional virtual switch and overlay virtual switch, etc

The thing-user centric networking control block manages the process of socializing a thing-user and establishing a thing-to-thing connection.

The thing-user social networking block performs the process of organizing or disbanding a thing-user social community. This block controls an NoE terminal to join or leave a thing-user social community and publish or subscribe an experience sharing with a thing-user social community. The control protocols between the NoE terminal social networking blocks are defined at the reference point R2.

The coordinated experience management block maintains the collaborative networking experiences of the NoE terminals who joined thing-user social communities. This block finds experiences that meet requests from NoE terminals or the thing-user social community.

The coordinated peer discovery block performs the process of discovering a peer NoE terminal or the NoE terminals with which to form a collaborative work group. This block searches for a proximal NoE terminal from a thing-user social community, i.e. it discovers the regional virtual switches or overlay virtual switches.

The proximal path management block constructs a route between two proximal end points specified by the thing-user service control sublayer and provides interworking between heterogeneous operating routes. The control protocols between the proximal path management blocks are defined at the reference point R1.

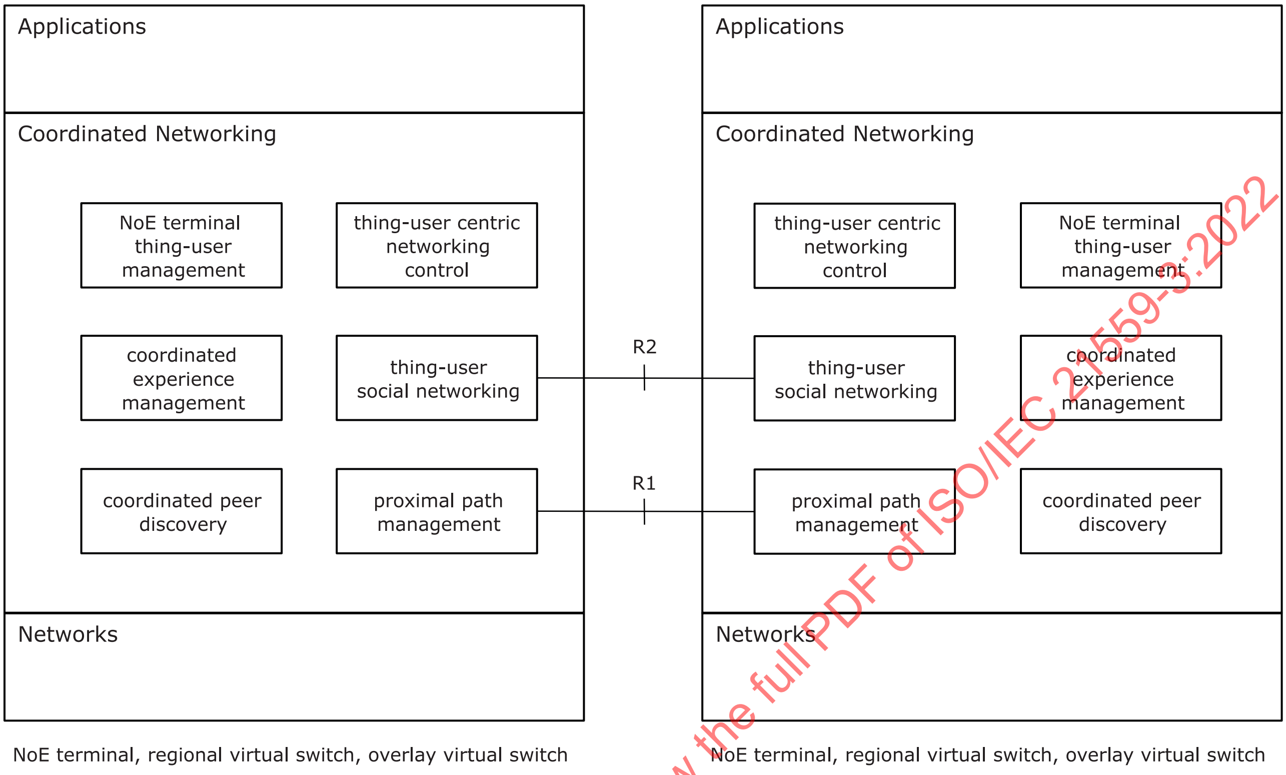
Applications
Coordinated Networking
NoE terminal thing-user management
thing-user centric networking control
coordinated experience management
thing-user social networking
coordinated peer discovery
proximal path management
Networks
NoE terminal, regional virtual switch, overlay virtual switch
Applications
Coordinated Networking
thing-user centric networking control
NoE terminal thing-user management
thing-user social networking
coordinated experience management
proximal path management
coordinated peer discovery
Networks
NoE terminal, regional virtual switch, overlay virtual switch
R2
R1

Figure 9 — Reference model of the FN-NoE

The control protocols between the thing-user social networking blocks of the peer-to-peer TSN sublayers are the TSN protocols. The control protocols between the proximal path management blocks of the peer-to-peer PDN sublayers are the PDN protocols.

The protocol message consists of communication method, command type and message fields.

NOTE The protocol message described in this clause is written in JSON text for explanation only. Consequently, it is also assumed that the message is transmitted in serialized JSON text format.

7.2 TSN protocols

The TSN protocols convey a message between the TSN sublayer of a thing-user and the TSN sublayer of a peer thing-user. The TSN protocol messages are: SearchTSN request and report; SetupTSN request and report; JoinTSN request and report; LeaveTSN request and report; RenewCORD request and report; VoteCORD request and report; AcceptCOORD report; SubscribeTSN request and report; and PublishTSN report.

7.2.1 Search a social community request message

The SearchTSN request message is used for a thing-user to contact a NoE Registry, which keeps all profile information about social communities and clusters, to find its community coordinator.

The SearchTSN request message shall have fields as follows:

```
{
    “method”: “request,”
```

```
        “CMD”: “SearchTSN,”
        “dTU-ID”: JSON text,
        “sTU-ID”: JSON text,
        “sTU-Profile”: JSON text
    }
```

The semantic of each field in the message is as follows:

- dTU-ID: NoE Registry Identifier
- sTU-ID: home Thing-user Identifier
- sTU-Profile: home Thing-user’s Profile

When the TSN sublayer of a thing-user wants to discover a social community that collaborates with the thing-user, this request message is sent to the NoE Registry.

When the NoE Registry receives the SearchTSN request message, the TSN sublayer of the NoE Registry searches the social community database and finds an associated TSN based upon sTU-Profile.

7.2.2 Search a social community report message

The SearchTSN report message is used to deliver the result to the thing-user who requests to search a TSN to join.

The SearchTSN report message shall have fields as follows:

```
    {
        “method”: “report,”
        “CMD”: “SearchTSN,”
        “dTU-ID”: JSON text,
        “sTU-ID”: JSON text,
        “TSN-Profile”: JSON text,
        “reason”: string
    }
```

The semantic of each field in the message is as follows:

- dTU_ID: home Thing-user Identifier
- sTU-ID: NoE Registry Identifier
- TSN-Profile: social group description objects of selected social group profile
- reason: error comment

When the TSN sublayer of the NoE Registry responds to a SearchTSN request message, NoE Registry sends this report message.

The TSN sublayer of the NoE Registry sends the SearchTSN report message, which contains a TSN-Profile. If it fails to find an appropriate TSN, the TSN-Profile field of the report message contains a blank object and the reason field contains an error comment text using the words registered in the vocabulary set of the NoE Registry.

7.2.3 Setup a new social group request message

The SetupTSN request message is used by the thing-user to ask the NoE Registry to allow a new setup of its social group.

The SetupTSN request message shall have fields as follows:

```
{
    “method”: “request,”
    “CMD”: “SetupTSN,”
    “sTU-ID”: JSON text,
    “dTU-ID”: JSON text,
    “TSN-Profile”: JSON text
}
```

The semantic of each field in the message is as follows:

- sTU_ID: home Thing-user Identifier
- dTU-ID: NoE Registry Identifier
- TSN-Profile: social group description profile

When the TSN sublayer of a thing-user requests for the NoE Registry to register a new social group in a social community, this request message is sent.

When the NoE Registry receives the SetupTSN request message, the TSN sublayer of the NoE Registry validates the new social group configuration and updates the social community database.

7.2.4 Setup a new social group report message

The SetupTSN report message is used by the NoE Registry to deliver the response to the new social group setup request. The SetupTSN report message shall have fields as follows:

```
{
    “method”: “report,”
    “CMD”: “SetupTSN,”
    “dTU-ID”: JSON text,
    “sTU-ID”: JSON text,
    “TSN-ID”: JSON text,
    “reason”: string
}
```

The semantic of each field in the message is as follows:

- dTU_ID: home Thing-user Identifier
- sTU-ID: NoE Registry Identifier
- TSN-ID: identifier of TSN (community) requesting to register
- reason: error comment

When the TSN sublayer of the NoE Registry responds to a SetupTSN request message, this report message is sent.

The TSN sublayer of the NoE Registry sends the SetupTSN report message, which contains TSN-ID value. If it fails to validate or to register the social group configuration, the TSN-ID field of the report message includes a blank object and the reason field contains an error comment written with the words registered in the vocabulary set of the NoE Registry.

7.2.5 Join a thing-user network request message

The JoinTSN request message is used for a thing-user to request a TSN coordinator to allow thing-user to join a TSN.

The JoinTSN request message shall have fields as follows:

```
{
    “method”: “request,”
    “CMD”: “JoinTSN,”
    “dTU-ID”: JSON text,
    “sTU-ID”: JSON text,
    “TSN-ID”: JSON text,
    “TU-Profile”: JSON text
}
```

The semantic of each field in the message is as follows:

- dTU-ID: TSN coordinator Identifier
- sTU_ID: home Thing-user Identifier
- TSN-ID: identifier of TSN (community) requesting to join
- TU-Profile: Thing-user profile

When the TSN sublayer of a thing-user requests the TSN coordinator for a thing-user to join the TSN, this request message is sent.

When the TSN coordinator receives the JoinTSN request message, the TSN sublayer of the TSN coordinator validates the thing-user and assigns a cluster to be managed.

The TSN sublayer sends the JoinCLUSTER request message to the cluster coordinator, if the assigned cluster coordinator is not located in the TSN coordinator.

7.2.6 Join a thing-user network report message

The JoinTSN report message is used to carry the result of joining the TSN.

The JoinTSN report message shall have fields as follows:

```
{
    “method”: “report,”
    “CMD”: “JoinTSN,”
    “dTU-ID”: JSON text,
        “sTU-ID”: JSON text,
        “TSN-ID”: JSON text,
        “CLUSTER-ID”: JSON text,
        “TSN-Profile”: JSON text,
        “reason”: string
    }
```

The semantic of each field in the message is as follows:

- dTU_ID: home Thing-user Identifier
- sTU-ID: TSN coordinator Identifier
- TSN-ID: identifier of TSN (cluster) requesting to join
- CLUSTER-ID: identifier of the cluster to manage the thing-user
- TSN-profile: social group profile of the TSN to join
- reason: error comment

When the TSN sublayer of the TSN coordinator responds to the JoinTSN request message, this report message is sent.

The TSN sublayer of the TSN coordinator sends the JoinTSN report message, which contains a TSN-ID, CLUSTER-ID and TSN-ID. If it fails to join the TSN, the reason field of the report message contains an error comment.

7.2.7 Join a cluster request message

The JoinCLUSTER request message is used by the thing-user or the TSN coordinator to request the cluster coordinator to join the thing-user to its cluster.

The JoinCLUSTER request message shall have fields as follows:

```
    {
        “method”: “request,”
        “CMD”: “JoinCLUSTER,”
        “dTU-ID”: JSON text,
        “sTU-ID”: JSON text,
        “TSN-ID”: JSON text,
        “TU-Profile”: JSON text
    }
```

The semantic of each field in the message is as follows:

- dTU-ID: assigned cluster coordinator Identifier
- sTU_ID: home Thing-user Identifier or the TSN (cluster) coordinator identifier
- TSN-ID: identifier of TSN (cluster) requesting to join
- TU-Profile: Thing-user profile

When the TSN sublayer of the thing-user requests the cluster coordinator to join a thing-user to the cluster, this request message is sent.

When the cluster coordinator receives the JoinCLUSTER request message, the TSN sublayer of the cluster coordinator validates the thing-user. The cluster coordinator updates the cluster profile and registers the thing-user profile to the cluster database.

7.2.8 Join a cluster report message

The JoinTSN report message is used to deliver the result of joining the cluster.

The JoinTSN report message shall have fields as follows:

```
{
    “method”: “report,”
    “CMD”: “JoinCLUSTER,”
    “dTU-ID”: JSON text,
    “sTU-ID”: JSON text,
    “TSN-ID”: JSON text,
    “CLUSTER-ID”: JSON text,
    “TSN-Profile”: JSON text,
    “reason”: string
}
```

The semantic of each field in the message is as follows:

- dTU_ID: home Thing-user Identifier or TSN coordinator Identifier
- sTU-ID: cluster coordinator Identifier
- TSN-ID: identifier of TSN (community) requesting to join
- CLUSTER-ID: identifier of the cluster to manage the thing-user
- TSN-profile: social group profile of the TSN to join
- reason: error comment

When the TSN sublayer of the TSN coordinator responds to a JoinCLUSTER request message, this report message is sent.

The TSN sublayer of the TSN coordinator sends the JoinCLUSTER report message, which contains a TSN-ID, CLUSTER-ID and TSN-ID. If it fails to join the cluster, the reason field of the report message contains an error comment.

7.2.9 Leave a TSN request message

The LeaveTSN request message is used when a thing-user requests the cluster coordinator to allow a thing-user to leave the cluster.

The LeaveTSN request message shall have fields as follows:

```
{
    “method”: “request,”
```

```
        “CMD”: “LeaveTSN,”
        “dTU-ID”: JSON text,
        “sTU-ID”: JSON text,
        “TSN-ID”: JSON text
    }
```

The semantic of each field in the message is as follows:

- dTU-ID: TSN coordinator or cluster coordinator Identifier
- sTU_ID: home Thing-user Identifier or the cluster coordinator identifier
- TSN-ID: identifier of TSN requesting to leave

When the TSN sublayer of a thing-user requests the cluster coordinator to allow the thing-user to leave the cluster, this request message is sent.

When the cluster coordinator receives the LeaveTSN request message, the TSN sublayer of the cluster coordinator validates the thing-user, deletes the thing-user profile from the cluster database, and updates the cluster profile.

7.2.10 Leave a TSN report message

The LeaveTSN report message is used to deliver the result of leaving the cluster.

The LeaveTSN report message shall have fields as follows:

```
    {
        “method”: “report,”
        “CMD”: “LeaveTSN,”
        “dTU-ID”: JSON text,
        “sTU-ID”: JSON text,
        “TSN-ID”: JSON text,
        “reason”: string
    }
```

The semantic of each field in the message is as follows:

- dTU_ID: home Thing-user Identifier or the cluster coordinator identifier
- sTU-ID: TSN coordinator or cluster coordinator Identifier
- TSN-ID: identifier of TSN requesting to leave
- reason: error comment

When the TSN sublayer of the TSN coordinator responds to a LeaveTSN request message, this report message is sent.

The TSN sublayer of the TSN coordinator sends the LeaveTSN report message, which contains the TSN-ID.

If it fails to leave the cluster, the reason field of the report message contains an error comment.